Department of Defense
High Performance Computing Modernization Program

DOD HPC MODERNIZATION PROGRAM

High Performance Computing

contributions to

DoD Mission Success 2012

2012 High Performance Computing Modernization Program

contributions to

DoD Mission Success

September 2012

2012 High Performance Computer Modernization Program Contributions to DoD Mission Success

The papers in this book comprise research endeavors and their successes conducted by users of the Department of Defense high performance computing community. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change.

ISBN # 978-0-9839566-1-7
Additional copies may be ordered from:

HPCMP
10501 Furnace Road
Suite 101
Lorton, VA 22079-2632
ATTN: Strategic Opportunities and Outreach

Editorial production by Denise O'Donnell and Lisa Powell
Cover art production by Lisa Powell
Printed in the United Stated of America by Minuteman Press

High Performance Computing Modernization Program (HPCMP)

Approved for public release; distribution is unlimited.

2012 High Performance Computing Modernization Program Contributions to DoD Mission Success

Table of Contents

Introduction 1
HPCMP Overview 1
HPCMP Impact on the Mission of the DoD 2

6.1 Basic Research

Air Force 3

Ab-Initio Molecular Dynamics Simulations of Molten Ni-Based Superalloys 5
Christopher Woodward and James Lill

Aero-Structural Characterization of Transonic Limit Cycle Oscillations 10
Crystal L. Pasiliao, A. Lechniak, and Keerti K. Bhamidipati

Application of CFD in Development of Large-Scale Scramjets (Challenge Project C4S Year 2) 20
John A. Boles and Mark A. Hagenmaier

Complex Network Modeling with an EMULAB HPC 31
Virginia Watson Ross and David L. Hench

Direct Numerical Simulation of Transition in a Hypersonic Cone Boundary Layer 40
Jayahar Sivasubramanian and Hermann Fasel

Highly Accurate Simulations of Low Reynolds Number Micro Air Vehicle Wing Aerodynamics 52
Raymond E. Gordnier, Miguel R. Visbal, and Caleb J. Barnes

Modeling Optical Distortions in a Turret Shear Layer for Feedback Flow Control 64
Christopher Porter, Robert Decker, and Jürgen Seidel

Numerical Investigation of Boundary-Layer Separation at Different Reynolds Numbers 73
A. Gross, S. Hosseinverdi, and H.F. Fasel

Army 85

3D Computational Analysis of the Turbulent Mixing of an Angled Jet in Crossflow 87
Luke Grant

Aerodynamic Performance of Perforated Wings for Micro-Aerial Vehicle Application 98
Shawn Aram, Rajneesh Singh, Surya Dinavahi, Aaron Harrington, and Christopher Kroninger

First-Principles Characterization of Armor Ceramics 107
DeCarlos E. Taylor and Iskander Batyrev

Large-Scale Molecular Dynamics Simulations of Early- and Intermediate-Stage Sintering of Nanocrystalline SiC 113
Bryce D. Devine, Jeffrey B. Allen, Charles F. Cornwell, and Charles R. Welch

Modeling the Combustion Chamber Dynamics of Two Selectable-Thrust Rocket Motor Concepts 122
Michael J. Nusca, Chiung-Chu Chen, and Michael J. McQuaid

Navy ***135***

Computational Studies of Oxidation States, Polar Ferroelectric Films, and Defects 137
Lai Jiang, Miguel Angel Méndez Polanco, Ilya Grinberg, Sergey V. Levchenko, Alexie M. Kolpak, and Andrew M. Rappe

Direct Numerical Simulation of Stably Stratified Turbulence at High Reynolds Number and Low Froude Number 147
S. Almalkie and S.M. de Bruyn Kops

High Power Density Energy Storage and Bio-Inspired Devices 160
J. Bernholc, V. Ranjan, B. Tan, M. Hodak, W. Lu, and M. Buongiorno Nardelli

High-Fidelity Large Eddy Simulations of Supersonic Jet Noise for Complex-Geometry Nozzles 168
Guillaume A. Brès, Frank Ham, Hung Lê, Joseph W. Nichols, Sanjiva K. Lele, and John T. Spyropoulos

Hot Supersonic Jet Noise Predictions using High-Resolution Large Eddy Simulation 180
James P. Erwin, Neeraj Sinha, and Gregory P. Rodebaugh

Numerical Investigation of Three-Dimensional Separation in External Flows 192
A. Gross, R. Jacobi, and H.F. Fasel

Structural and Magnetic Properties of $EuTiO_3$ from First Principles 201
Yurong Yang and L. Bellaiche

Other DoD Agencies ***207***

Multi-Scale Simulation of Deflagration-to-Detonation Transition in Turbulent Dense Two-Phase Reactive Mixtures 209
Kalyana C. Gottiparthi and Suresh Menon

Strong Coupling Expansion for the Nonequilibrium Properties of Ultracold Atoms on an Optical Lattice 219
J.K. Freericks, K. Mikelsons, and H.R. Krishnamurthy

6.2 Applied Research

Air Force ***229***

Can Legacy Codes Scale on Tens of Thousands of PEs or Do We Need to Reinvent the Wheel? 231
Robert F. Tomaro, William Z. Strang, and Kenneth E. Wurtzler

CFD Calculation of Aerodynamic Indicial Functions for a Generic Fighter Configuration 238
Martiqua L. Post, Mehdi Ghoreyshi, and Russell M. Cummings

Detecting Laminar-to-Turbulent Transition using TDLAS EXPLORER 249
Robert J. Yentsch, Nathan J. Mullenix, Datta V. Gaitonde, and Michael S. Brown

Engine-Level Simulation of Liquid Rocket Combustion Instabilities: Trans-critical Combustion Simulations in Single Injector Configurations 257
Nicolas Guézennec, Matthieu Masquelet, Suresh Menon, Ramakanth Munipalli, Edward D. Lynch, Brian Lariviere, and Douglas G. Talley

High Resolution Near-Real-Time Image Processing and Support for MSSS Modernization 268
R. Bruce Duncan, Chris A. Sabol, Kathy Borelli, Scott Spetka, Jason Addison, Adam Mallo, Brad Farnsworth, Ron Viloria, and George Ramseyer

High-Fidelity Analysis in Support of the HIFiRE-1 and -2 Programs 276
F.J. Malo-Molina, D. Risha, Houshang B. Ebrahimi, R. Yentsch, and D.V. Gaitonde

Investigation of Total Pressure and Swirl Distortion in a Single-Stage Transonic Fan 290
Michael G. List and Rebecca M. Howard

Simulation of Distortion Generation in a Modern Serpentine Diffuser to Improve Experimental Effectiveness 299
Darius D. Sanders and Michael G. List

Using Kestrel in the Cloud 308
Josh Calahan, Todd Tuckey, Alex Cabello, Jason Hirata, and Steve Caldwell

Verification of a GPU Enabled 3D Solver for Hypersonic Flight Analysis 315
Evgeny Titov, Jonathan Burt, and Eswar Josyula

Army 323

A 3D Studio Max Rendering Approach to Achieve Optimal Single-Dimension Geometry Exaggeration Necessary in Some Scientific Visualization Analysis Scenarios 325
Richard I. Walters and Mary A. Cialone

Accuracy and Efficiency Enhancements for High-Speed Aero-Propulsive Flows with Combustion 330
A.C. Zambon, K. Brinckman, G. Feldman, D.C. Kenzakowski, S.M. Dash, and K.D. Kennedy

Computational Investigation and Fundamental Understanding of a Slowed UH-60A Rotor at High Advance Ratios 342
Mark Potsdam, Anubhav Datta, and Buvana Jayaraman

Construction of Accurate Reactive Potentials for Large Scale Molecular Dynamics Simulations of Materials Under Extreme Conditions 354
Edward F.C. Byrd and Neil Scott Weingarten

Emulation of Forward-Looking Radar Technology for Threat Detection in Rough Terrain Environments: A Scattering and Imaging Study 363
DaHan Liao, Traian Dogaru, and Andy Sullivan

High Performance Java 375
Jordan J. Ruloff, James A. Ross, David A. Richie, Song J. Park, Dale R. Shires, and Brian J. Henz

Multiple Simultaneous Domain Decomposition of Interdisciplinary Mobile Network Simulations via Distributed Shared Memory 382
Jerry Clarke, Kenneth Renard, and Kenneth W. Leiter

Technology Transfer to Army Fielded Combat Robots 387
Peter Raeth, Juan Carlos Chaves, and Sean M. McGhee

Unsteady Aerodynamic Simulations of Maneuvering Projectiles 390
Jubaraj Sahu, James DeSpirito, and Karen R. Heavey

Workstation Footprint Tactical Computing Demonstration 401
Song J. Park, Dale R. Shires, Brian J. Henz, James A. Ross, Jordan J. Ruloff, and David A. Richie

Navy 407

Application Development on Convey's HC-1ex Using Bluespec Tools at the Naval Research Laboratory 409
Stephen Bique and Rishiyur Nikhil

Computational Analysis for Air/Ship Integration: Final Report 420
Susan A. Polsky

Effect of Nozzle-Exit Turbulence Level on the Flow and Acoustic Properties of Imperfectly Expanded Supersonic Jets 428
J. Liu, K. Kailasanath, J.P. Boris, N. Heeb, D. Monday, and E. Gutmark

General Purpose Computing on Graphics Processing Units: Optimization Strategy 438
Henry Au and Gregory Lum

High-Resolution and Probabilistic Tropical Cyclone Track and Intensity Prediction 444
James Doyle, Carolyn Reynolds, Sue Chen, Yi Jin, Hao Jin, Chi-Sann Liou, Justin McLay, Jon Moskaitis, James Ridout, and Richard Hodur

Numerical Flow Analysis of Planning Boats 455
Kyle A. Brucker, Thomas T. O'Shea, and Douglas G. Dommermuth

Predicting Tides in a High Resolution Ocean General Circulation Model 467
James G Richman, E. Joseph Metzger, Jay F. Shriver, Alan Wallcraft, Ole Martin Smedstad, and Brian K. Arbic

Prediction of Rogue Waves by Large-Scale Phase-Resolved Nonlinear Wave-Field Simulations 476
Wenting Xiao, Yuming Liu, and Dick K.P. Yue

Toward Efficient Calculations of High Speed Jet Noise 484
Chris C. Nelson, Alan B. Cain, Philip J. Morris, and Yongle Du

6.3 Advanced Technology Development

Air Force 495

Application of the Graphics Processing Unit to Computational Electromagnetics Using the Finite Element Method 497
Huan-Ting Meng, Baolin Nie, Jian-Ming Jin, Steven Wong, and Charles Macon

PETTT Signal and Image Processing CTA 507
Juan Carlos Chaves, Jose Unpingco, Bracy Elton, and Peter Raeth

Portal Development for High Performance Computing (HPC) at Maui High Performance Computing Center (MHPCC) 512
David Morton

Army 523

Accelerating Chemical Kinetics Calculations with GPUs 525
Christopher P. Stone and Professor Roger R. Davis

Augmenting Live Network Experiments with High Performance Network Simulation and Emulation 537
Kenneth D. Renard and Brian J. Henz

Isolating Runtime Faults with Callstack Debugging using TAU 541
Sameer Shende, Allen D. Malony, John C. Linford, Andrew Wissink, and Stephen Adamec

LWIR Background Signature Prediction for Very Large Scenes using the High-Fidelity Background Thermal Solver (HBTS) 551
Denise Talcott, Kyle Rokos, Udaya Ranawake, Randal Cannon, and Frank Carlen

Navy 561

Towards Prediction of Hydro-Acoustics using Large Eddy Simulations 563
Peter A. Chang III, Michael D. Mattson, Joseph Slomski, Hua Shan, and Krishan Mahesh

6.4 Advanced Component Development and Prototypes

Army 579

Distributed HPC-enabled LVC S&T Experimentation Environments 581
Daniel Ku, Rodney Smith, and Young Choi

Early Experiences with Energy-Aware Scheduling 586
Kathleen A. Smith, Thomas M. Kendall, William DeSalvo, Michael P. Knowles, Phillip G. Matthews, James C. Ianni, and Chris Sauerwald

OneSAF as a Simulation Service using High Performance Computing 595
Christina Bouwens, Amy Henninger, Gloria Flowers, and Alesya Paschal

Progress Report on the Microsoft HPC Portal 605
Rob Fisher, Pat Collins, Jim Waterman, Robert Ames, Ian Shields, and Manuel Vindiola

Author Index 613

Introduction

The High Performance Computing Modernization Program (HPCMP) was initiated in 1992 in response to Congressional direction to modernize the Department of Defense (DoD) laboratories' HPC capabilities. The Program was executed as an activity of the Office of the Director of Defense Research and Engineering (DDR&E) (now Assistant Secretary of Defense for Research and Engineering (ASD(R&E)), until the devolvement of management of the HPCMP to the Office of the Deputy Assistant Secretary of the Army, Research and Technology (DASA(R&T)) in FY 2012. Since its inception, the Program has realized great success in its efforts to modernize the Department's laboratories and test centers to levels on par with other world-class organizations.

This technical report, *2012 High Performance Computing Modernization Program Contributions to DoD Mission Success*, will provide a variety of papers highlighting the research that is being accomplished using HPCMP resources. This report is divided into four of the seven Budget Activity (BA) categories: 6.1 Basic Research, 6.2 Applied Research, 6.3 Advanced Technology Development, and 6.4 Advanced Component Development and Prototypes. The three BA categories not represented in the book are: 6.5 System Development and Demonstration, 6.6 RDT&E Management Support, and 6.7 Operational Systems Development. Each category is subdivided by the Service or Agency. Below is a brief description of the categories represented in this book[1]:

6.1 Basic Research is the systematic study to gain knowledge or understanding of the fundamental aspects of phenomena and of observable facts without specific applications, processes, or products in mind. Basic research involves the gathering of a fuller knowledge or understanding of the subject under study. Major outputs are scientific studies and research papers.

6.2 Applied Research is the systematic study to gain knowledge or understanding necessary for determining the means by which a recognized and specific need may be met. It is the practical application of such knowledge or understanding for the purpose of meeting a recognized need. This research points toward specific military needs, with a view toward developing and evaluating the feasibility and practicability of proposed solutions and determining their parameters. Major outputs are scientific studies, investigations, research papers, hardware components, software codes, and limited construction of, or part of, a weapons system to include non-system-specific development efforts.

6.3 Advanced Technology Development is the systematic use of the knowledge or understanding gained from research directed toward proof of technological feasibility and assessment of operational and productibility, rather than the development of hardware for service use. Major outputs are demonstrations intended to prove or test a technology or method.

6.4 Advanced Component Development and Prototypes evaluates integrated technologies, in as realistic an operating environment as possible, to assess the performance or cost reduction potential of advanced technology. Programs in this phase are generally system-specific. Major outputs are hardware and software components, or complete weapons systems, ready for developmental and operational testing and fielding.

HPCMP Overview

The HPCMP was consolidated out of two dozen small HPC departments that had independently evolved within the Army, Air Force, and Navy laboratories and test centers. Today, the HPCMP provides the supercomputer capability, high-speed network communications, and computational science expertise that enables the Defense laboratories and test centers to conduct a wide range of focused research,

[1] DoD Financial Management Regulation, Volume 6A, Chapter 11, November 2011.

development, and test activities. The goal of the Program is not simply to provide supercomputing services to the DoD, but in so doing to create a pervasive culture of computation within DoD that drives the routine use of advanced computational environments to solve the Department's most critical mission challenges. In support of this broader goal, the HPCMP provides an array of services to its users.

The HPCMP accelerates research, development and test efforts by providing computational tools and environments that allow DoD scientists and engineers to manage risks in S&T, T&E, operations, and acquisitons by more quickly and effectively finding solutions from a wider selection of robust alternatives. The DoD uses these tools across a wide breadth of disciplines, such as physics, chemistry, materials science, electromagnetics, acoustics, aerodynamics, and hydrodynamics in support of basic and applied research, systems development, and test and evaluation. Within the past ten years, the impact has been documented through numerous success stories that track technology innovations, often starting as basic research or advanced development, and continuing through weapons systems development and testing. Although economic analysis only reveals part of the story, a recently completed Return on Investment study for the Program, validated by the National Defense University, showed that, conservatively, the HPCMP has yielded a $7 return to the DoD for every dollar invested.

HPCMP Impact on the Mission of the DoD

Beginning with the vital role of electronic calculation machines in support of allied efforts during World War II, computers have become a key element of the economic, technological, and scientific successes of the United States. Today US institutions of innovation, discovery, and economic well-being—from industry and academia to the agencies of the Federal government, including the DoD—cannot achieve their goals without HPC.

HPC consists of a cluster of activities that enable the development and application of computer hardware, support software, and advanced algorithms to address highly complex, computationally challenging problems. HPC is distinguished from the general class of computing by its scale: HPC addresses challenges that are either compute-intensive, requiring a very large number of processors working together at one time; communication-intensive, requiring specialized networks supporting high-speed communications among processors and/or distributed systems; or both.

Within the DoD, HPC is used in a remarkably broad array of activities, and HPCMP supercomputers and expertise advance the Department's fundamental understanding of materials, aerodynamics, chemistry, fuels, acoustics, signal image recognition, electromagnetics, and other areas of research; as well as enabling advanced test and evaluation environments such as synthetic scene generation, automatic control systems, and virtual test environments. HPC is proving to be an essential part of the DoD's ability to design, test, produce, and sustain DoD weapons systems at reduced cost, time, and risk with higher performance.

The importance of supercomputing to our security is demonstrated by the direct, measurable impact the High Performance Computing Modernization Program resources and expertise have had, and continue to have, on the mission of the DoD. HPC was identified as a national strategic asset by President Obama during his 2011 State of the Union address.

6.1 Basic Research

Air Force

Ab-Initio Molecular Dynamics Simulations of Molten Ni-Based Superalloys

Christopher Woodward
US Air Force Research Laboratory, Materials and Manufacturing Directorate (AFRL/RX), Wright-Patterson AFB, OH
christopher.woodward@wpafb.af.mil

James Lill
High Performance Technologies Inc. (HPTi), Wright-Patterson AFB, OH
james.lill@wpafb.af.mil

Abstract

Fundamental properties of molten Ni-based super-alloys are calculated using ab initio molecular dynamics simulations based on Density Functional Theory. In this work, predictions for density inversion increases in liquid metal density with increasing temperature, are made for a model Ni-Al-W alloy and the Ni-based super-alloys RENE-N4 and CMSX-4. Previous calculations on simple, binary and ternary Ni alloys illustrated proof-of-concept of this method, predicting liquid-phase molar volumes (V(c,T)) within 0.6–1.8% of available experimental data. Simulation cells of 500 atoms, run for approximately 10 psec, were adequate to converge liquid metal densities and diffusion rates. Predicted liquid metal densities for Ni-Al-W alloys, for target compositions and temperatures consistent with 0 and 0.5 solid volume fractions in the melt, produce approximately a 2% density inversion. The liquid metal density for a Ni-based super-alloy, RENE-N4, is calculated at the liquidus and solidus temperatures expected in the mushy zone. A series of 500 atom instantiations of the super-alloy melt produce self-consistent densities that are well within expected numerical error. This illustrates that simulations of moderate spatial and temporal scales sample enough degrees-of-freedom to properly represent the configuration entropy found in a liquid metal super-alloy. Density inversions are predicted for both RENE-N4 and CMSX-4 for chemistries and temperatures consistent with 0 and 0.4 solid volume fractions in the melt.

1. Introduction

Over succeeding generations of turbine engines, the increasing complexity in alloy chemistry and geometry of turbine airfoils has increased the probability of forming processing defects (freckles) during directional solidification. These calculations offer a new approach for determining the fundamental parameters required for modeling aspects of this processing path. Convective instabilities responsible for freckle defects are produced by variations in liquid-metal density with composition and temperature across the solidification zone. A quantitative model, predicting processing regimes for defect-free castings over the widest range of refractory metal compositions, is required for a systems design approach leading to optimal material and component performance. Models of thermo-chemical convection in the mushy zone have shown that such instabilities occur when a Rayleigh number exceeds a critical value[1]. This Rayleigh number, R, is a measure of the ratio of the buoyancy force to the retarding frictional force:

$$R = \left(\Delta\rho / \bar{\rho}\right) gKl / \alpha v \tag{1}$$

Where l is an appropriate length scale, K is the average permeability, g gravitational acceleration, α is the thermal diffusivity, v is the kinematic viscosity of the fluid, and $(\Delta\rho/\rho)$ is the density contrast. This last term is the variation in mass density between the liquid metal near the solid-liquid dendrite interface, and the cooler melt at the bottom and sides walls of the mold. This depends on casting temperature gradients, as well as the variations in solute concentration produced by the equilibrium partitioning between solid and liquid phases. Verifying models of freckle formation for a specific alloy system requires accurate values for the parameters that describe this Rayleigh number. Many of these constants, such as the viscosity, permeability and density, are difficult to measure and are not well-documented for the super-alloys. Recently, Mukai and coworkers have developed a multi-component parameterization for V(c,T) in super-alloys based on extensive experimental measurements of the densities of binary liquid Ni alloys, and a few representative ternaries[2]. Assuming that other parameters for the Rayleigh criteria are constant across the mushy zone, the signature for

freckle formation is reduced to the condition when the liquid metal at higher-temperature has a greater density than at lower-temperatures. Density inversion is more complex than just a negative coefficient of thermal expansion, because the (equilibrium) composition of the liquid changes with temperature across the height of the mushy zone. The current simulations offer a new way of determining such densities and an independent test of the accuracy of the model for Mukai's parameterization of $\rho(c,T)$.

2. Method

The properties of molten Ni-based alloys are calculated using *ab-initio* molecular dynamic (AIMD) simulations on relatively small simulation cells consisting of 500 atoms. Ionic positions are evolved in time using classical Nosé-Hoover dynamics, for a fixed-number, volume, and temperature (NVT) and fixed-number, pressure and temperature (NPT) ensembles. Time evolution is based on inter-atomic forces (i.e., the Hellmann-Feynman forces) computed directly from density functional theory (DFT) using the commercial software VASP (Vienna *ab-initio* simulation package)[3]. Previously, we have used NVT dynamics to calculate the molar volumes of a variety of simple, binary and ternary metal alloys[4]. These included Ni, Al, Ni-X, and Ni-Al-X (X=W, Re, Ta) alloys at several compositions and temperatures. The previous DFT calculations made use of ultrasoft pseudo-potentials[5,6] and the PW91 generalized-gradient approximation (GGA)[7] with a plane-wave basis cutoff energy of 260 eV. The simulations employed a single k-point (Γ) and time-steps (Δt) for the MD simulations were chosen as Δt=0.002 ps or Δt=0.003 ps, to ensure that the conserved energy in the Nosé-Hoover dynamics displayed a drift in time no larger than 1 meV/atom·ps. Radial distribution functions from these simulations converge to the uncorrelated limit of unity in a distance less than half the diagonal distance across the simulation cell. This illustrates that the cell dimensions are large enough to produce uncorrelated motion of individual and groups of atoms. Total simulation times ranged from 5–10 ps, a time-interval long enough to obtain good statistical precision in calculated molar volumes and diffusion rates[4].

The current calculations follow the procedure described above[4] with the following exceptions. The 500 atom NPT ensembles were modeled using the Projected Augmented Wave (PAW) method,[8,9] and the targeted temperatures and compositions are representative to the conditions expected in the solidification (mushy) zone. The PAW method was found to give a more robust solution for the electronic structure for the complex chemistries found in the RENE and CMSX alloys.

3. Results

Three alloys are considered for possible density inversion, Ni-Al-W, CMSX-4, and Rene-N4. For the more complicated alloys, it is first necessary to demonstrate that the 500 atom cell AIMD simulations can give accurate and reproducible densities for simulations running for less than ~10 ps. To illustrate this, four instantiations of Rene-N4, based on different random initial atomic distributions, were run at the solidus and liquidus temperatures for compositions determined from the Scheil solidification path, estimated using CalPhad (Pandat®) methods. The predicted compositions are Ni62.2-Cr11.0-Co5.2Mo1.3-W1.9-Al10.-Ti5.8-Ta2.3-Nb0.5 and Ni63.0-Cr11.2-Co7.6-Mo0.9-W1.9-Al9.2-Ti4.3-Ta1.6-Nb0.3 at 1,635K and 1,582K, respectively. The very dilute Nb additions were removed from the calculations with the balance in composition being spread proportionally overall the remaining species yielding the 500 atom cell compositions: $Ni_{315}Cr_{56}Co_{38}Mo_5W_{10}Al_{46}Ti_{22}Ta_8$ and $Ni_{312}Cr_{55}Co_{26}Mo_6W_{10}Al_{50}Ti_{29}Ta_{12}$ for 1,635K and 1,582K respectively. This notation for the super-cell composition denotes the number of each atomic species included in the 500 atom simulation cell. These chemistries were then used to construct 500 atom cells with random distributions of the appropriate elements. The different instantiations at each temperature were run in order to assess the influence of configurational entropy on the molar volume or alloy density. After 8 ps the derived time-averaged densities of the four cases at each temperature were found to be within statistical error, illustrating the efficacy of this approach. For this range of compositions and temperatures we conclude that a random configuration is a reasonable approximation of the melt, and will give reproducible liquid metal densities.

For the model Ni-Al-W alloy, we adopted a nominal composition of Ni-14Al-3W (at%) and evaluated the equilibrium liquid compositions and temperatures for 0.5 and 1.0 liquid fraction (wt%) using the Scheil model. This alloy was chosen because it has been demonstrated to produce freckle defects in conventional Bridgeman directional solidification. The predicted liquid compositions at 1,711 and 1720K are shown in Table 1. The AIMD calculations predict a 2% increase in the density going from 1,711 to 1,720K, reflecting a relatively strong density inversion. The Scheil model predicts a change in composition that favors an increase in density at the higher temperature. However, for this system there is a decrease in the molar volume with increasing temperature, from 7.782(5) to 7.767(8) cm^3/mole, so density inversion is

predicted even when the increase in elemental mass is not taken into account. When applied to this model alloy, the Mukai parameterization produces a large density inversion at these temperatures and chemistries (Table 1). The estimated molar volumes are 7.7367 and 7.7302 for 1,711 and 1,720K respectively, which in consistent with a small density inversion based on just volumetric effects within the Mukai parameterization.

Table 1. Predicted liquid metal densities for the Ni-Al-W model system. Here f_L is the fraction liquid (wt%) and the numbers in parenthesis indicate the 95% confidence level in the last digit.

		Composition	*Density (g/cm^3)*	
f_L	**T (K)**	**Atomic (/500)**	**AIMD**	**Mukai[2]**
1.0	1720	$Ni_{415}Al_{70}W_{15}$	7.46(2)	7.5040
0.5	1711	$Ni_{410}Al_{78}W_{12}$	7.29(1)	7.3352

Temperatures and compositions were also assessed from Sheil calculations for 0.6 and 1.0 liquid fraction (wt%) for Rene-N4 and CMSX-4. Compositions for these temperatures and alloys are shown in Table 2. Results for all these calculations are shown in Figure 1. AIMD predictions for both Rene-N4 and CMSX-4 suggest density inversions under these conditions, though they are smaller than that seen in Ni-Al-W. Note that two instantiations were run for each temperature composition for these super-alloys. Also, the increase in density for CMSX-4 is approaching the statistical error of the predictions. The Mukai parameterization does not predict a density inversion for CMSX-4 these compositions and temperatures. However, the DFT predictions and the Mukai parameterization for the Ni-Al-W alloys are very similar in both absolute size and in the degree of density inversion. Finally, both methods predict a density inversion for Rene-N4 at these temperatures and compositions. For Rene-N4 the molar volume also reflects a density inversion for both the AIMD and Mukai parameterization.

Table 2. The compositions used in the current calculations for Rene-N4 and CMSX-4 at 0.6 and 1.0 liquid fraction (wt%). The atomic percent and the number of atoms in the simulation cell are given for each atomic species.

Alloy	Liquid fraction	Conc	Ni	Co	Cr	Mo	W	Re	Al	Ta	Ti
Rene-N4	0.6 f_L (1604K)	at%	62.5	7.2	11.3	1.0	1.7	0.0	9.4	2.0	4.9
		#	312	36	57	5	8	0	47	10	25
	1 f_L (1622K)	at%	63.3	7.5	11.2	0.9	1.9	0.0	9.2	1.6	4.3
		#	316	37	56	5	10	0	46	8	22
CMSX-4	0.6 f_L (1635K)	at%	63.2	8.8	7.7	0.4	1.8	0.7	13.2	2.8	1.4
		#	316	44	38	2	9	4	66	14	7
	1 f_L (1654K)	at%	63.8	9.3	7.6	0.4	2.0	1.0	12.6	2.2	1.3
		#	319	46	38	2	10	5	63	11	6

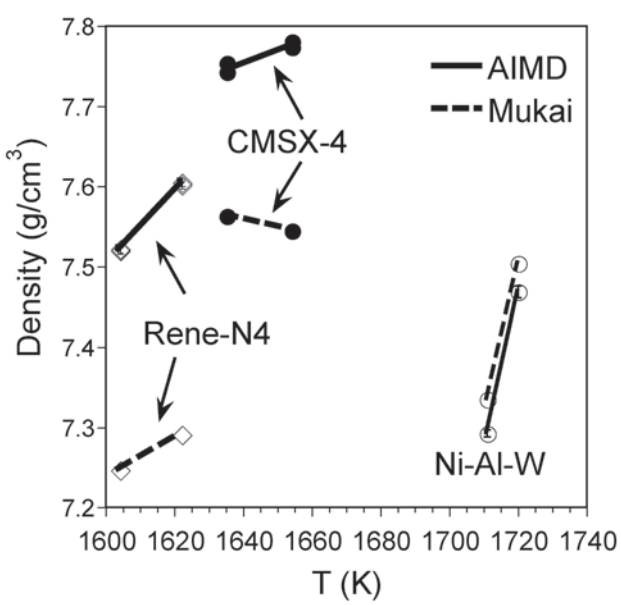


Figure 1. Density as a function of temperature for Ni-Al-W, Rene-N4 and CMSX-4 for 0.6 and 1.0 liquid fraction (wt%)

Given the different results for CMSX-4, and the shallow gradient found in the AIMD calculations for this alloy, further calculations were undertaken to better describe the change in density in the mushy zone. A series of calculations were performed for liquid volume fractions ranging from the expected liquidus and solidus. These results, shown in Figure 2, illustrate the expected trends in liquid metal density during solidification. The additional data points suggest that within statistical error CMSX-4 is predicted to sample density inversion between 0.4 and 1.0 liquid fraction in the melt. All three alloys are shown in Figure 2 to illustrate the relative strength of the density inversion. Finally, from Equation 1, another measure of the influence of the density inversion is the density contrast over the mushy zone. This is the dimensionless quantity $\Delta\rho/\rho$ defined as $(\rho_0-\rho(h))/\rho_0$, where ρ_0 is the density at the liquidus and $\rho(h)$ is the density at a specific height in the mushy zone. In this case we use the values of the density at 0.6 liquid fraction (wt%) for $\rho(h)$. Results for the three alloys are shown in Table 3. While the AIMD predictions for the density contrast of CMSX-4 have the opposite sign of the Mukai predictions (consistent with a density inversion) the rank ordering of density contrast is consistent between the two methods. Also, it is worth noting that the rank ordering of the density contrast is in agreement with the predictions of Reed and coworkers, based on ThermoCalc predictions (for chemistry) across the height of the mushy zone[10].

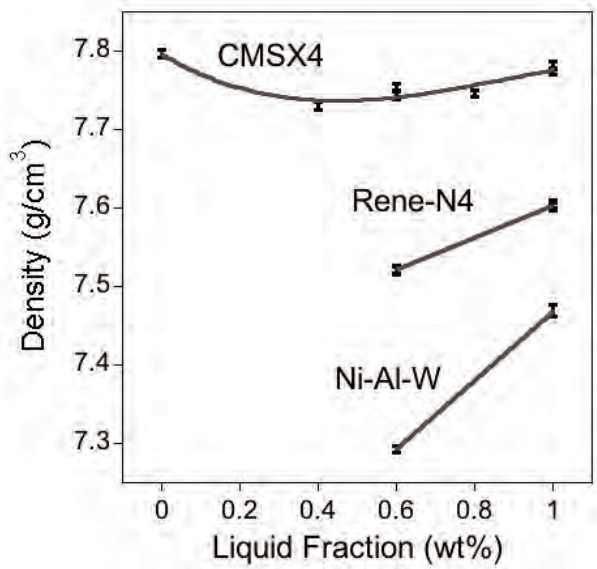


Figure 2. Calculated liquid metal densities for CMSX-4, Rene-N4 and a model Ni-Al-W alloy for a range of liquid fraction (wt%)

Table 3. Predicted density contrast (Δρ/ρ) based on the liquidus and 0.6 liquid fraction (wt%) densities

Method	CMSX-4	Rene-N4	Ni-Al-W
AIMD	0.00373	0.01085	0.02355
Mukai [2]	-0.00245	0.00609	0.02251

4. Summary

AIMD simulations have been performed to derive the time-averaged properties of liquid metals for alloys ranging from simple metals to Ni-based super-alloys such as Rene-N4 and CMSX-4. For the simpler alloys, diffusion rates are in good agreement with available experimental measurements, and the radial distribution functions show signs of short-range ordering consistent with Ni being surrounded by unlike neighbors. Results show that the AIMD method is sampling an adequate volume and time-domain. Predicted molar volumes for elemental, binary and ternary Ni alloys are typically within ~2% of measured values, but underestimate the coefficients of thermal expansion as compared to experimental measurements using contact methods. The calculations confirm the assumption that the molar volumes of complex alloys can be described using a linear combination of results from simpler binary alloys.

Densities of three freckle prone alloys, Ni-Al-W, Rene-N4 and CMSX-4, were studied for compositions and temperatures expected in the mushy zone. Calculations for the model ternary Ni-Al-W alloy predict a 2% density inversion, consistent with convective instabilities that produce freckle defects. Simulations of reference Rene-N4 alloys, based on random distributions of atomic species, were used to illustrate the precision and reproducibility of molar volume predictions using a 500 atom cell. Calculations for Rene-N4 and CMSX-4 also produce density inversions, though smaller

than that found for the model Ni-Al-W alloy. While the Mukai parameterization produced the same ranking in terms of the size of density contrast for these three alloys, CMSX-4 is not predicted to have a density inversion in contrast with the AIMD results. Overall, these results suggest that even for highly-complex chemistries, moderately-sized AIMD simulations can produce reliable predictions for density inversions. Proof-of-concept has been demonstrated on two commercial alloys used in current high-performance turbine engines. With proper validation, this approach could be used to assess convection instabilities in a wide-range of materials. Similar methods can be used to predict other parameters used in the Rayleigh number criteria, such as the shear viscosity, that are difficult to measure and are not well-known for complex alloys.

Acknowledgements

This research was supported in part by a grant of computer time from the DoD High Performance Computing Modernization Program at the US Air Force Research Laboratory DoD Supercomputing Resource Center (AFRL DSRC). The authors gratefully acknowledge insightful discussions with D. Trinkle, M. Asta, T. Pollock, P. Voorhees, J. Miller, O. Senkov, S.H. Davis, and the technical assistance provided by S. Kajihara at the AFRL DSRC.

References

1. Beckermann, C., J.P. Gu, and W.J. Boettinger, *Met. Trans. A*, 31A, 2545, 2000.
2. Mukai, K., Z.S. Li, and K.C. Mills, *Met. Mater. Trans. B*, 36, 255, 2005.
3. Kresse, G. and J. Hafner, *Phys. Rev. B*, 49, 14251, 1994; Kresse, G. and J. Furthmüller, *J. Computat. Mater. Sci.*, 6, 15, 1996.
4. Woodward, C., M. Asta, D.R. Trinkle, J. Lill, and S. Angioletti-Uberti, *Jour. Applied Phys.*, Vol. 107, 113522, 2010.
5. Vanderbilt, D., *Phys. Rev. B*, 41, 7892, 1990.
6. Kresse, G. and J. Furthmüller, *Phys. Rev. B*, 54, 11169, 1996.
7. Perdew, J.P. and Y. Wang, *Phys. Rev. B*, 45, 13244, 1992.
8. Blöchl, P.E., *Phys. Rev. B*, 50, 17953, 1994.
9. Kresse, G. and J. Joubert, *Phys. Rev. B*, 59, 1758, 1999.
10. Reed, R.C., T. Tao, and N. Warnken, *Acta Mat.*, 57, 5898, 2009.

Aero-Structural Characterization of Transonic Limit Cycle Oscillations

Crystal L. Pasiliao
US Air Force Research Laboratory, Munition Aerodynamic Sciences Branch (AFRL/RWWV), Eglin AFB, FL
crystal.pasiliao@eglin.af.mil

Jason A. Lechniak and Keerti K. Bhamidipati
US Air Force Flight Test Center (812 TSS/ENTT), Edwards AFB, CA
{jason.lechniak, keerti.bhamidiapti}@edwards.af.mil

Abstract

There exists a significant need for detailed understanding of the physical mechanisms involved in Limit Cycle Oscillation (LCO) that can lead to a unified theory and analysis methodology. This work aims for a more thorough comprehension of the nature of the nonlinear aerodynamic interactions with structural dynamics for transonic LCO mechanisms, providing a significant building block in the understanding of the overall aero-elastic effects during the LCO. The aerodynamic and structural physics of the aero-elastic F-16 Typical LCO weapon configuration are represented numerically using full-order computational fluid dynamics (CFD) and modal Finite Element (FE) coupled simulations. Characterization of the flow-field is accomplished temporally via traditional flow visualization techniques, Lissajous, and wavelet analyses. Lissajous analyses provide key insight into the highly non-sinusoidal tracking of surface pressure with respect to aircraft motion. Additionally, wavelets are vital in identifying localized frequency differences at any point in time. Finally, once fluid-structure interaction (FSI) codes are capable of accurately modeling a true LCO mechanism, these techniques will be crucial in identifying and characterizing the underlying physics that would otherwise be missed by frequency-domain-based techniques that are currently relied upon. Integration of fluid and structure, necessity of small time-steps, and complex weapon shapes drive up the complexity of the simulations and the required resources, extending the simulation times and post-processing requirements. The US Air Force (USAF) requirement for numerous, simultaneous and quick-reaction solutions for a wide variety of stores and aircraft can only be accomplished through application of parallel high performance computing resources that meet the significant computational and memory demands associated with the certification computational environment.

1. Introduction

1.1 Limit Cycle Oscillation (LCO)

Limit cycle oscillation (LCO) is a sustained non-divergent periodic motion experienced by aircraft with configurations which often include stores. The external stores consist of any object carried on the aircraft, such as a missile, bomb, fuel tank, or pod. The primary concern with regard to LCO relates to the pilot's ability to carry out his mission safely and effectively. While experiencing LCO, the pilot may have difficulty reading cockpit gauges and the heads-up display (HUD), and an unexpected LCO condition can lead to premature termination of the mission. There are also concerns regarding the effects of LCO on the stores and UAVs. These issues include such things as potential degradation in reliability of components, whether or not stores can be safely released during LCO, the possible effects of LCO on target acquisition for smart weapons, and the effects of LCO on weapon accuracy for unguided weapons. Therefore, the prediction of LCO behavior of fighter aircraft is critical to mission safety and effectiveness, and the accurate determination of LCO onset speed and amplitude often requires flight test verification of linear flutter models. The increasing stores certification demands for new stores and configurations, the associated flight test costs, and the accompanying risk to pilots and equipment all necessitate the development of tools to enhance computation of LCO through modeling of the aerodynamic and structural sources of LCO nonlinearities.

Flutter, an instability caused by the aerodynamic forces coupling with the structural dynamics, and LCO are related phenomena. This association is due, in part, to the accuracy with which linear flutter models predict LCO frequencies and modal mechanisms.[1] However, since the characteristics of LCO motion are a result of nonlinear effects, flutter models do not accurately predict LCO onset speed and amplitude. Current engineering knowledge and theories are not sufficient to provide an analytical means for direct prediction of LCO. The current approach relies heavily on historical experience and interpretation of traditional flutter analyses and flight tests as they may correlate to the expected LCO characteristics for the configuration of concern.

1.2 Fluid-Structure Simulation Build-up Approach

There exists a significant need for a detailed understanding of the physical mechanisms involved in LCO that can lead to a unified theory and analysis methodology that is capable of predicting LCO responses. Many in the aero-elasticity community believe that LCO is indeed classical flutter at the onset, based on the strong similarity of the measured LCO deflection characteristics to linear flutter analysis computed deflections. However, the mechanism that bounds the oscillations still eludes the research community. Preliminary findings suggest that transonic aerodynamics is a potential bounding mechanism for typical LCO. Cunningham developed a semi-empirical method for LCO prediction based on his observation of "shock induced trailing-edge separation (SITES)" during LCO wind tunnel testing.[2–4] His work established the foundation that LCO is predominantly an aerodynamic phenomenon.

The present work aims to identify flow-field features, such as pressure gradients, flow separation, and shock interactions, using computational fluid dynamics (CFD) analyses at known transonic LCO conditions observed in flight tests. The results are used to assess how characteristics in the flow may be influencing the occurrence of LCO. Examination of a true fluid-structure interaction (FSI) LCO case (flexible structure coupled with CFD) is considered quasi-incrementally, since this capability does not yet exist in the flutter community. The first step is to perform fluid-structure reaction (FSR) simulations, examining the flow-field via CFD analyses on the F-16 wing during prescribed rigid-body pitch-and-roll oscillations, simulating the torsional and bending nature of an LCO mechanism.[10] Four configurations have been examined so far: clean-wing aircraft without tip missile launchers[5], clean-wing aircraft with tip launchers[6], wing-only with tip launchers[7], and one aircraft with under-wing stores.[11] The next steps are to perform FSR simulations of the flow-field during prescribed aero-elastic modes, and then graduate to fully-coupled FSI simulations.[24] Through this build-up fluid-structure simulation approach, insight is to be gained into the characteristics of the flow-field during LCO conditions in order to assess any possible influences on the LCO mechanism. It is hypothesized that these flow features interact with the structure in a way that limits the magnitude of an LCO mechanism, effectively bounding the divergent flutter.

Additionally, temporal data analysis techniques will be applied to the resulting time-accurate CFD data.[8–9] Traditional flow visualization techniques will be utilized in order to capture the overall flow-field features of interest, such as surface pressure variations, Mach=1 boundary and vorticity iso-surfaces, and flow separation. Lissajous analysis will be applied to the data in order to isolate the phase relationships in regions of interest identified via the flow visualization. Wavelet analysis is also applied at these regions, in order to accurately capture the nature of the frequency content of the mechanism.

2. Analysis Methodology

For the aero-elastic simulations, one or more predicted flutter modes will be prescribed using AERO suite of codes.[26–28] To determine the theoretical flutter-sensitive mode and its stability characteristics, linear flutter analyses are accomplished using a nominal worst-case aerodynamic condition with sea-level air density at 0.90 Mach number. The flutter analysis results for a flight-tested configuration are presented in Figure 2A and B, and show a flutter speed of 493 knots true airspeed (KTAS) at 8.45 Hz for the outboard wing torsion mode (Figure 2C), which is coupled with the aft wing bending mode (Figure 2D). The damping curve for the flutter mode shows a moderately steep slope and crosses the 1% damping level at 540 KTAS. The flutter analyses indicate the flutter critical mode to be moderately sensitive to changes in airspeed, and thus show good correlation to the flight-test data for velocity sensitivity, but there is no consistent correlation to the rate of oscillation amplitude increase. The predicted flutter frequency shows good correlation to the flight-test frequency, which ranges from 7.9 to 8.2 Hz.

The chosen validation case-study chosen for this work is the typical LCO configuration #1 (TLCO1) F-16C fighter aircraft, tail number 88-0441, due to the available flight test data for comparison.[23] This aircraft is a Block 40 F-16 modified for flutter testing. The aircraft carries a symmetric load of: wing-tip launchers at stations 1 and 9; under-wing launchers, pylon-wing interfaces, and AIM-9P missiles at stations 2 and 8; pylon-wing interfaces and air-to-ground missiles at stations 3 and 7; pylon-wing interfaces and 370-gallon empty fuel tanks at stations 4 and 6. Flight conditions chosen for

comparison are Mach numbers between 0.70 and 0.95 in increments of 0.05. Response amplitude and frequency data is available for accelerometers at stations 1 through 4. In level flight, measurable oscillations are present from 0.70 to 0.95 Mach, as shown in Figure 1A). At the 10,000-, 5,000-, and 2,000-ft test altitudes, the LCO amplitude increases somewhat exponentially with Mach number from initial onset up to 0.90 Mach. Above 0.90 Mach at all altitudes, the LCO amplitudes show a trend of leveling off and remained relatively constant. For elevated load factor flight, measurable oscillations are present from 0.75 to 0.95 Mach, as shown in Figure 1B). At all three test altitudes, the oscillation amplitudes increase almost linearly from 0.75 to 0.85 Mach, and show a significant increase in amplitude at 0.90 Mach. At 0.95 Mach for the 10,000- and 5,000-ft test altitudes, the LCO amplitude is lower than at 0.90 Mach. Elevated load factor flight is not investigated at 0.95 Mach, 2,000 ft. Overall, the dynamic aero-elastic characteristics of this configuration are well-behaved and can be categorized as typical LCO. The instability response is anti-symmetric and shows frequency variations from 7.9 to 8.2 Hz with the lowest frequencies seen at 0.90 Mach for all test altitudes.

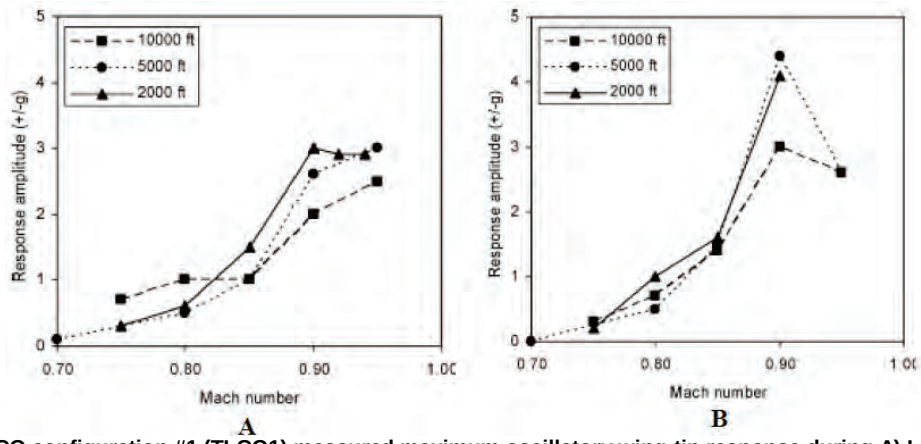


Figure 1. Typical LCO configuration #1 (TLCO1) measured maximum oscillatory wing-tip response during A) Level and B) Elevated Load Factor Flutter Flight Testing (Forward Vertical Accelerometer).[23]

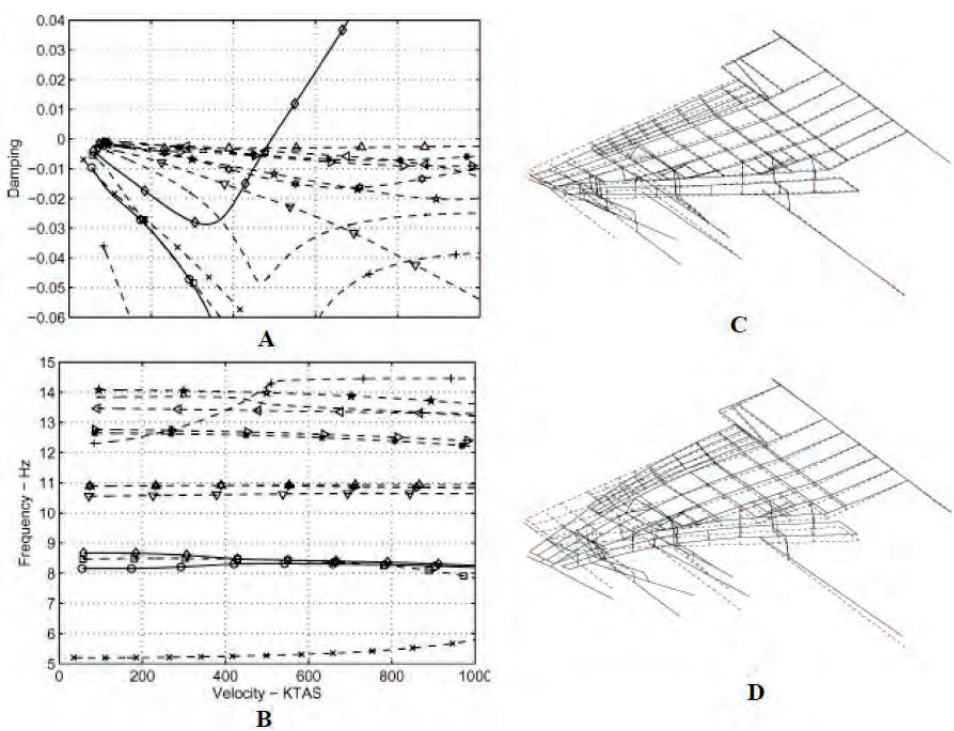


Figure 2. Flutter analyses at M_∞=0.9, sea level, V_f=493.6 KTAS, and f_f=8.45 Hz: A) Damping vs. Velocity and B) Frequency vs. Velocity plots; C) Linear flutter mechanism unstable mode, anti-symmetric out-board wing torsion, (right wing from full-span model) f_n=8.67 Hz; and D) Linear flutter mechanism coupled mode, anti-symmetric aft wing bending (right wing from full-span model) f_n=8.16 Hz.[23]

3. Analysis Tools

3.1 Flow Solvers

Rigid-body computations are performed using Cobalt, a commercial cell-centered, finite-volume CFD code. It solves the unsteady, three-dimensional, compressible RANS equations on hybrid unstructured grids. It is fundamentally based on Godunov's[13] first-order-accurate, exact Riemann solver. Second-order spatial accuracy is obtained through a Least-Squares Reconstruction. A Newton sub-iteration method is used in the solution of the system of equations to improve time accuracy of the point-implicit method. Arbitrary Lagrangian-Eulerian (ALE) formulation is used to perform grid movement, where the grid is neither stationary nor follows the fluid motion, but is reoriented without being deformed. The coordinate values change, but the relative positions between the grid-points are unchanged. As a result, terms such as cell volume and face area remain constant and equal to the values in the original grid. The motion can include both translation and rotation. Complex motions can be defined by specifying arbitrary rotations and displacements of the grid in a motion file. This file then forms part of the required input deck. The hybrid Reynolds-averaged Navier-Stokes (RANS)/Large-Eddy Simulation (LES), Delayed Detached-Eddy Simulation (DDES)[14] turbulence model based on the one-equation Spalart-Allmaras (SA)[15] is used for all calculations with and without Rotation and Curvature corrections (SARC).

Aero-elastic computations are performed using Kestrel and AERO suites. Kestrel is a cell-centered, finite- volume CFD code which solves the unsteady, three-dimensional (3D), compressible RANS equations on hybrid unstructured grids of arbitrary cell topology. The flow solver used in Kestrel is Air Vehicles Unstructured Solver (AVUS), formally known as Cobalt_{60}[12], an ancestor code of commercial Cobalt. AVUS is developed and maintained at the Air Force Research Laboratory at Wright Patterson Air Force Base, Ohio. Arbitrary Lagrangian-Eulerian formulation is used to perform grid movement, where the grid is neither stationary nor follows the fluid motion but is reoriented without being deformed. The coordinate values change, but the relative positions between the grid points are unchanged. As a result, terms such as cell volume and face area remain constant and equal to the values in the original grid. The motion can include both translation and rotation. Complex motions can be defined by specifying arbitrary rotations and displacements of the grid in a motion file. This file then forms part of the required input deck. Adaptive mesh refinement (AMR) techniques are also being incorporated into Kestrel, and will be utilized as they become available.

Kestrel is a research code being developed as part of the Department of Defense High Performance Computing Modernization Program (DoD HPCMP) Computational Research and Engineering Acquisition Tools and Environments (CREATE) Program[16–17]. The Kestrel software product is a modularized multi-disciplinary fixed-wing virtual aircraft simulation tool incorporating aerodynamics, structural dynamics, kinematics, and kinetics. The first version of Kestrel is targeted to subsonic, transonic, and supersonic flight conditions with three major capabilities of a single static grid simulation, a single grid rigid body motion simulation, and a deforming single grid aero-elastic simulation.[16] This initial capability couples a modal structural model with the core flow solver component resulting in an ability to perform time-marching simulations of elastic bodies. This basic capability serves as an incremental build-up to full elastic aircraft analyses, which will be included in Kestrel v2.0. The structural model component includes the necessary fluid-structure interaction capabilities to properly transfer computed surface loads down to the underlying structural nodes, as well as determine an updated surface mesh due to the structural response. A separate mesh deformation component is tasked with deforming the fluid mesh based on the updated surface mesh position.[19] Morton[18] has shown that aero-elastic Kestrel results of the AGARD 445.6 wing have compared favorably with experiment.[20] Due to the limited FSI flutter cases available for validation, these results provide a confidence metric for the Kestrel code's ability to capture the relevant flow-field physics for aero-elastic phenomena.

The AERO suite of codes consists of AERO-F for the fluids simulations and AERO-S for structural simulations.[21] AERO-F is a three-dimensional, domain decomposition based Euler/Navier-Stokes solver whose development started at the Center for Aerospace Structures at the University of Colorado at Boulder under the name AERO-F3D, to address high-performance compressible-flow, fluid-structure interaction, and aero-elastic computations. AERO-F operates on unstructured meshes that can combine tetrahedra, prisms, pyramids, and hexahedra. These meshes can move and/or deform in a prescribed manner (forced body oscillations), or be driven via interaction with the structural solver, AERO-S. For this reason, the governing equations are formulated in AERO-F in an ALE setting. AERO-F utilizes mesh motion solvers with an energy-conserving method for computing the flow-induced load on the structure on non-matching CFD and structural grids.[26] For turbulent flow computations, it offers one- and two-equation turbulence models, static and dynamic LES and Variational Multi-Scale LES (VMS-LES), as well as DES methods, with or without a wall-function. The spatial discretization adopted by AERO-F blends the finite-volume and finite-element methods. More specifically, this

semi-discretization combines a second-order-accurate Roe or HLLE upwind scheme for the convective fluxes, and a Galerkin centered approximation of the viscous fluxes. It can achieve fifth-order spatial dissipation error and sixth-order spatial dispersion error; and therefore, fifth-order spatial accuracy. Time integration in AERO-F can be performed with first- and second-order-implicit, and first-, second-, and fourth-order-explicit algorithms that satisfy their respective discrete geometric conservation laws (DGCLs). AERO-S is a finite-element modal and dynamic nonlinear capable solver. The AERO suite is also suitable for distributed computing on supercomputers utilizing thousands of processors.[27,28] The AERO suite has also been exercised with the aero-elastic AGARD 445.6 wing and compared favorably.[22]

3.2 Computational Models

For the rigid-body fluid-structure reaction (FSR) pitch oscillation case examined, a half-span viscous, rigid, full-scale model of the F-16, Grid8, is used for the tip-launcher-only case, modeling the fore-body bump, diverter, ventral fin, fuselage gun port, and leading-edge antennae. Grid8 consists of 3.18 million points (9.27 million cells). The engine duct is modeled and meshed up to the engine face. The boundary conditions for all computations are symmetry, adiabatic solid-wall for the surface of the aircraft and the engine duct, and modified Riemann invariants for the far-field boundaries. A source boundary condition based on Riemann invariants is used to create an inflow condition at the engine exhaust. A sink boundary condition is used at the engine face to model the corrected engine mass flow.

For the aero-elastic fluid-structure interaction case examined, a full-span Euler, flexible, full-scale model of the F-16, ADM1, is used for the tip-launcher-only case, modeling the fore-body bump, diverter, ventral fin, and fuselage gun port. ADM1 consists of 1.22 million points (6.63 million cells). Trim angle-of-attack (AoA) for the known flight test derived full-stores configuration is used for all of the simulations. Flight representative positioning of the control surfaces is not simulated by any of the aerodynamic models (ADMs). All of ADM control surface positions are in the local zero-angle arrangement for every simulation presented. Each of the grid parameters is outlined in Table 1.

Table 1. Aerodynamic computational grid parameters

Grid Name	Span	# points	# cells	Comments
Grid8	Half	3.18E+06	9.27E+06	Viscous, RB, Tip-only
ADM1	Full	1.22E+06	6.63E+06	Euler, FSI, Tip-only

The finite-element model used for this is comprised of a dynamically representative modal NASTRAN model of the typical LCO configuration #1 (TLCO1) structure with 8,532 degrees-of-freedom representing the aircraft structure, under-wing stores, pylons, and launchers. This model is used for store clearance analysis in the US Air Force SEEK Eagle Office at Eglin AFB, FL.[23] The 26 modes included in the TLCO1 model contain all wing, store, and empennage modes which have both symmetric and anti-symmetric characteristics. To guarantee accurate motion matching between the fluid and structure grids, the structural model is modified using satellite-rigid elements and phantom-shell elements. The mode shapes and frequencies are re-verified to confirm that nothing has changed in the structural model. The model frequency and mode shapes used in the simulation evaluation are found to be equal to the original model, as shown in Figure 2C and 2D.

4. Current Results

4.1 FSI Simulation Set-up

Each FSI simulation is begun at the flight representative trim angle-of-attack corresponding to the Mach number and altitude. Steady-state solutions are run to a residual of 10e-8 for each Mach number.[24] Each FSI simulation is then restarted using the corresponding Mach and altitude steady-state solution. FSI simulations are run at a physical time-step of 0.00025 seconds with a target of at least 15 seconds total simulation time in order to provide confidence that the simulated response is consistent with, and can be compared to, responses observed during flight tests. At the start of each new FSI simulation, the initial wing response is symmetric due to simulation-to-flight representative wing loading characteristics. In the initialization strategy, modes 1–24 are critically-damped, except modes 2 and 4 from Figure 2C and 2D, respectively; which are known to be the modes of interest from classical flutter analysis methods. The known anti-symmetric modes are expected to interact more closely, modeling the excitation performed during flight testing. After the initial symmetric response has transitioned into the desired anti-symmetric response, the simulation is restarted with no damping of any modes

for the remainder of the simulation time. This procedure is similar to the LCO flight testing process of using the critical flutter frequencies found with classical flutter analysis methods as input perturbations.

4.2 Wing-Tip Response Amplitude

The flow-field break-down study is extended by performing coupled CSD/CFD aero-elastic FSI simulations using the modal structural CSD model for typical LCO configuration #1 (TLCO1), and the clean-wing with tip-launcher F-16 Euler CFD fluid model (ADM1). The simulation time history wing-tip amplitude results, analogous to LCO flight test data, are presented for the total wing structural response when subjected to increasingly more realistic aerodynamic conditions at Mach 0.65 through 0.75 at 5,000 feet MSL altitude in Figure 3, Figure 4, and Figure 5. No evaluation of structural damping is addressed or presented in this paper.

Simulation data for the TLCO1/ADM1 clean-wing aerodynamic configuration at Mach 0.65, Figure 3, show that the response is damped and remains consistent at an anti-symmetric frequency of 8.75 Hz. These results are characteristic of the expected response below the LCO onset velocity. TLCO1/ADM1 clean-wing aerodynamic configuration data at Mach 0.70, Figure 4, shows the flight-test-expected neutrally-stable response, characterized as the LCO onset velocity prediction at 8.75Hz. Oscillations for the TLCO1/ADM1 clean-wing aerodynamic configuration at Mach 0.75, Figure 5, case is shown to slowly diverge at a dominant frequency of 8.75Hz. This trend of divergent flutter response continues at each subsequent increase in Mach number, diverging more rapidly at Mach 0.8, 0.85, 0.90, and 0.95. Using ADM1 aerodynamics, the Euler FSI simulations are able to predict the correct mode shape and onset speed, but do not capture the correct amplitude when compared to flight test data. Also, the ADM1 clean-wing with tip-launcher Euler aerodynamics did not show evidence of bounded oscillations above the onset speed.

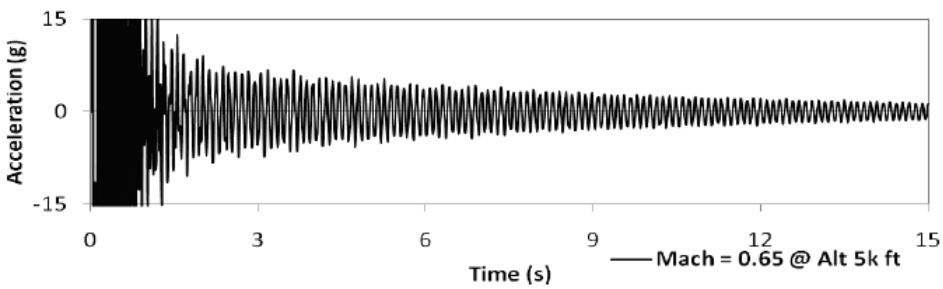


Figure 3. Typical LCO configuration #1 (TLCO1) structure with Euler clean-wing aerodynamics (ADM1) wing-tip response at Mach 0.65 altitude 5,000 ft: anti-symmetric 8.75 Hz

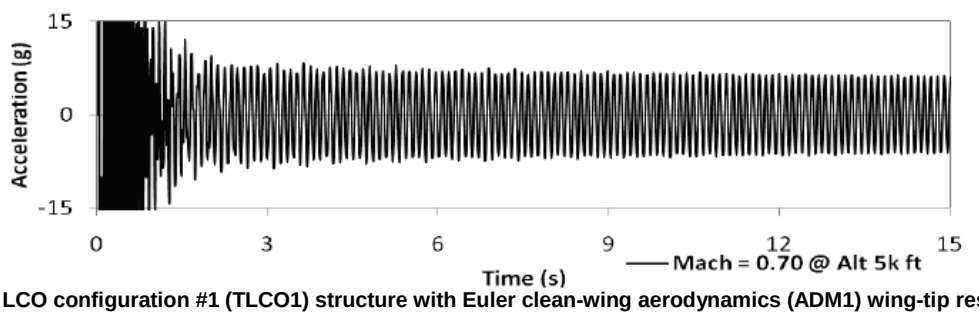


Figure 4. Typical LCO configuration #1 (TLCO1) structure with Euler clean-wing aerodynamics (ADM1) wing-tip response at Mach 0.70 altitude 5,000 ft: anti-symmetric 8.75 Hz

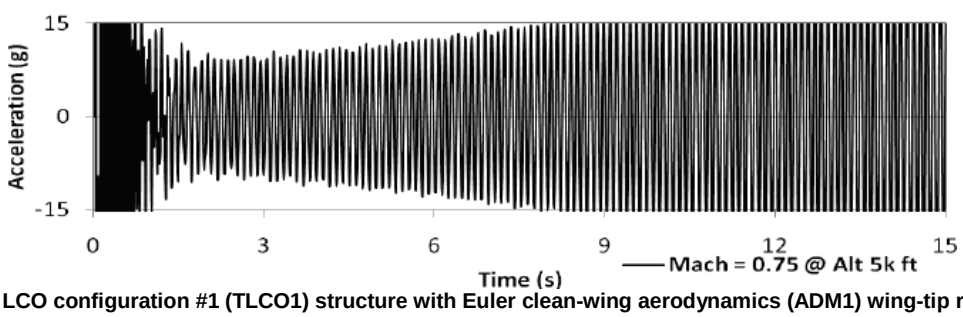


Figure 5. Typical LCO configuration #1 (TLCO1) structure with Euler clean-wing aerodynamics (ADM1) wing-tip response at Mach 0.75 altitude 5,000 ft: anti-symmetric 8.75 Hz

Additional Euler simulations modeling under-wing store aerodynamics coupled with the TLCO1 structural model have shown trends that result in significantly reducing the predicted amplitude of the response, while shifting the predicted onset velocity beyond the expected Mach 0.70 from flight test.[25] Therefore, addition of the under-wing store aerodynamics in the simulation may introduce aerodynamic damping. However, the Euler fluid equations may not be capturing the relevant nonlinear aerodynamics to the LCO bounding mechanism. The characterization of the flow-field physics, such as pressure distributions on the wing due to the presence of shocks/shock oscillations and flow separation/shock-induced flow separation must be conducted in order to discover whether these are contributing to the bounding of the LCO mechanism.

4.3 Flow Visualization

The coupled typical LCO configuration #1 (TLCO1) modal structural model and clean-wing with tip-launcher F-16 Euler CFD fluid model (ADM1) simulations are compared to the viscous, rigid-body, FSR clean-wing with tip- launcher (Grid8) results to evaluate the effects of flexibility and viscosity. For the Grid8 FSR simulations, an 8Hz ±0.5° rigid-body pitch oscillation is examined at trim angle-of-attack (AOAi) of 1.34° with the DDES-SA turbulence model at the same flow conditions (Mach=0.9, 5,000 feet) for comparison to the Euler TLCO1/ADM1 at trim of AOAi=1.4°.

Figure 6 illustrates the instantaneous Mach=1 boundary iso-surfaces colored by pressure. The top set of images display the viscous, FSR tip-launcher-only Grid8 case A) upper- and B) lower-surface results, and the TLCO1/ADM1 Euler FSI comparisons are presented on the bottom for the C) upper- and D) lower-surfaces. It can be seen from Figure 6 that including the effects of viscosity significantly influences the nature of the Mach=1 boundary iso-surface. It is observed from animation of Figure 6 that the trends between the viscous Grid8 FSR and Euler ADM1/TLCO1 FSI cases are similar; such as the Mach=1 iso-surfaces at their maximum sizes at the top of the oscillation, and that the aft extent of the upper-surface Mach=1 iso-surface maintains a mostly constant position on the aft portion of the wing, parallel to the trailing-edge (TE). The upper-surface profiles of the shocks are different with the viscous Grid8 FSR case being much larger (stronger).

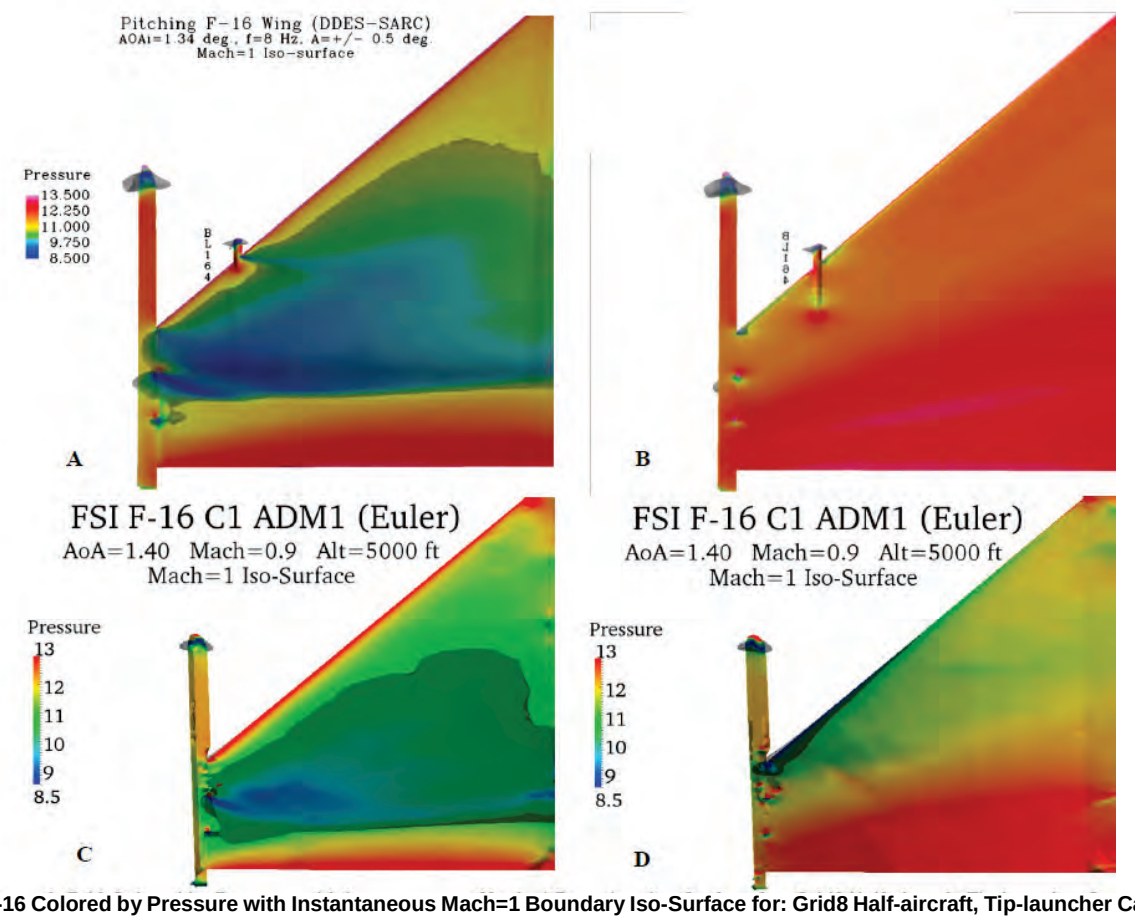


Figure 6. F-16 Colored by Pressure with Instantaneous Mach=1 Boundary Iso-Surface for: Grid8 Half-aircraft, Tip-launcher Case in 8 Hz Sinusoidal FSR Pitching Motion A) Upper- and B) Lower-Surfaces; and ADM1/TLCO1 Full-aircraft, Tip-launcher Case in Euler FSI Aero-elastic Motion C) Upper- and D) Lower-Surfaces.

Upon animation of the viscous Grid8 FSR case in Figure 6A and 6B, it is revealed that as the aircraft pitches, the Mach=1 iso-surface exhibits significant breathing motion on the upper-surface, and wraps around the tip-launcher at approximately half the way down the tip. The shock along the LE also wraps around the bottom surface of the wing. Upon animation of Figure 6C and 6D, the Euler ADM1/TLCO1 FSI case illustrates that as the aircraft oscillates, the Mach=1 iso-surface maintains a fairly constant position on the wing, with very little movement. Amplification of the motion also shows that the aerodynamic loading on the wing causes significant deflection of the wing, bending the tip launchers toward the fuselage.

The instantaneous C_p measurements plotted against non-dimensional chord are plotted in Figure 7 at 88% span on the left wing (looking aft-forward). The 88% span location (Butt Line 159) is chosen as the region of interest due to its vicinity to the under-wing missile launcher location. Results for the ADM1/TLCO1 FSI case are illustrated in B); and the comparison Grid8 FSR case results are shown in Figure 7A. For the Grid8 FSR case in Figure 7A, the C_p on both the upper and lower surface shows an overall larger magnitude of the shock due to the presence of viscosity. Upon examining Figure 7A and animating the C_p for all iterations on the upper-surface, a rapid suction build-up with increasing AoA is revealed at the 50–70% chord location. As AoA decreases, there is a rapid loss of suction which progresses from the aft portion of the wing forward. The motion of this loss resembles that of a double-hinged door slamming shut; the first near 70% chord and the second near 60% chord. This is an indication of the hysteresis behavior in the shock structure. Significant pressure differences occur on the lower-surface in the 0–10% chord region, with the lower surface having more suction than the upper surface and resulting in a loss of lift. For the ADM1/TLCO1 FSI case in Figure 7B, animation reveals that there is oscillatory behavior of the C_p on the upper surface in the 40% chord location ahead of the shock structure, as well as in the 60–70% location corresponding to the shock. However, the motion is much less oscillatory than that seen in the viscous, rigid-body case.

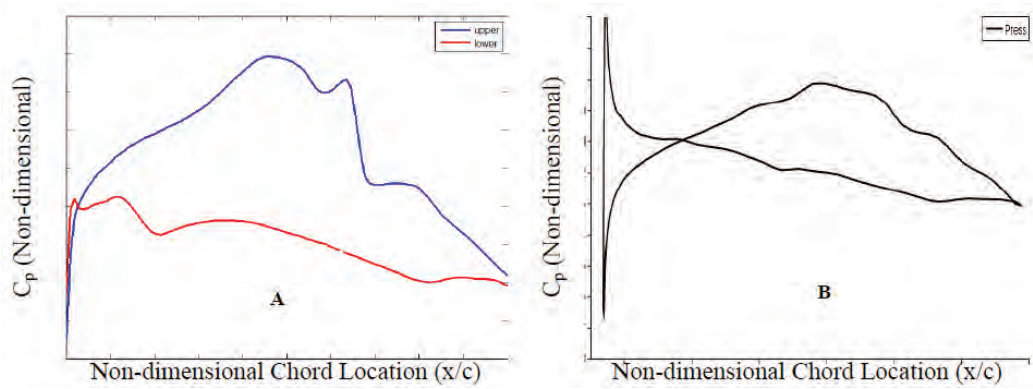


Figure 7. Instantaneous C_p Measurements at 88% Span for: A) Grid8 Half-aircraft, Tip-launcher Case in 8 Hz Sinusoidal FSR Pitching Motion; and B) ADM1/TLCO1 Full-aircraft, Tip-launcher Case in Euler FSI Aero-elastic Motion

5. Future Work

The current simulation work is part of the larger effort to quantify key fluid and/or structural physical features/interactions that are required to accurately model LCO and identify LCO bounding mechanisms. Multiple interacting mechanisms may make identifying each one separately difficult. This work will be extended to evaluate several other F-16 LCO configurations to further validate the simulation processes and verify that any sensitivities found are repeatable and consistent with other configurations. Additionally, future simulations will be conducted using viscous fluid-structure reaction (FSR) and fluid-structure interaction (FSI) simulations in order to fully-identify the added effects of viscosity on the LCO bounding mechanisms. Temporal analysis techniques will be implemented on the FSI and FSR data to identify nonlinear and non-periodic response behaviors.

6. Conclusion

The fundamental goal of this work is to gain a more thorough comprehension of the nature of the nonlinear aerodynamic effects for transonic limit cycle oscillation (LCO) mechanisms, providing a significant building block in the understanding of the overall aero-elastic effects during the LCO. The main benefits of successfully understanding the fundamental physics driving the mechanism are: 1) early discovery of complex aerodynamic phenomena that are typically only discovered in flight testing, and 2) rapid clearance of aircraft envelopes with new store combinations when aero-elastic phenomena are not

predicted. The chosen computational approach for solving these complex aircraft/weapons configurations is an unstructured grid, Euler, and Navier-Stokes CFD solver with RANS and DES turbulence treatments, finite-element or modal structural models, and computational grid deformation strategies. This high-fidelity approach will provide the ability to "virtually fly" missions before actual tests, as well as, provide a "measuring stick" for developing faster, lower-fidelity approaches suitable for screening hundreds of cases before analyzing the few most critical cases.

Unsteady fluid-structure reaction (FSR) CFD solutions of a viscous, rigid, full-scale F-16 clean-wing with tip-launchers configuration were computed and analyzed for prescribed rigid-body pitch oscillations, emulating the torsional component of an LCO motion. Unsteady, fluid-structure interaction (FSI) coupled CFD/CSD solutions of an Euler, flexible, full-scale F-16 clean-wing with tip-launchers (ADM1) aerodynamic configuration coupled with the modal structural model for the typical LCO configuration #1 (TLCO1) were computed and analyzed for comparison. Using ADM1 aerodynamics, the Euler FSI simulations are able to predict the correct mode shape and onset speed, but do not capture the correct amplitude when compared to flight test data. Also, the ADM1 clean-wing with tip-launcher Euler aerodynamics did not show evidence of bounded oscillations above the onset speed. Including the effects of viscosity significantly influences the nature of the Mach=1 boundary iso-surface, with the viscous Grid8 FSR case being much larger (stronger) and showing more oscillatory behavior. Additionally, surface pressure measurements reveal the hysteresis behavior in the shock structure for the viscous case not seen in the Euler simulations.

It is hypothesized that flow-field features leading to varying pressure distributions, shock locations and strengths, and vortical structures on the wing; and resulting in added time and flow inertia lags could be participating in bounding the LCO mechanism. Temporal analysis techniques are vital at uncovering the non-periodic behavior in the response. Once the state of FSI codes is capable of accurately modeling a true LCO mechanism, these techniques will be crucial in identifying the underlying physics that would be otherwise missed by frequency-domain-based techniques that are currently relied upon.

Acknowledgements

The authors would like to acknowledge the support of Dr. Michael Kendra, US Air Force Office of Scientific Research Test and Evaluation portfolio. The DoD HPC Modernization Program supported this project by supplying supercomputer time under the Computing Challenge Project C4P. This time was made available at the DoD Supercomputing Resource Centers at the US Air Force Research Laboratory (raptor) and the US Army Engineer Research and Development Center (diamond and garnet). We would like to thank the US Air Force Flight Test Center, Air Force SEEK EAGLE Office, and the CREATE-AV team for their support.

References

1. Bunton, R.W. and Denegri, C.M., "Limit Cycle Oscillation Characteristics of Fighter Aircraft." *Journal of Aircraft*, Vol. 37, No. 5, pp. 916–918, 2000.

2. Cunningham Jr., A.M., "A Generic Nonlinear Aeroelastic Method with Semi-empirical Nonlinear Unsteady Aerodynamics-Volume 1: Technical Discussion." *Report AFRL-VA-WP-TR-1999-3014*, US Air Force Research Laboratory, February 1999.

3. Cunningham Jr., A.M., "The Role of Nonlinear Aerodynamics in Fluid-Structure Interaction." *AIAA Paper 98-2423*, June 1998.

4. Cunningham Jr., A.M. and Meijer, J.J., "Semi-Empirical Unsteady Aerodynamics for Modelling Aircraft Limit Cycle Oscillations and Other Nonlinear Aeroelastic Problems." *Proceedings of the International Forum on Aeroelasticity and Structural Dynamics*, Royal Aeronautical Society, London, Vol. 2, pp. 74.1–74.14, 1995.

5. Pasiliao, C.L. and Dubben, J.A., "Analysis of CFD Pressure Coefficients on the F-16 Wing Associated with Limit Cycle Oscillations." *AIAA Paper 2008-0409*, January 2008.

6. Pasiliao, C.L. and Dubben, J.A., "Determination of Flow Field Characteristics of F-16 Wing Associated with Limit Cycle Oscillations Using CFD." *Proceedings of the NATO RTO Symposium AVT-152 on Limit-Cycle Oscillations and other Amplitude-Limited, Self-Excited Vibrations*, Norway, May 2008.

7. Pasiliao, C.L. and Dubben, J.A., "CFD Based Determination of Transonic Flow Field Characteristics of F-16 Wing Associated with LCO." *AIAA-2009-1084*, January 2009.

8. Pasiliao C.L., *Temporal Analysis of Transonic Flow Field Characteristics Associated with Limit Cycle Oscillations*, Ph.D. thesis, Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, May 2009.

9. Pasiliao, C.L. and Dubben, J.A., "Fluid-Structure Reaction Study of Flow Field Physics Associated with Limit Cycle Oscillations." *IFASD Paper 2009-136*, International Forum on Aeroelasticity and Structural Dynamics, Seattle, WA, June 2009.

10. Pasiliao, C.L. and Dubben, J.A., "Fluid-Structure Reaction Study Roadmap for an F-16 Wing Undergoing Prescribed Limit Cycle Oscillations." *Proceedings of the 2010 ITEA Aircraft/Stores Compatibility Symposium*, Fort Walton Beach, FL, April 2010.

11. Pasiliao, C.L. and Dubben, J.A., "Fluid-Structure Reaction Study of an F-16 Limit Cycle Oscillation Case with Stores." *AIAA 2010-2633*, 51st AIAA SDM Conference, Orlando, FL, April 2010.

12. Strang, W.Z., Tomaro, R.F., and Grismer, M.J., "The defining methods of Cobalt60: A parallel, implicit, unstructured Euler/Navier-Stokes Flow solver." *AIAA Paper 1999-0786*, January 1999.

13. Godunov, S.K., "A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics." *Matematicheskii Sbornik*, (47), pp. 357–393, 1959.

14. Spalart, P.R., Jou, W.H., Strelets, M., and Allmaras, S.R., "Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach." *Proceedings of the First AFOSR International Conference on DNS/LES*, Greyden Press, August 1997.

15. Shur, M.L., Strelets, M.K., Travin, A.K., and Spalart, P.R., "Turbulence modeling in rotating and curved channels: Assessing the Spalart-Shur correction." *AIAA Journal*, 38(5), May 2000.

16. Morton, S.A., McDaniel, D.R., Sears, D.R., Tillman, B., and Tuckey, T.R., "Kestrel – a fixed wing virtual aircraft product of the CREATE program." *AIAA Paper 2009-338*, January 2009.

17. Morton, S.A., McDaniel, D.R., Sears, D. R., Tillman, B., and Tuckey, T.R., "Kestrel v2 - 6DOF and control surface additions to a CREATE simulation tool." *AIAA Paper 2010-511*, January 2010.

18. Morton, S.A., McDaniel, D.R., Sears, D.R., Tillman, B., and Tuckey, T.R., "Rigid, maneuvering, and aeroelastic results for Kestrel - a CREATE simulation tool." *AIAA Paper 2010-1233*, January 2010.

19. McDaniel, D.R. and Morton, S.A., "Efficient mesh deformation for computational stability and control analyses on unstructured viscous meshes." *AIAA Paper 2009-1363*, January 2009.

20. Yates, C.E., "AGARD standard aeroelastic configurations for dynamic response. Candidate configuration I - Wing 445.6." *Technical report, NASA Technical Memorandum*, Hampton VA, 1987.

21. Geuzaine, P., Brown, G., Harris, C., and Farhat, C., "Aeroelastic dynamic analysis of a full F-16 configuration for various flight conditions." *AIAA Journal*, 41, pp. 363–371, 2003.

22. Lesoinne, M. and Farhat, C., "Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems." *AIAA Journal*, 36(9), pp. 1754–1756, 1998.

23. Denegri Jr., C.M., Dubben, J.A., and Maxwell, D.L., "In-Flight Wing Deformation Characteristics During Limit-Cycle Oscillations." *Journal of Aircraft*, Vol. 42, No. 2, pp. 500–508, 2005.

24. Lechniak, J.A. and Bhamidipati, K.K., "Fluid-Structure Interaction Evaluation of F-16 Limit Cycle Oscillations." *AIAA Paper 2012-42*, AIAA ASM Conference, Nashville, TN, Jan 2012.

25. Lechniak, J.A., Bhamidipati, K.K., and Pasiliao, C.L., "Characterizing the Aerodynamic Influence of F-16 Stores on Limit Cycle Oscillations using Aero-Structural Simulation." *AIAA T&E Days Conference*, New Orleans, LA, June 2012 (abstract accepted).

Application of CFD in Development of Large-Scale Scramjets (Challenge Project C4S Year 2)

John A. Boles
Taitech, Inc., Beavercreek, OH
john.boles2@wpafb.af.mil

Mark A. Hagenmaier
US Air Force Research Laboratory, Aerospace Propulsion Division (AFRL/RZAS), Wright-Patterson AFB, OH
mark.hagenmaierwpafb.af.mil

Abstract

Challenge Project C4S is focused on the use of computational fluid dynamics (CFD) to address challenges in the development of large-scale scramjets. In this second year of the Challenge Project, the technical effort has focused on the use of techniques that blend the best features of Large-Eddy Simulation (LES) and Reynolds-averaged Navier-Stokes (RANS) approaches for more accurate prediction of critical flow phenomena. In particular, details of two efforts will be presented. The first effort addresses the unsteady nature of flow in a scramjet isolator, while the second studies the effect of shock waves impinging on a fuel injector. Hybrid LES/RANS is shown to provide a more detailed and accurate understanding of the flow phenomena than steady-state RANS. The use of high performance computing-class resources is required to gain this insight into the complex scramjet flows.

1. Introduction

The supersonic combustion ramjet, also known as scramjet, is an efficient propulsion device for hypersonic flight. However, development of the scramjet has been slowed by technical challenges related to the complex physical phenomena outlined in Figure 1, by limitations in facilities to test the engines, and by limited success in using numerical modeling approaches for these engines. Despite these challenges, several recent successes[1,2] have been achieved.

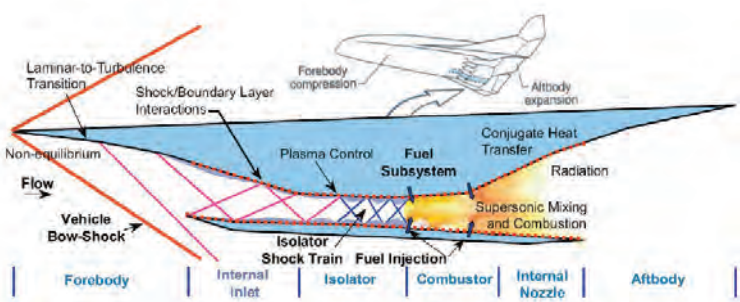


Figure 1. Idealized scramjet flowpath diagram (Image from NASA[3])

Ground testing of small-scale scramjets is only possible in a limited number of test facilities in the United States, namely the High Temperature Tunnel (HTT) at NASA Langley Research Center and the Aero Propulsion Test Unit (APTU) at Arnold Engineering Development Center. Air Force interest in larger-scale scramjets is challenging the limits of these facilities, and development of larger test facilities in the near future is unlikely.

Direct-connect testing are one technique to provide the capability to test larger-scale scramjets. In this approach, the engine inlet is not tested; rather, a nozzle is used to provide flow directly to the isolator and combustor at the appropriate conditions. However, the flow from the nozzle does not contain all of the shock waves that would be present from the inlet in flight.

Research under Challenge Project C4S is aimed at addressing some of the challenges that arise for developing large-scale scramjets. In the prior year, this team showed results of an effort to improve the capability of direct-connect testing, by using computational fluid dynamics (CFD) techniques in the design of wind tunnel hardware that can reproduce the shock structure from an inlet that is typically missing in the direct-connect test approach. Also in the prior year, validation studies of hybrid approaches using the Large-Eddy Simulation (LES) and Reynolds-averaged Navier-Stokes (RANS) methods were completed, showing the hybrid LES/RANS approach provided superior results for a scramjet fuel-air mixing problem.

In the current year, the validation of the hybrid LES/RANS approach was performed for the unsteady flow in a scramjet isolator, and for a more complex fuel-air mixing problem, where the influence of shock waves in the injector region was studied. These two studies from the current year are documented in Sections 3 and 4.

2. Numerical Methods

The REACT-MB code of North Carolina State University's CFD Lab[4] was used for the simulations shown in this work. A hybrid LES/RANS method[5] combining a Menter BSL RANS model near the wall with a Smagorinsky sub-grid model away from the wall was used. The transition from LES to RANS is determined by a blending function based on the ratio of the closest wall-distance to a modeled form of the Taylor microscale. An unsteady inflow condition was provided using a recycling and rescaling routine which is detailed in earlier work[5]. A standard Menter BSL RANS simulation was also performed. Inviscid fluxes are discretized using Edwards' LDFSS scheme[6] and viscous and diffusive fluxes used second-order central differences. The Piecewise Parabolic Method (PPM) of Colella and Woodward[7] was used to extend the LDFSS scheme to higher-order-accuracy. This method has proved successful in predicting Mach 2 cross-flow cases with air, helium and ethylene injection[8].

3. Isolator Study

The isolator is a nearly constant area duct between the inlet and the combustor in a scramjet. It is called an isolator because it keeps the pressure-rise caused by combustion from moving upstream and unstarting the inlet. This pressure-rise is typically in the form of a shock train, which is a series of successive compressions and expansions that result in pressure-rise. While there is often a large subsonic separated-region near the walls, the core flow may remain supersonic.

Steady-state RANS predictions[9,10] give the impression that these shock trains are stationary, though they typically are not[11]. High-frequency pressure measurements from an experiment have revealed large-scale oscillatory behavior of the shock trains at frequencies on the order of 100 Hz. During these oscillations, the shock trains can move as much as three duct heights. Other experiments have also confirmed this type of large-scale movement of the shock train[12,13,14,15,16,11,17]. Better understanding of the unsteady nature of shock trains within isolators can help in the design of control systems to mitigate unstart phenomena. In fact, monitoring of pressure within the isolator was used in an unstart prevention algorithm in the X-43A[18] and there is current work to further use unsteady CFD simulations to help with the development of feedback controllers for future scramjet platforms[19].

This work will focus on numerical predictions of the unsteady nature of the shock train in a back-pressured isolator. High-speed pressure measurements from a previously unpublished set of experiments from the 2"×6" isolator in Research Cell 19[20] at the US Air Force Research Laboratory at Wright-Patterson Air Force Base, Ohio, will be compared with numerical predictions of the same conditions.

3.1 Experimental Details

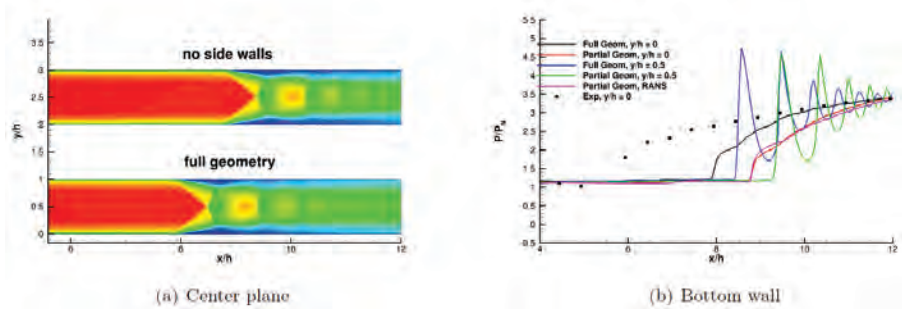
The 36" long, 2" by 6" constant-area Mach 2-inflow isolator set-up for Research Cell 19 was simulated in this work. Multiple back-pressures were achieved by adjusting a downstream mechanical valve. This study focuses on the back-pressure ratio that put the shock train in the middle of the isolator (P_b/P_N=3.56 where P_b is the pressure measured at the end of the isolator and P_N is the reference pressure measured just downstream of the facility nozzle). Pressure was measured by 27 static pressure ports along the centerline on the bottom wall and 16 on the top wall. Eight high-frequency Kulite pressure taps were used to measure span-wise pressure variations and were recorded at 10 kHz. A shadowgraph system was also used to take snapshots (exposure time of 200 μs) of the shock structure of the isolator.

3.2 Computational Results and Discussion

A hybrid LES/RANS simulation of a back-pressured 2"×6"×24" rectangular, constant-area isolator was performed. The computational simulation was performed with conditions to match an experimental case where the shock train was observed to be in the middle of the isolator. The experimental isolator was 36" long, so the computational exit boundary was set so as to match the static pressure at x=24" in the experiment. Statistics were recorded for the last 10.4 ms of physical time after transients associated with the initialization process were eliminated. Each millisecond of physical time took approximately 45 hours of computational time on 512 cores (23,040 CPU hours) on an SGI Altix cluster.

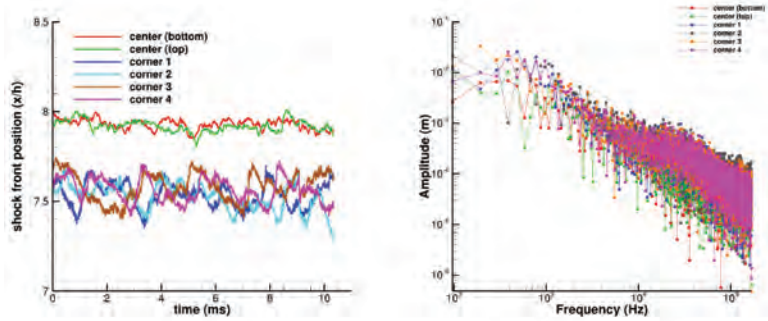
As a result of this high computational expense for the full-domain, a partial-domain hybrid LES/RANS simulation was also performed at the same conditions to determine if there were computationally less-expensive ways to get valuable unsteady data. This domain consisted of approximately the middle 28% of the six-inch width of the domain ($|z|\leq 0.847$"), with the full domain height and 24" length to match the previous case while using only a quarter of the grid-cells (34 million). Due to better computational efficiency for smaller problems, this simulation ran at more than double the speed on a quarter of the cores—leading to computing a millisecond of physical time in approximately 24 hours of computational time on 128 cores (3,128 CPU hours). Thus a reduction to a quarter of the domain resulted in computational savings of nearly 90%. This simulation will now be referred to as Case 2, and the full geometry simulation will be referred to as Case 1. Statistics were saved for the last 9.4 ms of physical time.

The boundary conditions at the span-wise maximum and minimum were treated as periodic for Case 2. This treatment gives the isolator an effective infinite-aspect ratio, due to the lack of any corners. Cox-Stouffer[21] showed that a two-dimensional (2D) simulation of an isolator (also effectively an infinite-aspect ratio) has the effect of moving the shock train downstream of its position with a finite-aspect ratio. The theory presented in that work is that the corner and sidewall influence on shock train position is critical and is the main driver of the shock train's movement upstream. This same effect held for the no sidewall simulation performed here. In Figures 2(a) and 2(b), it can be seen that the shock train for Case 2 is downstream of its full-width counterpart. It can also be seen from Figure 2(b) that the LES/RANS simulations did not accurately capture the shock train location seen in the experiment. This discrepancy could be due to the shortening of the computational domain, as well as the method of the imposition of back-pressure. In these simulations, the back-pressure is imposed directly in the subsonic portion of the exit plane. In the experiment; however, this pressure is imposed with a valve downstream of a large subsonic chamber past the 36" isolator. Figure 2(b) also shows the centerline (y/h=0.5) pressure profiles for Cases 1 and 2. This profile shows that the pressure changes are much more abrupt outside of the boundary-layer, as they are due to a series of normal shocks and expansions as opposed to the steady-compression seen at the wall.



(a) Center plane (b) Bottom wall

Figure 2. Static pressure for Case 1 and Case 2 compared with bottom wall experimental data



(a) Time history (b) FFT of time history

Figure 3. Shock-front tracking for Case 1

Figure 3 shows a shock-front tracking plot for the last 10.4 ms of computation—the time for which statistics were recorded for Case 1. The shock-front position is defined as the farthest upstream the pressure surpasses 1.25 times the inflow wall-pressure. Oscillations in shock position have larger-amplitude at the corners than at the center. This is to be expected because the larger separated region associated with the corners causes these larger oscillations. Fourier analysis of the shock-front tracking shows a peak in all but the "corner 2" profile at 389 Hz, and all the profiles have a peak somewhere between 389 and 600 Hz. This frequency range is consistent with the frequency associated with the approximate length of the shock train in the time-average and the average "U-a" eigenvalue for the subsonic region of the flow, that is:

$$\lambda = \frac{\left(\overline{U-a}\right)_{sub}}{\overline{L_{ST}}} \tag{1}$$

For Case 1, the eigenvalue was found to have a magnitude of 130 m/s and the average length of the shock train at the bottom (as seen in the time-averaged centerline wall-pressure in Figure 2(b)) is approximately four duct heights (0.2032 m) giving an expected frequency of 640 Hz. For Case 2, the frequency was found to be 914 Hz—due to the shock train being further downstream, and the eigenvalue being slightly larger in magnitude than Case 1. It is expected that with more time over which Fourier analysis can be performed, the dominant low-frequencies can be better determined.

Though shock train position tracking seems a natural fit for analyzing frequency content associated with unsteady shock train, experimentally it is not feasible to measure conditions at a high enough frequency at the number of locations needed to accurately track shock position. In this case, high-speed pressure data was measured with Kulite transducers along the bottom wall at four locations close to the centerline, and four locations off-center at a frequency of 10 kHz for duration of one second. Fourier analysis of the experimental data did not show any obvious dominant frequencies in Figure 4(a), though the amplitude peaks at approximately 7 Hz for the transducer near the shock train location, as seen in Figure 4(b), and high-amplitude low-frequency content exists between 1 and 95 Hz. Computational time histories of pressure were recorded as well, and can be seen in Figures 5(a) and 5(b). The time histories were taken at locations where the standard deviation of pressure peaked along the bottom wall (shown in green) and centerline (shown in red) and just upstream of the peak on the bottom wall (shown in black).

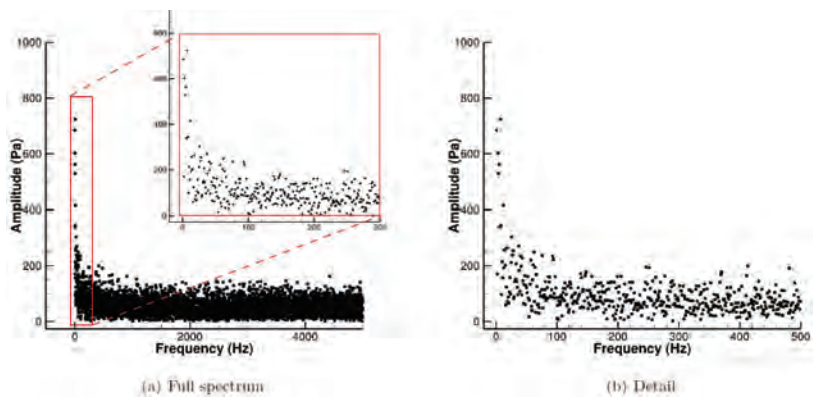


Figure 4. Fourier analysis of pressure readings for the experiment near the shock train

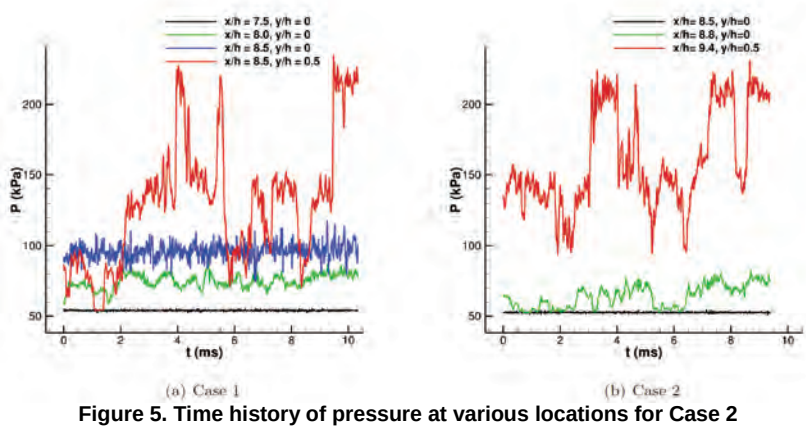


Figure 5. Time history of pressure at various locations for Case 2

Ganapathisubramani[22] and others[23] have shown that long coherent structures exist in flat plate turbulent boundary-layers. Figures 6(a) and 6(b) show streamwise velocity contours at similar wall-distances to the Particle Image Velocimetry (PIV) data shown in Ganapathisubramani's work. At y/δ_{in}=0.15 (δ_{in} is the boundary-layer thickness based on velocity at the beginning of the isolator), the coherent structures are evident in elongated regions of high- and low-momentum flow, and their length is approximately 10 boundary-layer thicknesses. The length of the structures further away from the wall cannot be as easily characterized from visual inspection. Ganapathisubramani showed that the structure length increased in the outer-part of the boundary-layer using statistical analysis. At this time, this statistical analysis has not been done for the computational simulations because of inadequate data collection for snapshots of constant y planes in the boundary-layer. If one uses this super-structure length and convective velocity (80% of free-stream velocity) to determine a characteristic frequency of the boundary-layer associated with the superstructures, it would be approximately 8 kHz for this case.

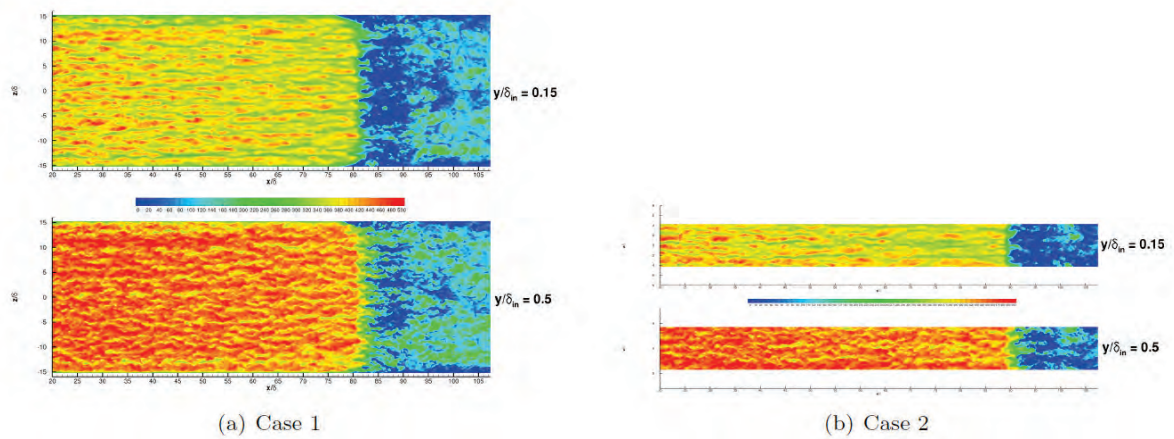


Figure 6. Snapshots of contours of streamwise velocity at constant y slices

4. Injector Study

In the continuing effort to better understand hypersonic flow phenomena, mixing characteristics of various fuels and fueling strategies are a critical part. RANS simulations have proved to be adequate for predicting penetration heights, but have not typically been as successful at predicting lateral spreading, vortical structures within plumes, or mixing within the core of injectant plumes[24,8,25]. LES and direct-numerical-simulation (DNS) methods have shown success in predicting many turbulent flows, but are very expensive computationally for wall-bounded high-Reynolds number flows, as the turbulent scales that need to be resolved by the grid become very small. Hybrid methods[26,27,28] have been developed to use LES for the majority of the flow-field and use RANS near solid surfaces, where the turbulent length-scales are small. Another aspect of scramjet physics that needs further investigation is the role of flow-distortion caused by oblique shock from the cowl in real scramjet flows (such as that in Figure 1). In typical direct connect wind tunnel facilities, the flow is basically uniform coming out of the facility nozzle—completely missing the impact of this cowl shock. Studies have shown[29,30] that these shocks could have significant impact on mixing and plume structures. This work will discuss the impact of an oblique shock impinging downstream of a normal injector. This will be accomplished through the use of hybrid LES/RANS and comparing it to new experimental observations.

4.1 Experimental Details

An experimental investigation of the impact of oblique shock waves on normal jet injection of air has been conducted to complement the computational studies being performed. The tunnel conditions and jet conditions will mimic those used in prior studies[31,32] of normal jet injection. A single 3/16" jet will be located in the bottom wall of the tunnel, approximately 5.9" from the end of the Mach 2 facility nozzle of Research Cell 19 at the US Air Force Research Laboratory[20]. On the upper-wall, a beveled plate was mounted, which can be translated axially. The leading bevel will turn the flow by either 5 or 7 degrees, before turning back parallel to the lower-wall. The shock generator thickness is 3/8". During calibration runs, a shadowgraph system was used to verify the position of the shock relative to the injector.

Following the calibration runs, the shadowgraph system was replaced by an nitrous-oxide-planar laser-induced fluorescence (NO-PLIF) system. NO-PLIF images were obtained on span-wise planes located at 0, 2.5, 5, 10, 15, 20, and 25 jet diameters downstream of the injector. The NO-PLIF imagery has the advantage of providing snapshots of fluorescence (which is a good analog for injectant mass fraction for limited temperature ranges such as this). This imagery provides detailed information on the normal and lateral spreading characteristics of the ethylene jet.

4.2 Computational Results and Discussion

Hybrid LES/RANS and RANS simulations of normal sonic ethylene injection with an oblique impinging shock downstream of the injector have been performed. The theoretical waves formed by the 5-degree shock generator can be found in Figure 7. As can be seen, a pair of shocks and expansion fans emanate from the shock generator. As discussed in Section 2, only the oblique shock is imposed in the numerical investigations and no expansion waves are introduced. These waves only impact the injection plume downstream of most of the region of interest, and it is thought that they would have minimal impact on mixing. The oblique shock from the 5-degree shock generator interacts with the injection plume between the x/D=2.5 and x/D=5 data collection planes, and strikes the bottom wall at approximately x/D=8.4. The oblique shock caused by the 7-degree shock generator strikes the bottom wall at approximately x/D=5.25.

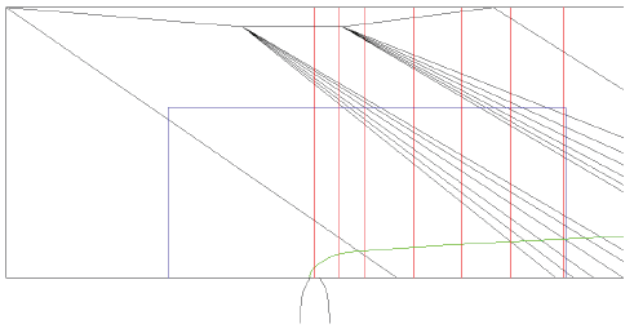


Figure 7. Theoretical diagram of shocks and expansion fans due to the shock generator (interactions not shown). The vertical red lines indicate data planes at which cross-plane NO-PLIF data was gathered experimentally and compared with CFD results in this work. The blue rectangle indicates the computational domain in X and Y.

RANS simulations for a dynamic pressure ratio of 1.5 have been performed for injection with no shock generator, and then with the 5- and 7-degree shock generator as described above. Center plane Mach number contours for these simulations can be seen in Figure 8(a). The red contour line denotes zero axial velocity and implies a significantly larger boundary-layer separation for the 7-degree shock generator case than the other two simulations.

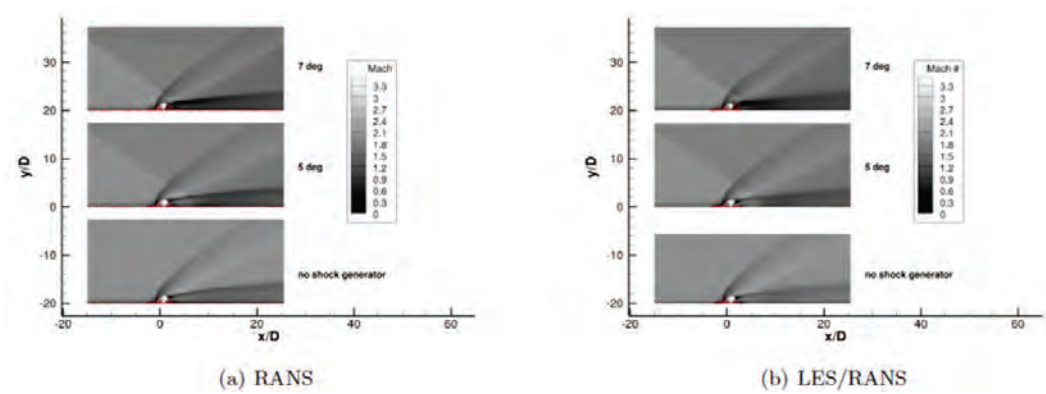


Figure 8. Comparison of Mach number contours for centerline without shock generator and with 5- and 7-degree shock generators using (a) RANS and (b) LES/RANS. Zero axial velocity line shown in red.

Hybrid LES/RANS simulations were performed to better capture the unsteady mixing and the impact of the shock impingement. Time-averaged center plane contours of Mach number can be seen in Figure 8(b). The contours look similar to the RANS contours, which are consistent with other mixing simulations in that the center plane flow characteristics are well-predicted by RANS, but off-center characteristics are not. Mixing within the fuel plume shows the biggest advantage

of LES/RANS over RANS methods. Figure 9 shows the same plane with cross-flow species density contours so that the wave structure, as well as the injection plume, can be discerned. The sonic line is shown in red and reveals that nearly all of the plume is subsonic in the case with the 7-degree shock generator. This confirms that there is indeed more separation in the 7-degree shock generator case that than the 5-degree case. Since there was limited wind tunnel time, these simulation results were used as evidence that only the 5-degree shock generator should be used, as the 7-degree shock generator introduces too much separation on the bottom wall.

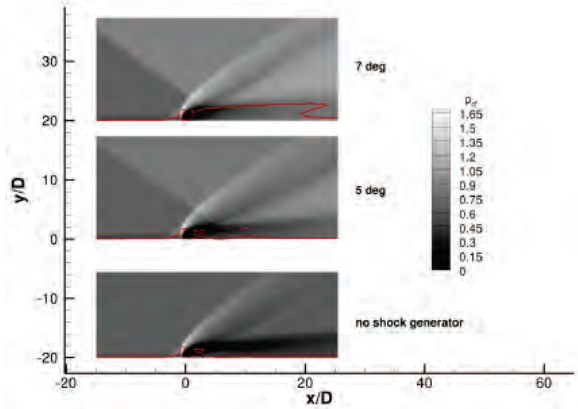


Figure 9. Comparison of cross-flow air species density contours for center-line without shock generator and with 5- and 7-degree shock generators using LES/RANS (Sonic line shown in red)

Side-view high-speed shadowgraph imagery was collected and the time-average of the q=1.5 case can be seen in Figure 10(a). Since the 5-degree generator was used in the experiment, the oblique shock wave and expansion wave interact with the injection plume further downstream than in the 7-degree theoretical diagram shown in Figure 7. It is also notable that the upstream expansion wave hardly interacts with the region of interest. The view outlined in red in Figure 10(a) is recreated numerically in Figure 10(b) by the LES/RANS simulation. The major flow features are all simulated accurately, including the separation and bow shocks upstream of the fuel injection. The experiment shows a shock that is reflected off of the bottom wall that does not appear in the CFD. This appears because the experimental shadowgraph is formed by the density changes across the entire width of the tunnel, whereas the numerical shadowgraph is only determined by the center plane. In the center plane the reflected shock occurs away from the wall because of the injection plume.

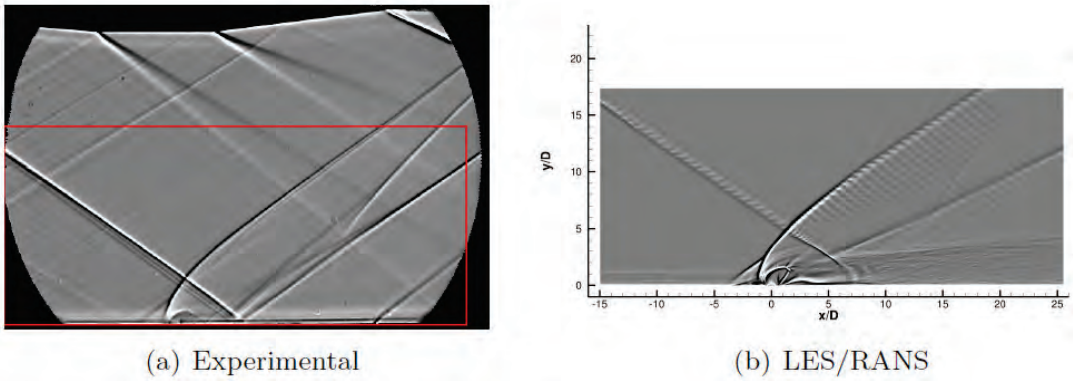
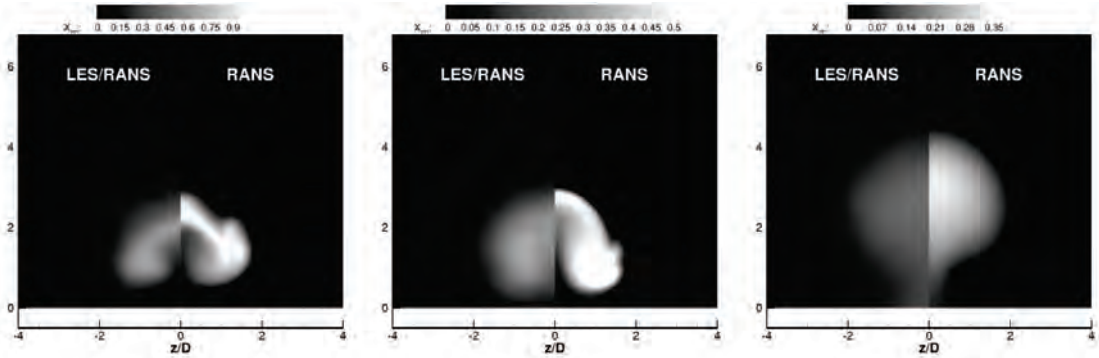


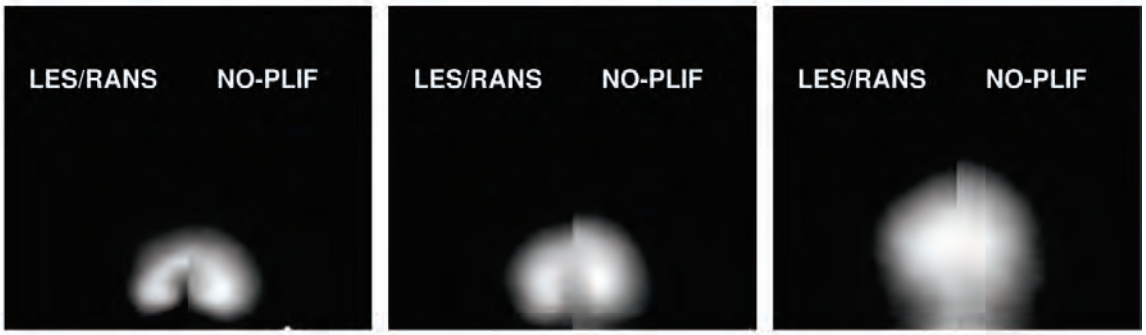
Figure 10. Time-averaged shadowgraph for q=1.5 ethylene injection with 5-degree shock generator

Earlier studies[8,33] have shown that the REACT-MB LES/RANS model is able to capture the shape and mixing within plumes better than RANS simulations. A comparison of RANS to LES/RANS simulations can be seen in Figure 11. The RANS and LES/RANS simulations look strikingly different in their shape and mixing, though the height of the plumes and separation from the bottom wall is similar. The LES/RANS simulation looks very similar to the experimental NO-PLIF imagery seen in Figure 12. The standard deviation imagery seen in Figure 13 again shows that the LES/RANS method did well to simulate the unsteady motion of the plume.



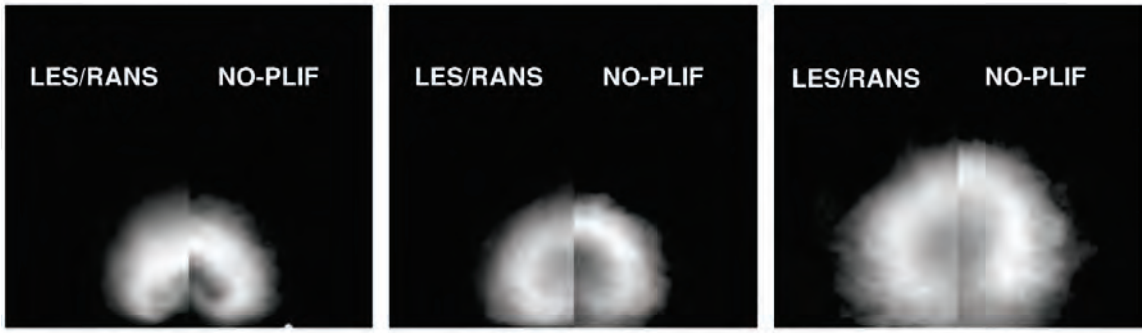
(a) x/D=5 (b) x/D=10 (c) x/D=25

Figure 11. Time-averaged LES/RANS and RANS cross-plane contours of injectant mass fraction for q=1.5, 5-degree shock generator case



(a) x/D = 5 (b) x/D = 10 (c) x/D = 25

Figure 12. LES/RANS time-averaged cross-plane contours of injectant mass fraction compared with NO-PLIF time-average for 5-degree shock generator case



(a) x/D = 5 (b) x/D = 10 (c) x/D = 25

Figure 13. Standard deviation of LES/RANS cross-plane contours of injectant mass fraction compared with the standard deviation of NO-PLIF imagery for 5-degree shock generator case

In order to gauge the impact of the shock generator on the mixing of the flow, Figure 14 shows contours of time-averaged injectant mass fraction with and without the 5-degree shock generator using LES/RANS. Downstream of the shock impingement (x/D=10 and x/D=25 stations in this case), the injection plume is closer to the wall than the undistorted case. There is also more mixing in the simulation with the shock generator than without. Both of these impacts are predictable because the flow is slowing down allowing for more time for mixing, as well as the pressure increase from the shock is keeping the plume closer to the wall. The simulation of the 7-degree shock generator case is compared to the 5-degree shock generator case in Figure 15. This shows that the 7-degree shock generator causes substantially more mixing than the 5-degree case. This falls in line with the reasoning that a stronger shock slows the flow down more, allowing for more mixing. However, the plume itself is further from the wall than the 5-degree case even though the pressure jump from oblique shock would be greater for the 7-degree shock generator. It is theorized that this is due to the large separation seen in Figures 8(b) and 9.

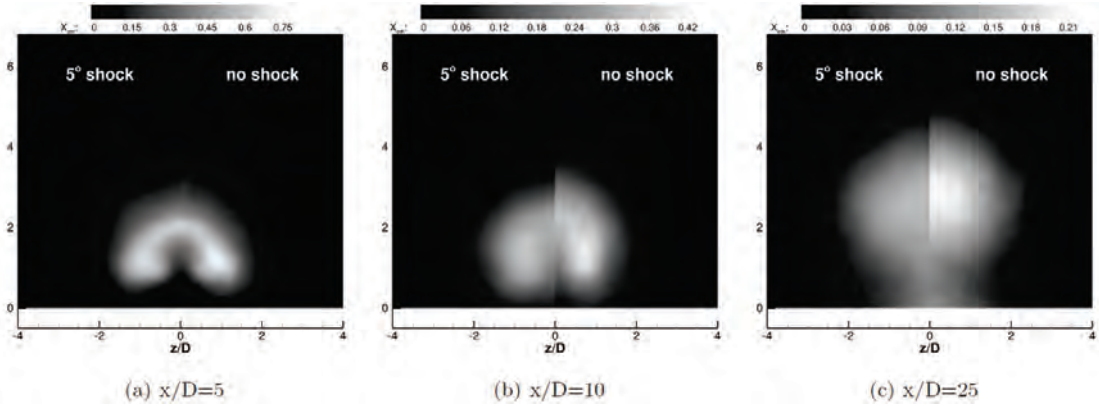


Figure 14. Time-averaged cross-plane contours of mass fraction with a 5-degree shock generator and no shock generator

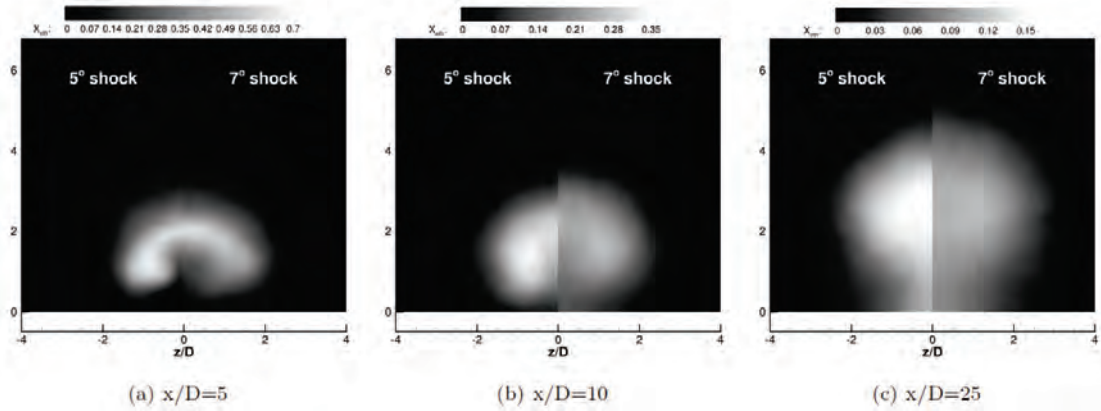


Figure 15. Time-averaged cross-plane contours of injectant mass fraction with 5-and 7-degree shock generator

5. Conclusion

In one study, two hybrid LES/RANS simulations of a back-pressured isolator were performed – a four-wall, full-width case and a case with just the partial-width and only top and bottom walls. The partial-width simulation showed similarities in unsteady behavior to the full-geometry condition. Neither case matched the shock train position observed in the experiment. While the partial width predicted a shock train downstream of the full-width simulation (and thus farther from the experimental observation), the reduced grid-size reduced the computational time to just over 10% of the full-geometry with a grid the quarter of the full-size grid while maintaining the unsteady behavior of the full-geometry. Long coherent structures in the turbulent boundary-layer similar to those seen in similar experimental studies were predicted upstream of the shock train. The frequency associated with the super-structure length was shown to be similar to a frequency peak in the Fourier transform of wall-pressure upstream of the shock train.

In a second study, hybrid LES/RANS simulations of sonic injection of ethylene into a Mach 2 cross-flow have been compared with new experimental NO-PLIF and high-frequency shadowgraph imagery of air injection with inflow distortion. The impact of shock generators on mixing has been shown. The LES/RANS model is shown to be very capable of predicting the mixing and the impact of the 5-degree shock generator on the mixing of the injectant. A 7-degree shock generator configuration was also simulated, though that has not been tested experimentally yet. This configuration shows even more mixing than the 5-degree generator due to further slowing of the flow. Unfortunately this shock causes a large separation in the wall boundary-layer, as well as a significant drop in total pressure.

Acknowledgments

This work was funded by the Air Force Office of Scientific Research (AFOSR) (Chiping Li, program manager) under Contract FA8650-08-D-2844. The DoD HPC Modernization Program supported this project by supplying supercomputer time under the Computing Challenge Project C4S. Most calculations were performed at the DoD Supercomputing

Resource Center (DSRC) at ERDC on the SGI Altix ICE 8200 (Diamond), and at the ARL DSRC on the SGI Altix ICE 8200 (Harold). We would like to thank the staff at the Consolidated Customer Assistance Center (CCAC) who provided assistance in optimal use of these resources. We would also like to thank Cam Carter and Mark Hsu for support with the experimental measurements and their processing.

References

1. http://www.af.mil/news/story.asp?id=123206525.

2. http://www.wpafb.af.mil/news/story print.asp?id=123301289.

3. "NASA Hyper-X Program Demonstrates Scramjet Technologies." *FS-2006-01-118-LaRC*.

4. http://legacy.mae.ncsu.edu/research/cfd lab/index.html.

5. Boles, J.A., Choi, J.-I., Edwards, J.R., and Baurle, R.A., "Multi-Wall Recycling/Rescaling Method for Inflow Turbulence Generation." *AIAA Paper 2010–1099*, Orlando, FL, January 2010.

6. Edwards, J.R., "A Low-Diffusion Flux-Splitting Scheme for Navier-Stokes Calculations." *Computers & Fluids*, Vol. 26, No. 6, pp. 635–659, 1997.

7. Colella, P. and Woodward, P.R., "The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations." *Journal of Computational Physics*, Vol. 54, pp. 174–201, 1984.

8. Boles, J.A., Edwards, J.R., and Baurle, R.A., "Large-Eddy/Reynolds-Averaged Navier-Stokes Simulations of Sonic Injection into Mach 2 Crossflow." *AIAA Journal*, Vol. 48, No. 7, pp. 1444–1456, July 2010.

9. Glasby, R.S., *A Numerical Investigation of the Influence of Cross-Section Geometry on Scramjet Isolator Performance*, Master's thesis, The Ohio State University, 2006.

10. Milligan, R.T., Eklund, D.R., Wolff, J.M., Gruber, M., and Mathur, T., "Dual Mode Scramjet Combustor: Analysis of Two Configurations." *AIAA Paper 2010–751*, Orlando, FL, January 2010.

11. Ikui, T., Matsuo, K., Nagai, M., and Honjo, M., "Oscillation Phenomena of Pseudo-Shock Waves." *Bulletin of the Japan Society of Mechanical Engineers*, Vol. 17, No. 112, pp. 1278–1285, October 1974.

12. Brown, M.S., Barone, D.L., Barhost, T.F., Eklund, D., Gruber, M., Mathur, T., and Milligan, R., "TDLAS-Based Measurements of Temperature, Pressure, and Velocity in the Isolator of an Axisymmetric Scramjet." *AIAA Paper 2010–6989*, Nashville, TN, July 2010.

13. Tan, H.-J., Sun, S., and Yin, Z.-L., "Oscillatory Flows of Rectangular Hypersonic Inlet Unstart Caused by Downstream Mass-Flow Choking." *Journal of Propulsion and Power*, Vol. 25, No. 1, pp. 138–147, 2009.

14. Wagner, J.L., *Experimental Studies of Unstart Dynamics in Inlet/Isolator Configurations in a Mach 5 Flow*, Ph.D. thesis, The University of Texas at Austin, May 2009.

15. Le, D.B., Goyne, C.P., Krauss, R.H., and McDaniel, J.C., "Experimental Study of a Dual-Mode Scramjet Isolator." *Journal of Propulsion and Power*, Vol. 24, No. 5, pp. 1050–1057, 2008.

16. Parrott, T.L., Jones, M.G., and Thurlow, E.M., "Unsteady Pressure ns also include the further analysis of the pressure sensitive paint data to include standard deviation and energy spectrum data in order to help evaluate Loads in a Generic High-Speed Engine Model." *Technical Paper 3189*, NASA, December 1992.

17. Lindstrom, C.D., Davis, D., Williams, S., and Tam, C.-J., "Shock Train Structure Resolved with Absorption Spectroscopy Part 2: Analysis and CFD Comparison." *AIAA Journal*, Vol. 47, No. 10, pp. 2379–2390, October 2009.

18. Jones, P.T. and Baumann, E., "Evaluation of the X-43A Scramjet Engine Controller Performance by Monte Carlo Technique." *AIAA Paper 2003–5192*, Huntsville, AL, July 2003.

19. Hoeger, T.C., King, P.I., Decker, D.D., Donbar, J.M., and Cox-Stouffer, S., "2-D Transient CFD Model of an Isolator Shock Train." *AIAA Paper 2011–2221*, San Francisco, CA, April 2011.

20. Gruber, M.R. and Nejad, A.S., "New Supersonic Combustion Research Facility." *Journal of Propulsion and Power*, Vol. 11, No. 5, pp. 1080–1083, 1995.

21. Cox-Stouffer, S.K. and Hagenmaier, M.A., "The Effect of Aspect Ratio on Isolator Peformance." *AIAA Paper 2001-0519*, Reno, NV, January 2001.

22. Ganapathisubramani, B., Clemens, N.T., and Dolling, D.S., "Large-scale motions in a supersonic turbulent boundary layer." *Journal of Fluid Mechanics*, Vol. 556, pp. 271–282, 2006.

23. Kim, K.C. and Adrian, R.J., "Very Large-Scale Motion in the Outer Layer." *Physics of Fluids*, Vol. 11, No. 2, pp. 417–422, 1999.

24. Palekar, A., Truman, C.R., and Vorobieff, P., "Prediction of Transverse Injection of a Sonic Jet in Supersonic Crossflow." *AIAA Paper 2005-5366*, Toronto, Ontario, June 2005.

25. Maddalena, L., Campioli, T.L., and Schetz, J.A., "Experimental and Computational Investigation of Light-Gas Injectors in Mach 4.0 Crossflow." *Journal of Propulsion and Power*, Vol. 22, No. 5, pp. 1027–1038, 2006.

26. Spalart, P.R., Jou, W.-H., Strelets, M., and Allmaras, S.R., "Comments on the Feasibility of LES for Wings and of a Hybrid RANS/LES Approach," *Advances in DNS/LES*, edited by C. Liu and Z. Liu, Greyden Press, Dayton, OH, pp. 137–147, 1998.

27. Spalart, P.R., Deck, S., Shur, M.L., Squires, K.D., Strelets, M., and Travin, A., "A New Version of Detached-Eddy Simulation, Resistant to Ambiguous Grid Densities." *Theoretical and Computational Fluid Dynamics*, Vol. 20, No. 3, pp. 181–195, May 2006.

28. Hassan, E., Aono, H., Boles, J., Davis, D., and Shyy, W., "Multi-Scale Turbulence Model Simulation of Supersonic Crossflow." *AIAA Paper 2010–111*, Orlando, FL, January 2011.

29. Schetz, J.A., Maddalena, L., and Burger, S.K., "Molecular Weight and Shock-Wave Effects on Transverse Injection in Supersonic Flow." *Journal of Propulsion and Power*, Vol. 26, No. 5, pp. 1102–1113, September 2010.

30. Campioli, T.L., Maddalena, L., and Schetz, J.A., "Studies of Shock Wave/Transverse Injection Interaction on Supersonic Mixing Processes." *AIAA Paper 2006–8135*, Canberra, Australia, July 2010.

31. Lin, K.-C., Ryan, M., Carter, C., Gruber, M., and Raffoul, C., "Raman Scattering Measurements of Gaseous Ethylene Jets in Mach 2 Supersonic Crossflow." *Journal of Propulsion and Power*, Vol. 26, No. 3, pp. 503–513, May–June 2010.

32. Gruber, M.R., Nejad, A.S., Chen, T.H., and Dutton, J.C., "Transverse Injection from Circular and Elliptic Nozzles into a Supersonic Crossflow." *Journal of Propulsion and Power*, Vol. 16, No. 3, pp. 449–457, 2000.

33. Boles, J., Milligan, R., Hagenmaier, M., Eklund, D., and Edwards, J., "Hybrid Large-Eddy Simulation / Reynolds-Averaged Navier-Stokes Simulation of Sonic Injection into Mach 2 Crossflow." *AIAA Paper 2011–769*, Orlando, FL, January 2011.

Complex Network Modeling with an EMULAB HPC

Virginia Watson Ross and David L. Hench
US Air Force Research Laboratory, Information Directorate (AFRL/RI), Rome, NY
{virginia.ross, david.hench}@rl.af.mil

Abstract

To support the Department of Defense (DoD) networks in the field, next-generation complex network product designs need to be evaluated for optimum performance. Network emulation plays an important role in evaluating these next-generation complex network product designs. From the component level to the system-of-systems level, emulation enables evaluation in a real-system context, greatly reducing the cost and time of testing and validation throughout the design cycle. For accurate network synthesis, emulation must support real-time speed, full-packet fidelity, and provide transparency.

For example, the Joint Tactical Radio System (JTRS) has critical needs for network evaluation, including researching the JTRS networking waveforms. With JTRS currently undergoing massive revision, this emulation can help save time and resources in modeling the network for system development and testing.

The Network Modeling and Simulation Environment (NEMSE) capability was developed and installed on the US Air Force Research Laboratory, Information Directorate (AFRL/RI) EMULAB high performance computer (HPC), a network emulation testbed, to demonstrate this capability for future network modeling.

The NEMSE environment has demonstrated the capability to incorporate hardware and software elements to provide hardware-in-the-loop network emulation testing and support true network emulation. NEMSE provides parallel execution and highest fidelity models and the scalability and interactivity required to test and evaluate advanced network communication devices and architectures.

This capability benefits the DoD by enabling rapid technology transition of complex network architectures from research laboratories to the field. Actual Joint Tactical Radio System (JTRS) radios, Operations Network (OPNET) emulations, and GNU (recursive definition for **G***NU is* **N***ot* **U***nix) open-source software-defined-radio software/ firmware/ hardware emulations can be accommodated.*

1. Introduction

To address next-generation complex network product designs, the Network Modeling and Simulation Environment (NEMSE) program was dedicated to facilitating the evaluation and maturation of fundamental research being conducted within the US Air Force Office of Scientific Research (AFOSR) Complex Networks program, by creating an emulation test bed environment that addresses transition from simulation to hardware.

NEMSE provides a modeling and simulation environment where the best of current emulation techniques are available to users in an easy-to-use, integrated form on the Air Force Research Laboratory, Information Directorate's (AFRL/RI's) EMULAB. This integration includes: 1) easy installation of software and emulation tools on individual processors, 2) the ability to interact with other tools and the tool's own application programmer interface (API), and 3) license management. NEMSE was designed to provide model development, protocol development, packet statistics, hardware evaluation, prototype, hardware development and transition paths for all levels of the Open Systems Interconnection (OSI) protocol stack of a network architecture. NEMSE is currently available for use by Department of Defense (DoD) scientists and engineers.

The AFRL/RI EMULAB, shown in Figure 1, consists of 86 experimental nodes, each with a 3.0 GHz Quad Core Intel Xeon E5450 processor, 12MB cache, 16GB of memory, and a 500GB hard drive for storage. There are six Network Interface Cards (NICs): (four experimental, one control, one reserved) per node. The system employs network emulation software from the University of Utah (UU) master Emulab facility. Three dedicated servers are used for EMULAB setup

and control. Networks for testing are set up using Cisco 6509E programmable routers. Note that "EMULAB" (all capital letters) refers to the AFRL system, while "Emulab" is a more generic term for a system using the UU Emulab software.

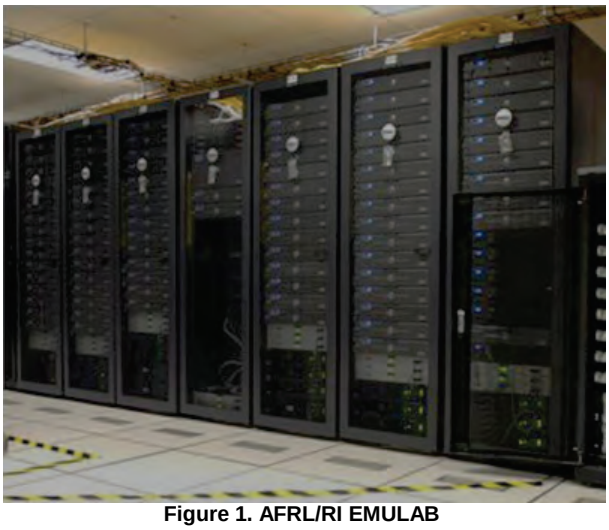


Figure 1. AFRL/RI EMULAB

The EMULAB system is accessed through a high performance computer, and is remotely accessible, allowing outside researchers to take advantage of these unique system capabilities from geographically-separate research sites.

2. Background

The design, scaling and portioning of network emulation test beds is an area of currently active research. Alvarez, et al. (2011) advanced the idea that for a system-under-test (SUT), one can consider simulation to model the behavior entirely, emulation to utilize actual entities in real-time, and a test bed to be the hardware and software where a scaled version of the targeted system is evaluated. Alvarez, et al. also surveyed partitioning techniques and scalability.

2.1 Emulation Environments

Two of the currently popular test beds are the original Emulab test bed at the University of Utah and the DETER test bed, are hosted at both USC ICI and at the University of California, Berkeley (Emulab: Total network testbed, 2012). There are many other Emulabs, such as the AFRL/RI EMULAB, also employing the Emulab software from the University of Utah. In most Emulab experiments, one processor of the available processor pool is assigned to each node of the simulation.

The basic Emulab system is set up to use the university standard Network Simulator 2 (NS2) network emulation package, and uses Tool Command Language (TCL) scripts. This emulation technique is effective for studying protocols that are above the physical layer, including the critical-link layer where routing protocols are developed. While a subset of NS2 is available in the basic EMULAB package, the full NS2 can be easily installed on individual nodes.

This one-node to one-machine type of assignment leads quickly to a trade-off between network fidelity and resource utilization. If the investigator is not prepared to be selective in fidelity and chooses to maintain perfect fidelity for each of the multiple hops between networks, all of the applications on all of the machines feeding each node would have to be scripted. However, this action would quickly exhaust an 86 node EMULAB cluster, and would require many programmers to create scripts and run applications. Even expanding each of the processors to fully-utilize their eight cores would be insufficient to maintain perfect fidelity. Therefore, selective fidelity is necessary.

Alvarez, et al. (2011) reviewed efforts to partition scenarios into test beds, and proposed a method of partitioning larger experiments into sub-experiments using flow graph techniques. On the other hand, simulation tools such as MATLAB, Optimized Network Engineering Tools (OPNET), NS2, and CORE (a modeling environment from Vitech) generally use reduced fidelity with traffic either modeled by discrete event simulation, with simulated packets, or by statistical analytic flows and demands to generate results. Many of these tools can be used for emulation with real-time interfaces or in co-simulation, as described by Harding, et al. (2007).

Carneiro, et al. (2011) discussed the need for software reuse from the early simulation phases through final hardware implementation. This reuse is especially important in a research lab, where there is a very short time window for software evaluation between receipt of a contractor's or university's software and hardware to integrate the product and evaluate it. Also, the variety of software tools used by contractors makes the task doubly daunting, unless the contractor's development environment can be exactly emulated. NEMSE enables emulation and reuse of software, addressing this need.

Thuente (2004) summarizes twenty years of modeling and simulation (M&S) experience, identifying OPNET as the de facto standard simulation tool for modeling and simulation of military airborne communication systems. He points out that OPNET mixes Discrete Event Simulation (DES) and statistical flows. With its Wireless-for-Defense package, OPNET has excellent mobility and antenna modeling capability. OPNET is generally considered a modeling package with simulated packets in its DES simulation. However, with its System-in-the-Loop (SITL) models, it becomes a real-time emulation package with real-packets in the computer's Network Interface Card (NIC) and real-packets out of another NIC. OPNET has excellent packet data collection and statistical analysis. OPNET, however, is not a source of real-packets.

2.2 Motivation for NEMSE

Recent failures of networking implementation in the field demonstrate the need for a pragma change in DoD network system development. Corrin (2010) called networking in Afghanistan "the perfect storm," with hurdles in transition, network operations, and infrastructure issues that could be addressed by emulation in the proper test beds. The failures in the networking waveforms for the Joint Tactical Radio System (JTRS) program have been highlighted by Axe (2012).

Hench (2009b) has described the Close Air Support Connectivity (CASCON) modeling terminal which was used in the early phases of NEMSE before the EMULAB was available. This terminal used a patch panel to interconnect individual processors instead of the programmed method of the EMULAB.

NEMSE is designed to avoid such problems in network emulations by employing not just a test bed, but a complete modeling and simulation environment. In this environment, field data from hardware, or actual hardware-in-the-loop from the test bed, can be used to verify and validate models. These models can be developed using a variety of software development tools by different contractors and universities, and integrated into the over-all scalable networking emulation.

2.3 A New Paradigm: Converging Emulation with System Development and Testing

Future military network emulation environments can do well by following the NEMSE model. The emulation environment should be able to support war gaming with operator-in-the-loop to allow for evaluation by military personnel of system performance. This environment should then aid in rapidly prototyping new hardware that can be taken to the field and tested, generating new data to validate and verify new models, restarting a new iteration of the cycle. This concept is shown in the work flow diagram of Figure 2.

Chruscicki and Hall (2007) described an initial environment utilizing a smaller test bed that mixed network emulation with military simulation tools. NEMSE has considered these military simulation tools to be important but out-of-scope of current activities.

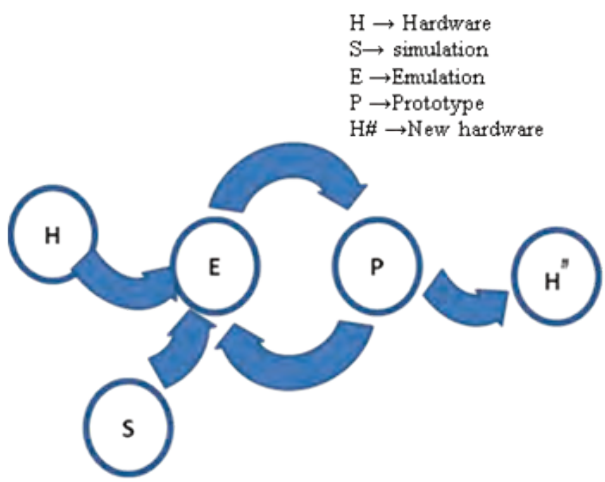


Figure 2. Idealized work flow diagram for emulation

3. Implemented Components

The AFRL/RI EMULAB has been enhanced by the Network Modeling and Simulation Environment developed by the NEMSE team. This environment allows incorporation of carefully selected university, commercial, and DoD standard emulation software. NEMSE provides effective support for both operator-in-the-loop and hardware-in-the-loop testing. During development, there has been an effort to allow reuse of contractor simulations, prototypes and code in emulating a scalable solution in the test bed and in designing updated hardware. The simulation and emulation tools that have been integrated into the environment of NEMSE are shown in Table 1. These tools are described in more detail below.

Table 1. Simulation/emulation tools for NEMSE

OSI	Layer or Function	EMULAB	NS2	OPNET	CORE	Click	MadWiFi	GNU	FPLE	RAVC	MATLAB	IPERF	WireShark
7	Application		■	■						■	■	■	
6	Presentation	■		■						■	■		
5	Session	■		■						■	■		
4	Transport	■	■	■	■					■	■		
3	Network	■	■	■	■								
2	Data Link	■	■	■	■	■							
1.2	Physical- MAC			■	■	■	■	■					
1.1	Physical-Hardware					■		■	■				
	Models		■	■	■	■							
	Protocols			■		■	■			■			
	Packet Statistics			■							■	■	■
	Hardware Evaluation							■				■	
	Prototype Hardware					■	■	■	■				

The left side of this table lists the seven layers of the open systems interconnection (OSI) stack of a network architecture, including the two sub-layers of the physical layer, and, below them, the capabilities emphasized by NEMSE: Models, Protocols, Packet Statistics, Hardware Evaluation, and Prototype Hardware.

3.1 7 EMULAB

NEMSE research has demonstrated that arbitrary networks of individual processors are easy to set up in the EMULAB. Therefore, a process of engineers designing networks using a standard symbol set, intelligent scripting of components, and a system administrator implementing experiments for the engineers, has been implemented using intelligent scripts and standard symbols. If the system administrators get overloaded with experiments, a GUI-based design could then be generated. This process is more efficient than a more traditional method of starting with the GUI, as described by Benchaib and Hecker (2011), and avoids the steep learning curves for scientists and engineers involved in using multiple emulation tools.

A typical network drawing using this technique is shown in Figure 3. This network uses a standard processor (FreeBSD operation system) accessible from secure shell over VPN, two Windows XP processors accessible from remote desktop over VPN, a network switch, and a two terminal cloud using OPNET run time with SITL.

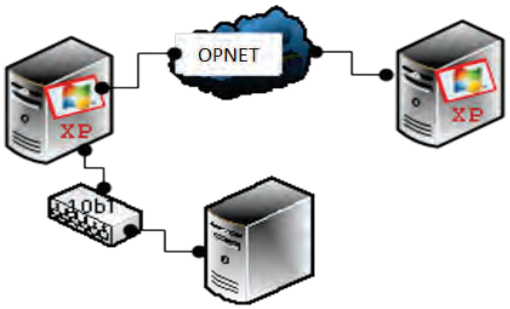


Figure 3. A typical network drawing

3.2 NS2

NS2, an emulation tool, has a variety of sources and sinks that can emulate real-traffic. A sub set of NS2 is utilized in the EMULAB Tool Command Language (TCL) scripts but the full ns-allinone code for network simulation can be included in a processor.

3.3 OPNET

An OPNET emulation setup is shown in Figure 4. The SITL module makes this emulation real-time, bringing real-digital video packets in, which have been captured from a ROVER in field experiments, and sending real-packets out.

3.4 CORE

The Common Open-Research Emulator is a DoD available network emulation tool that allows emulation on one or more PCs that is designed to connect in real-time to physical networks and routers. It is a commonly selected tool by DoD network development contractors and needs to be included.

3.5 Click

Click is a software architecture for building flexible and configurable routers that was developed at Massachusetts Institute of Technology. A Click router is assembled from packet processing modules called elements. Individual elements implement simple router functions like packet classification, queuing, scheduling, and interfacing with network devices. A typical Click router is meant to be used in the Linux kernel module. This means that a hardware implementation of the Click-developed router is a simple embedded Linux module with processing power matched to measured requirements in the emulation.

3.6 MadWiFi Driver

MadWiFi driver is an open-source 802.11 Linux driver with well-documented source code. A proprietary Hardware Abstraction Layer (HAL) in binary code is available for popular Atheros WiFi modules. Since MadWiFi is an actual Linux driver, emulation can be ported to a Linux-embedded system with an Atheros card. The Atheros cards are used by the University of Utah Emulab, and the RF physical-layer hardware emulator designed for NEMSE allows prototype hardware development in the physical layer.

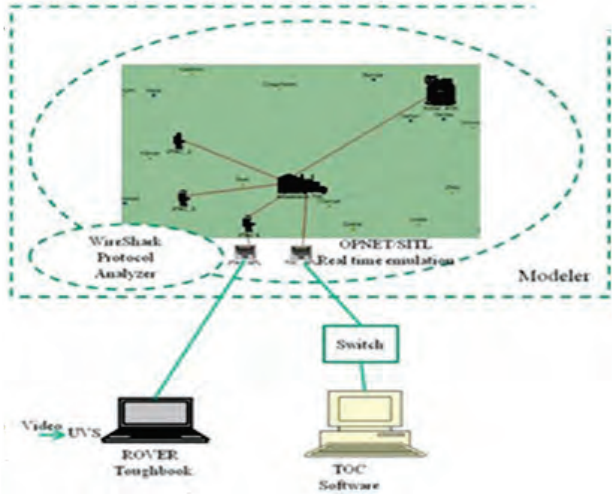


Figure 4. OPNET simulation with SITL

3.7 GNU Radio

GNU Radio, another simulation tool, is open-source software for building and testing software-defined radio systems. Used in connection with a suitable hardware device, such as the Universal Software Radio Peripheral (USRP), GNU Radio provides a simple, flexible means for emulating a variety of radio devices.

3.8 FPLE

An FPGA Physical Layer Emulator (FPLE) was added that allows (1) prototyping a Field-Programmable Gate Array (FPGA) interconnect between two computers that can be used to emulate physical layer and channel effects, and (2) integrating an FPGA development and testing environment into the EMULAB cluster to allow simple design by an engineer with limited training.

Graphical development using standard electrical engineering techniques avoids use of layout languages such as Virtual Hypertext Markup Language (VHTML). These techniques make FPGA design accessible to engineers with little additional training.

Operational hardware or a prototype can be incorporated into the EMULAB cluster by using a specially designed NEMSE box as shown in Figure 5.

Figure 5. A NEMSE box in the EMULAB for FPGA emulation

3.9 RAVC

Hench (2009a) described the Rate-Adaptive Video Coding (RAVC) technique, which has been integrated for video emulation. This technique addresses transmitting the best tactical video over bandwidths of 32 kbps to 4 Mbps.

3.10 MATLAB

M

ATLAB and OPNET have been used in co-simulations by Harding (2007); other methods of interoperating with MATLAB are also possible. Co-simulation allows MATLAB simulations to be reused in larger emulations.

3.11 Other Tools

The other tools listed are WireShark and Internet Performance Working Group (IPERF). WireShark is an open-source packet sniffer with an excellent API for packet statistics. WireShark, as a protocol analyzer, is shown in an OPNET emulation in Figure 4. IPERF is a standard network engineer's tool for generating network traffic and measuring network capacity.

4. Experimental Setup

NEMSE addressed the development of an environment that implements the paradigm for converging emulation with system development and testing. During the first two years, emulation techniques were mainly studied in isolation. During the third year, the techniques are being integrated and made interoperable through a series of use cases and exercises that can be run by new users.

A promising use case for further development outside of NEMSE is shown below.

4.1 Typical Use Case Results

NEMSE has demonstrated the ability to emulate all of the layers of the OSI stack via several use cases. In this demonstration, the team has completed the complex process of integrating all of the emulation packages for interoperability into a single emulation. This integration enables most network simulations and emulations to be performed.

An example use case emulation using OPNET terrain models has been demonstrated over the range from some radio towers across the Mohawk Valley in Upstate New York, demonstrating the ability to model terrain in field locations. Facilities and location make the area around AFRL/RI an ideal site for verifying and validating models of terrain as they interact with models of tactical radio waveforms, with a long valley surrounded by mountains. In addition to terrain, the Mohawk Valley is an area with four seasons and a variety of urban, farmland and forests, all of which are important to terrain modeling.

A demonstration of this modeling capability, generated by OPNET using NEMSE, is shown in Figures 6 to 8. Figure 6 shows a map of the Mohawk Valley in Upstate New York as generated by OPNET Wireless for Defense Terrain models. Wireless fixed-subnets are placed at AFRL's Rome, Newport, and Stockbridge facilities (blue symbols). A mobile node is placed on highway I-90. Figure 7 shows the attenuation over terrain from Rome to Newport, as generated by the OPNET Wireless for Defense Terrain models. Since there is a mountain in the pathway between Rome and Newport, runs from Rome to Stockbridge and Stockbridge to Newport were selected. Figure 8 shows the predicted Stockbridge to Newport attenuation. A generic 2 GHz, 10 MHz bandwidth transmitter was emulated with an OPNET TIREM model with default parameters.

Transmission has been verified between Stockbridge and Newport as predicted, but impossible between Rome and Newport as predicted. No further verification and validation of the model has yet been performed.

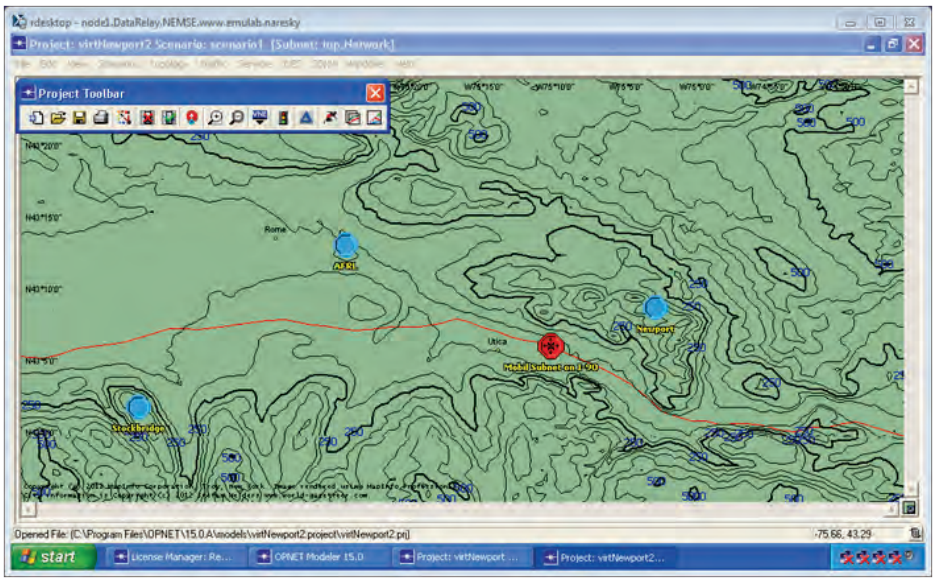


Figure 6. OPNET WIRELESS simulation of the ROME, NY area

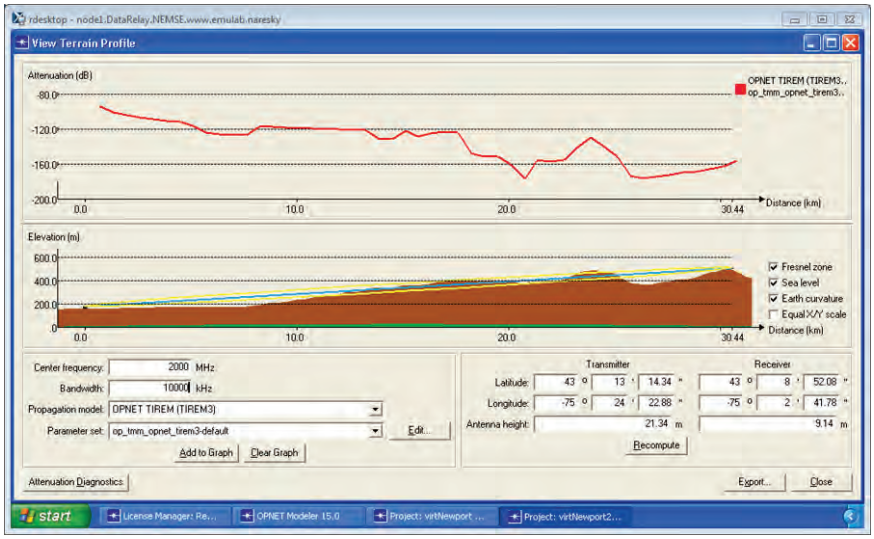


Figure 7: Attenuation from Rome to Newport

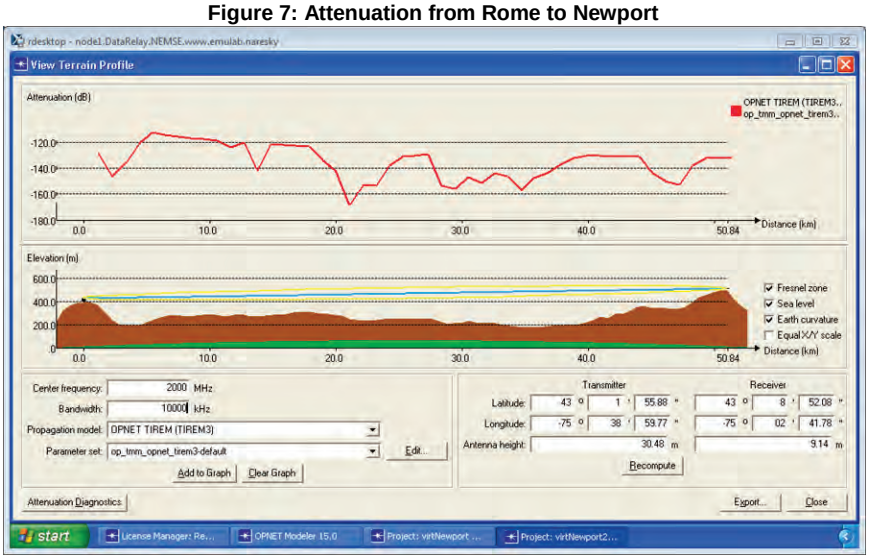


Figure 8. Attenuation from Stockbridge to Newport

5. Conclusion

The Network Simulation and Modeling Environment has demonstrated the ability to transition laboratory research experience into field capabilities by enabling simulations, code and prototypes to be used in iterative development of new network equipment. In addition, data from the field can be sanitized and used in emulation by academic or other researchers. This capability is expected to lead to quicker development of networked communication systems.

A variable fidelity approach to emulation is proving effective. This approach has been taken where high-fidelity approaches based on actual applications, and one emulation processor per one actual processor, are used for a small area close to the point of study; while emulation tools allowing reduced fidelity, with either discrete event simulation or statistical flows and demands, are used for an enlarged area.

This is a new pragma for DoD networked communications starting with contractor simulation and an older generation of hardware, proceeding through multiple iterations of emulation on a variable fidelity emulation test bed with hardware-in-the-loop, and finishing with field testing of new hardware. This approach is expected to reduce the long lead-time for communication equipment development and testing.

Future research can include verification and validation of models over the variety of terrain, weather, and crop cover. This approach also saves time and money by enabling reuse of simulations, prototypes, and code or multiple applications and simulations.

Acknowledgements

The authors wish to gratefully acknowledge the support of the AFOSR Complex Networks Program, with Laboratory Research Independent Research (LRIR) and the support for computer time on the EMULAB HPC at the AFRL/RI Affiliated Research Center.

References

Alvarez, A., Garcia, R., Cabrero, S., Paneda, X.G., Melendi, D., and Orea, R., "In pursuit of massive service emulation: a methodology for test bed building." *IEEE Communications Magazine*, vol. 49, no. 9, pp.162–168, 2011.

Axe, D., "Failure to communicate: inside the army's doomed quest for the 'perfect radio." www.iwatchnewx.org, Jan. 2012.

Benchaib, Y. and Hecker, A., "VIRCONEL: A Network Virtualizer." *Proceedings of 2011 IEEE 19th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp.429–432, 2011.

Carneiro, G., Fontes, H., and Ricardo, M., "Fast prototyping of network protocols through ns-3 simulation model reuse." *Simulation Modeling Practice and Theory*, vol. 19, no. 9, pp. 2063–2075, 2011.

Chruscicki, M. C. and Hall, F., "Airborne Networking Component Architecture and Simulation Environment (AN-CASE)." *Proceedings of IEEE Military Communications Conference, 2007*, MILCOM 2007, pp.1–7, 2007.

Corrin, A., "Army faces communications challenges in Afghanistan: 'Perfect Storm' of complex network issues requires attention." *Defense Systems Knowledge*, 09 December 2010.

Emulab: Total network testbed: Other Emulab testbeds, available at http://users.emulab.net/trac/emulab/wiki/OtherEmulabs, 2012.

Harding, C., Griffiths, A., and Yu, H., "An Interface between MATLAB and OPNET to Allow Simulation of WNCS with MANETs." *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, pp.711–716, 2007.

Hench, D.L., "Rate-Adaptive Video Compression (RAVC) Universal Video Stick (UVS)." *Proceedings of Mobile Multimedia/ Image Processing, Security, and Applications 2009*, vol. 7351, 73510 S, 2009a.

Hench, D.L., "Simulation for Close Air Support." *Proceedings of SPIE Modeling and Simulation for Military Operations IV*, Vol. 7348, 734809, 2009b.

Morris, R., Kohler, E., Jannotti, J., and Kaashoek, M.F., "The Click modular router." *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP '99)*, 1999.

Thuente, D.J., "Methodology and ground rules for simulating airborne military communication systems." *Proceedings of the 2004 IEEE Aerospace Conference*, vol.2, pp. 1061–1080, 2004.

Direct Numerical Simulation of Transition in a Hypersonic Cone Boundary Layer

Jayahar Sivasubramanian and Hermann Fasel
Department of Aerospace and Mechanical Engineering, University of Arizona, Tucson, AZ
{jayahar, faselh}@email.arizona.edu

Abstract

Direct Numerical Simulations (DNS) are performed to investigate the laminar-turbulent transition process in a hypersonic cone boundary-layer. The flow parameters used in the simulations are based on the experimental conditions of the Boeing/AFOSR Mach 6 quiet tunnel at Purdue University. The main objective of this research is to explore which nonlinear breakdown mechanisms may be dominant in a broad band "natural" disturbance environment, and then use this knowledge to perform controlled transition simulations to investigate these mechanisms in great detail. Towards this end, a "natural" transition scenario was modeled and investigated by the use of wave packet disturbances. These wave packet simulations provided strong evidence for a possible presence of second-mode fundamental and sub-harmonic resonance mechanisms in the nonlinear transition regime. However, the fundamental resonance was much stronger than the sub-harmonic resonance. In addition to these two resonance mechanisms, the wave packet simulations also indicated the possible presence of second-mode oblique breakdown mechanism. The detailed controlled transition simulations demonstrated that both, second-mode fundamental and oblique breakdown mechanisms may indeed be viable paths to complete breakdown to turbulence in hypersonic boundary layers at Mach 6.

1. Introduction

Hypersonic flow is physically different from low-speed flow. It is characterized by new physical phenomena that become important at high Mach numbers. For example, the enormous amount of aerodynamic heating (due to frictional effect of the flow over the skin of hypersonic flight vehicles) on the structure of the flight vehicles represents the main difficulty in the design and safe operation of hypersonic vehicles. It is indeed the limiting factor in aerodynamic designs for hypersonic vehicles. Most importantly, at large-Reynolds numbers, these aero-thermal loads are further increased due to the transition of a laminar boundary-layer to turbulence. Therefore, laminar-turbulent boundary-layer transition has important design implications for any Thermal Protection System (TPS) that is needed to protect a hypersonic flight vehicle and its crew (e.g., the space shuttle).

In the past and even on recent hypersonic vehicles, engineers have used a rather conservative approach for designing the TPS. In these designs, a turbulent boundary-layer was assumed to be present over the entire surface. Hence, appropriate measures to guard against the heat transfer due to these aero-thermal loads are expensive, and result in significant weight penalties. Accurate estimation of the laminar-turbulent transition region is of vital importance for designing future hypersonic vehicles, because only then can the aero-thermal loads and surface temperatures be adequately predicted, ultimately leading to enhanced payload capabilities. In addition to surface heating, transition to turbulence also has a significant effect on the aerodynamic performance of hypersonic flight vehicles, as the skin friction for turbulent boundary-layers is considerably higher than for the laminar boundary-layer.

Laminar-turbulent transition in hypersonic boundary-layers is a major unresolved topic in fluid dynamics. Even after many years of research, crucial aspects of the transition physics are still in the dark. A major difficulty here is the increased complexity of the transition physics due to the compressibility effects. The lack of a basic understanding of high-speed transition and, as a consequence, a lack of reliable transition prediction tools, hinders practical applications such as reusable launch vehicles, high-speed interceptors, high-speed civil transport (HSCT), high-speed reconnaissance aircrafts, Falcon hypersonic technology vehicle (HTV), and modern air breathing hypersonic cruise vehicles. Hypersonic transition research suffers from the fact that reliable experiments are very difficult and expensive. Wind tunnel noise level, short run-times and controlled disturbance generation are only a few of the difficulties faced in experiments. Therefore, relatively few reliable experimental studies of the growth of instability-waves have been conducted. On the other hand,

Direct Numerical Simulations (DNS) of transition in high-speed boundary-layers require high grid resolution and large computation time. As a consequence, most of the earlier research efforts for hypersonic transition have focused on the receptivity (the process by which free-stream disturbances generate instability waves in the boundary-layer) and the linear regime of boundary-layer transition. Very little is known about the late nonlinear stages of transition and, in particular, about the final breakdown to turbulence. Due to the stabilizing effect of compressibility; however, the nonlinear transition regime extends over a significant downstream extent of a hypersonic flight vehicle. As a result, transition prediction tools based on the linear N-factor calculations cannot estimate the transition location accurately. Therefore, new transition prediction tools, which incorporate nonlinear transition physics, are desperately needed for the development and safe operation of future high-speed flight vehicles. Hence, it is very important to understand the nonlinear stages of transition, including the final strongly-nonlinear breakdown, in order to identify which disturbance scenarios lead to turbulent flow and which do not. Therefore, accurate and reliable fully-resolved DNS of hypersonic boundary-layers are essential for advancing our understanding of hypersonic transition and for the development of reliable transition prediction tools.

The physics of high-speed boundary-layer transition are much more complex than for low-speeds. From linear stability theory (Mack, 1969), it is known that multiple instability modes exist for high-speed boundary-layer flows, in contrast to only one mode (Tollmien-Schlichting, TS) for the incompressible case. In addition to the so-called first-mode in supersonic boundary-layers, which is equivalent to the TS-mode in incompressible boundary-layers, higher modes exist for supersonic boundary-layers that result from an inviscid instability mechanism. According to Linear Stability Theory (LST), the most unstable higher modes are two-dimensional unlike oblique first-modes. Also from linear stability theory, it is known that the first-mode is dominant (higher amplification rates) for low-supersonic-Mach numbers while for Mach numbers above 4, the second-mode is dominant (most amplified). Due to the difficulties in carrying-out experiments (and "controlled" experiments, in particular) and due to the existence of multiple instability modes, the role and importance of the various instability modes in a realistic transition process are not understood at all. Of course, when amplitudes of the various instability modes reach high enough levels, nonlinear interactions of these modes can occur. As a consequence, the transition process in high-speed boundary-layers is highly non-unique, which means that slight changes in the environment or vehicle geometry may significantly alter the transition process.

Cones with circular cross sections represent a useful prototypical geometry for investigating boundary-layer stability and transition at hypersonic speed. In the present work, we are employing DNS to study the laminar-turbulent transition process for a 7° circular straight cone tested in the Boeing/Air Force Office of Scientific Research (AFOSR) Mach 6 Quiet Tunnel at Purdue University (Schneider, 2009). The two main goals for these investigations are to determine which nonlinear mechanisms are dominant in a broad band, "natural" disturbance environment and then to perform controlled transition simulations to study these mechanisms in great detail. Towards this end, a "natural" transition scenario is modeled using wave packet disturbances. The boundary layer is forced by a short duration (localized) blowing and suction pulse. The pulse disturbance generates a wave packet, which contains a wide range of disturbance frequencies and wave numbers. From the wave packet simulations the dominant nonlinear mechanisms were identified. Subsequently, we performed a series of highly-resolved controlled transition simulations. The results from these simulations are discussed in detail in this paper.

2. Physical Problem and Computational Method

Laminar-Turbulent transition in a hypersonic boundary layer on a 7° sharp circular cone at zero angle-of-attack is investigated using DNS. The computational setup follows the experiments in the Boeing/AFOSR Mach 6 Quiet Tunnel at Purdue University (Schneider, 2009). The cone model used in the Purdue experiments has a semi vertex angle of 7°, and a cone length of L^*=0.517*m*. The nose radius of the cone is 0.05*mm*, and therefore, the cone can be considered a "sharp cone". The approach flow has a Mach number of 6 and a unit Reynolds number of $10^6 m^{-1}$. The stagnation temperature and pressure are 430°*K* and 1,043*KPa*, respectively. The cone surface is set to be isothermal with 300°*K*.

For our simulations we apply a two-step computational approach as shown in Figure 1. In step 1, we simulate the entire cone geometry including the shock wave emanating from the nose tip of the cone. For this task, we have available an in-house developed research code based on the compressible Navier-Stokes equations (Gross and Fasel, 2010). This finite volume code has the option between several different numerical schemes. Among these is a second order TVD scheme which was used to obtain the basic state, i.e. the undisturbed laminar flow.

In step 2, we use the flowfield obtained from the finite volume code as initial and boundary conditions for the high-order accurate stability and transition simulations. Hereby, the computational domain extends only over a small part of the cone (see Figure 1) in order to channel all available computational resources into the region of interest. For these DNS of transitional and turbulent high-speed flows, we employ a state-of-the-art in-house developed high-order accurate Navier-

Stokes code (Laible, et al., 2008; 2009). The compressible Navier-Stokes equations are solved on a conical coordinate system. The fluid is treated as an ideal gas and the viscosity is calculated using Sutherland's law with a correction for low temperatures (Sivasubramanian and Fasel, 2011). For the time integration an explicit, low storage, forth-order Runge-Kutta scheme is adopted. The spatial discretization is based on high-order grid-centered upwind differences for the convective terms (Zhong, 1998) and high-order central differences for the viscous terms. The convective fluxes are split into upwind and downwind fluxes using the method proposed by van Leer (1982).

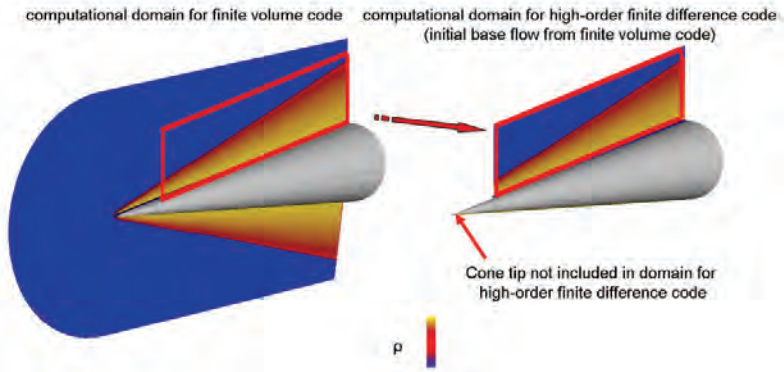


Figure 1. Schematic of simulation strategy used for our stability and transition investigations. Axisymmetric flowfield (showing iso-contours of density) over circular cone calculated by the finite volume code (left) and computational domain for the high-order accurate stability and transition simulations (right).

Using the Message Passing Interface (MPI), the finite difference code was parallelized for shared and distributed memory machines. The domain is decomposed in both downstream and wall-normal directions such that the interface area, and hence the communication time, is minimized. The domain decomposition in the third dimension would be more involved because of the pseudo-spectral discretization using Fourier modes. Each sub-domain consists of internal points, which are advanced in time, and ghost points, which are exchanged between the sub-domains after each time-step/sub-step. The ghost points allow the use of grid-centered finite difference stencils all the way to the boundaries of the sub-domains. The ghost point flow quantities are computed in the neighboring sub-domains. At the end of each Runge-Kutta sub-step the ghost points are updated (first in the x-direction and then in the y-direction). The efficiency of the implemented parallelization algorithm for distributed memory machines has been tested, and it turned out to be high enough to justify the use of at least 512 processors and possibly more. In order to evaluate the performance of the parallel algorithm, we define the speed-up and the parallel efficiency, as $Sp=t(1)/t(n)$ and $\eta=Sp/n$, respectively, where $t(n)$ is the elapsed time of the computation using n processors. The performance of the code on the Cray XT4 (US Army Engineer Research and Development Center [ERDC] JADE) architecture is evaluated by computing 100 time steps of two- and three-dimensional (2D and 3D) transition simulation at Mach 3.5. Tested are three different cases: a) 2D simulation with 1,200×400 points, b) 3D simulation with 1,200×400×129 grid points, and c) 3D simulation with 1,200×400×257 grid points. The resolutions of these cases are in the range of typical stability and transition simulations. The memory requirements for these cases are such that more than the tested number of processors is needed to fit the memory completely into the cache.

The results of these test cases are shown in Figure 2. Due to the memory constraints, not all cases could be run on a small number of processors. Therefore, the data points start at 128 processors. Figure 2 shows excellent speed-up and parallel efficiency for all three test cases for up to 512 processors. Bottlenecks, which may limit the speed-up for a large number of processors ($\gtrsim$1,024), are: i) the communication between processors, and ii) the overhead due to a small number of operations performed on the ghost points. The degree to which these bottlenecks affect the parallel efficiency is related to the ratio between the grid points and the ghost points of each sub-domain. Therefore, it can be expected that for larger number of grid points the speed-up and efficiency will be even better.

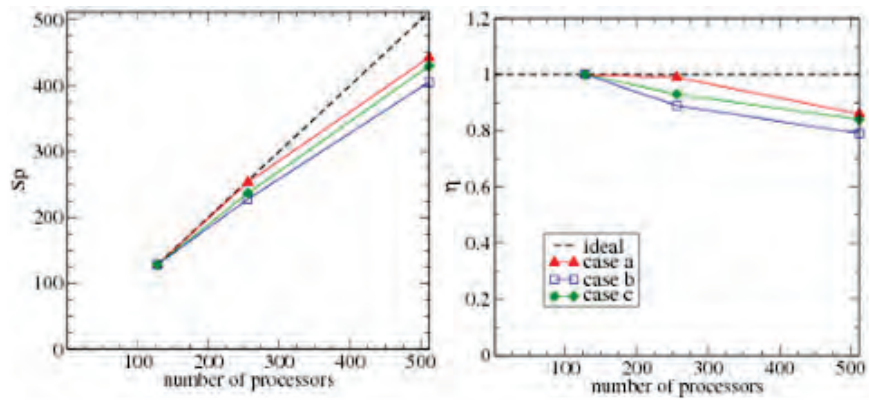


Figure 2. Parallel speed-up, Sp, and parallel efficiency, η, for typical cone boundary layer transition simulations on the Cray XT4 (JADE). (a) Left: speed-up comparison between cases (a-c). (b) Right: parallel efficiency comparison between cases (a-c).

3. Fundamental Breakdown

The wave packet simulations have provided strong evidence of the presence of a fundamental resonance mechanism in the nonlinear transition regime of a sharp cone boundary layer at Mach 6 (Sivasubramanian and Fasel, 2012). Fundamental resonance or (K- or Klebanoff-type) breakdown is initiated by a secondary instability mechanism involving an axisymmetric primary wave (1, 0) and a symmetric pair of oblique secondary waves ((1, 1) and (1, −1)) of the same frequency and/or a steady longitudinal vortex mode (0, 1). Note that in the present paper, a wave is denoted by (n, k), where n represents the frequency and k the azimuthal wave number, both normalized by the corresponding values of the primary pair of oblique waves. The strength of the fundamental resonance is strongly influenced by the amplitude of the primary wave (1, 0) and the wave angle of the secondary oblique wave pair. For a cone, this issue is even more complicated than for a flat-plate, because the wave angle of a disturbance wave changes in the downstream direction. Therefore, we first performed a parameter study to find the most strongly resonating oblique wave pair. Then, we performed a set of highly-resolved controlled fundamental breakdown simulations using the most strongly resonating oblique wave pair as secondary waves.

3.1 Parameter Study

Fundamental resonance can be initiated by forcing the primary wave (1, 0) at a large amplitude and the secondary disturbance waves (1,±1) at a low amplitude. The dominant (linearly most amplified) axisymmetric wave was chosen as the primary wave. The most amplified primary wave was found from comparing the N-factors for axisymmetric waves for various frequencies. The streamwise distribution of the N-factors is plotted in Figure 3 for axisymmetric disturbance waves (k_c=0) as obtained from a low-amplitude wave packet simulation. In Figure 3, the N-factor reached by the most amplified axisymmetric wave (k_c=0) is about 9.5. The axisymmetric wave that reached the highest N-factor within the computational domain has a nondimensional frequency of $F=1.071\times10^{-5}$ (f^*~210kHz) and was therefore chosen as the primary wave (1, 0) for a parameter study to determine the most strongly resonating oblique secondary wave pair (1,±1).

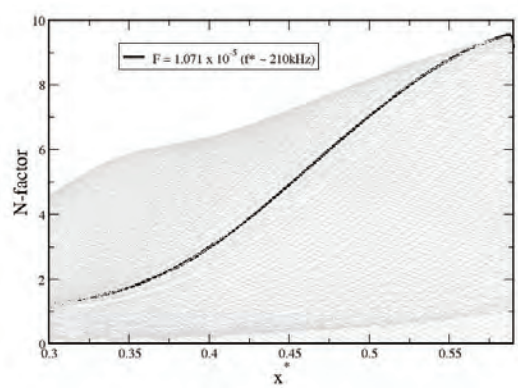


Figure 3. N-Factor curves for axisymmetric waves (k_c=0) obtained from a low-amplitude wave packet simulation

A series of low-resolution simulations were performed for several azimuthal wave numbers (k_c) ranging from 10 to 230. In these simulations, the axisymmetric primary wave (1, 0) with nondimensional frequency of $F=1.071\times10^{-5}$ (f^*~210kHz) was forced at a moderate amplitude ($A_{1,0}$=0.1% of the freestream velocity) and the secondary oblique waves (1,±1) were forced at a low amplitude ($A_{1,\pm1}$=0.001% of the freestream velocity). The results obtained from the parameter study are presented in Figure 4. Figure 4a shows the growth rate (σ) of the secondary oblique wave pair (1,±1) as a function of azimuthal mode number (k_c). The growth rates (σ) were extracted at the streamwise position x^*=0.578m (R_x=2,875). A broad band of azimuthal modes experiences resonant secondary growth, but according to Figure 4a, waves with k_c=150 have the strongest secondary growth rate. In Figure 4b, the downstream amplitude development of selected modes is shown for the low resolution simulation performed for azimuthal mode number k_c=150. These curves were obtained from plotting the wall-normal maximum of streamwise velocity disturbances. As can be observed in Figure 4b, when the axisymmetric primary wave (1, 0) exceeds a certain threshold amplitude, the secondary oblique wave pair (1,±1) and the steady longitudinal vortex mode (0, 1) experience strong secondary growth.

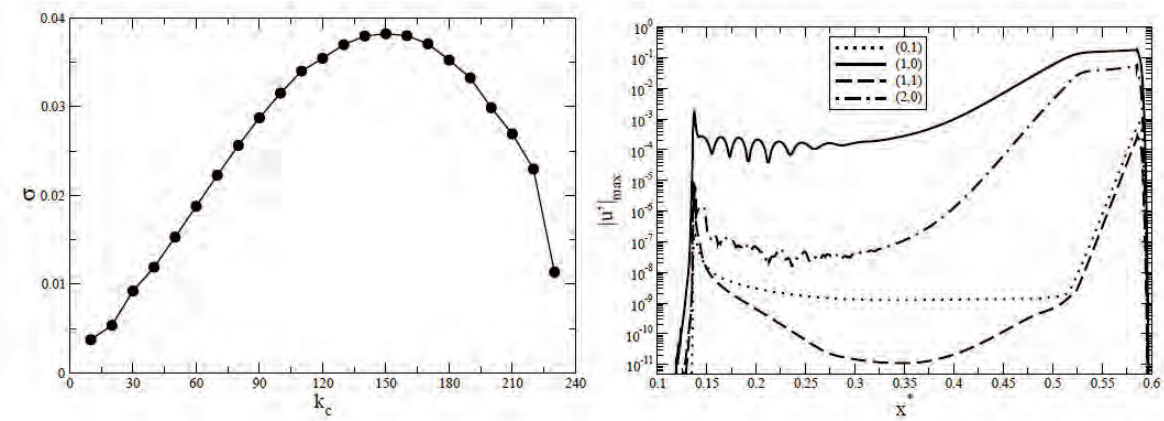


Figure 4. (a) Fundamental secondary growth rate (σ) as a function of azimuthal mode number (k_c) and (b) maximum of streamwise velocity disturbance versus downstream distance. Shown are selected modes, which play an important role in the early nonlinear stage of fundamental resonance.

3.2 High-Resolution Fundamental Breakdown Simulation

From the parameter study discussed previously, oblique waves with azimuthal mode number k_c=150 were identified as the most strongly resonating oblique waves. Therefore, we performed a highly-resolved fundamental breakdown simulation with azimuthal mode k_c=150 for the secondary wave pair. In these simulations, the axisymmetric primary wave (1, 0) with a nondimensional frequency of $F=1.071\times10^{-5}$ was forced with a large amplitude ($A_{1,0}$=4.0% of the freestream velocity).

The downstream development of the wall-normal amplitude maximum of the u-velocity disturbance as obtained from the fundamental breakdown simulation is presented in Figure 5. Selected modes that play an important role in the early nonlinear stage of a fundamental resonance mechanism are shown. As before, when the axisymmetric primary wave (1, 0) reaches a threshold amplitude, the secondary mode (1, 1) and the steady longitudinal vortex mode (0, 1) grow faster than the primary wave and eventually reach the same amplitudes as the primary mode (1, 0). Higher harmonics modes (2, 0) and (2, 1) are also generated. When modes (1, 1) and (0, 1) approached the amplitude of the primary mode (1, 0) all higher modes experienced rapid streamwise growth as the transition process becomes highly nonlinear, which is an indication of the onset of the final breakdown to turbulence. The downstream development of the time- and azimuthally-averaged skin friction coefficient, C_f is shown in Figure 6. The skin friction initially follows the laminar curve up to x^*~0.41m. Then the skin friction increases from the laminar curve towards the turbulent curve. However, farther downstream it unexpectedly drops sharply to almost the laminar value before rising again steeply and overshooting the turbulent skin friction curve close to the end of the computational domain.

A careful examination of the results shown in Figures 5 and 6 reveals that the initial rise in skin friction is due to the large amplitude reached by the primary wave (1, 0). The first peak in the skin friction corresponds to the streamwise location where the primary wave (1, 0) saturates. This peak in skin friction also roughly coincides with the location where the secondary disturbances (e.g., modes (1, 1) and (0, 1)) reach their largest amplitude levels. As the primary wave starts to decay following the nonlinear saturation, the skin friction dips to almost the laminar value. Then, a very steeper rise in skin friction occurs when all higher modes experience very strong nonlinear amplification. As a result of this steep rise the

skin friction reaches values higher than the theoretical turbulent curve. Note that this particular development of the skin friction is likely due to the "controlled" transition scenario where only a specific set of waves are forced. In a "natural" transition scenario this development may be less pronounced or even completely absent.

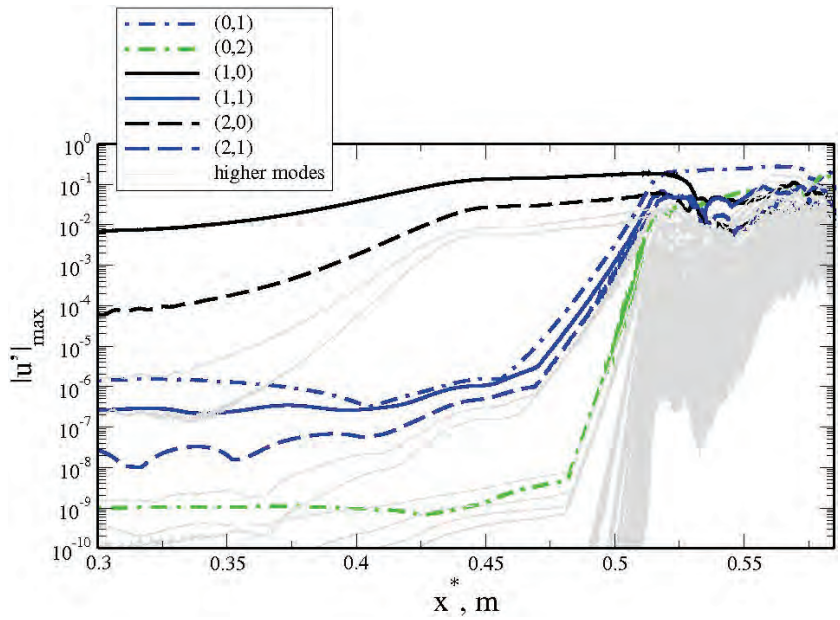


Figure 5. Streamwise development of the maximum u-velocity disturbance amplitude from the fundamental breakdown simulation. Modes (1, 1) and (0, 1) grows nonlinearly and eventually approach the amplitude of the primary wave (1, 0). Higher harmonics, modes (2, 0) and (2, 1) are generated and when modes (1, 1) and (0, 1) approached the amplitude of the primary mode (1, 0) all higher modes experienced rapid streamwise growth.

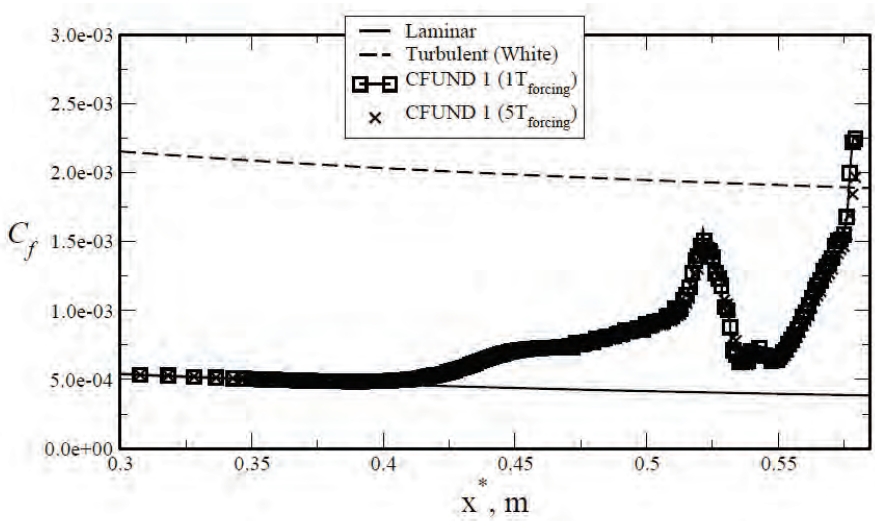


Figure 6. Time and azimuthal averaged skin friction coefficient from the fundamental breakdown simulation. Initial rise in skin friction is caused by the large amplitude primary wave (1, 0). This is followed by a dip caused by nonlinear saturation of the primary wave (1, 0). Then a steeper rise in skin friction occurs when all higher modes experience nonlinear growth. Note that the interval for the time average is indicated by the number of forcing periods.

Identification of dominant flow structures and their streamwise development will give further insight into the underlying physical mechanisms of the transition process. Instantaneous flow structures identified by the Q-criterion are shown in Figure 7. The flow structures reveal the typical evolution of a K-type breakdown: During the initial stage of the transition process the dominant wave is axisymmetric. Once nonlinear interactions cause the oblique secondary waves to amplify rapidly, the dominant axisymmetric waves become modulated in the azimuthal direction. The interaction of the dominant axisymmetric wave and finite amplitude oblique waves leads to the formation of Λ–vortices. The Λ–vortices

appear in an aligned pattern because the primary axisymmetric wave and the secondary oblique waves have the same frequency. The evolution of the flow structures during the final stage of transition can be better observed in the close–up view. The aligned arrangement of Λ–vortices can be clearly identified.

Isocontours of the time averaged wall heat flux (dT/dy) from the fundamental breakdown simulation is shown in Figure 8. Remarkable are the streamwise aligned "hot" streaks. These streaks look similar to the streamwise streaks observed in the Purdue experiments using temperature sensitive paint (Berridge, et al., 2010). The qualitative similarity of the "hot" streaks observed in our simulations and the streamwise streaks observed in the Purdue experiments for a flared cone (see Figure 7b in Berridge, et al., 2010), may be an indication that fundamental resonance might have played a role in the unforced "natural" transition experiments at Purdue.

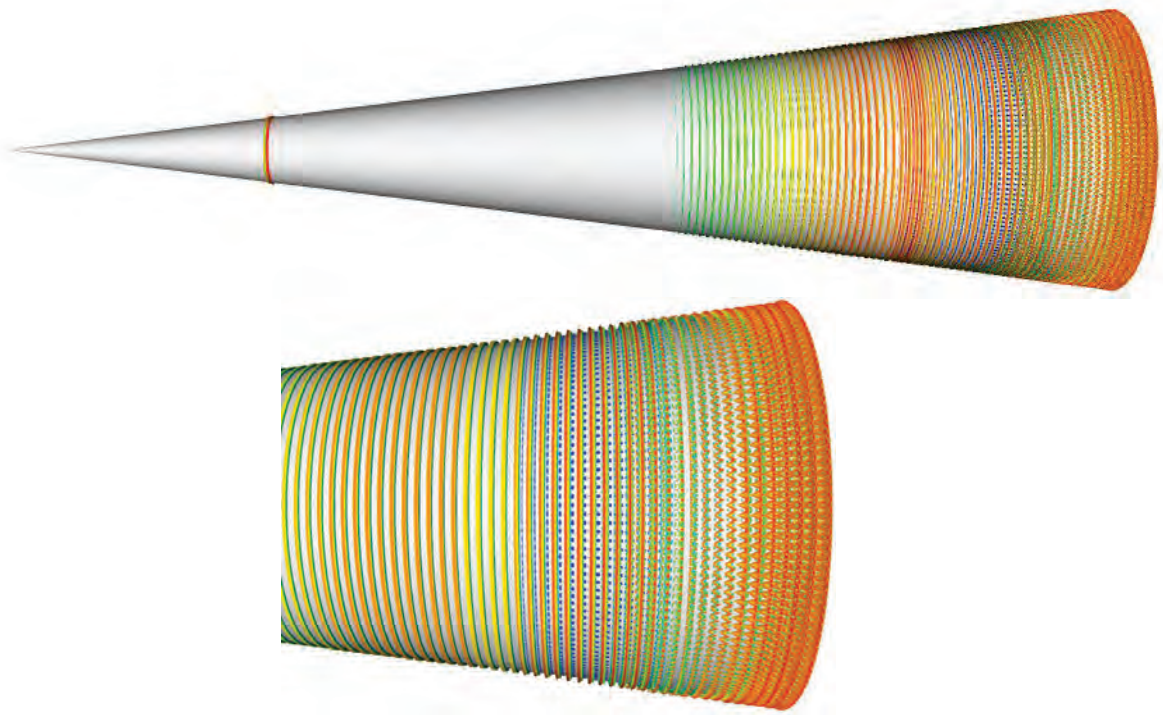


Figure 7. Visualization of flow structures by iso-surface of Q criterion from the fundamental breakdown simulation. The iso-surface is colored using the streamwise velocity magnitude.

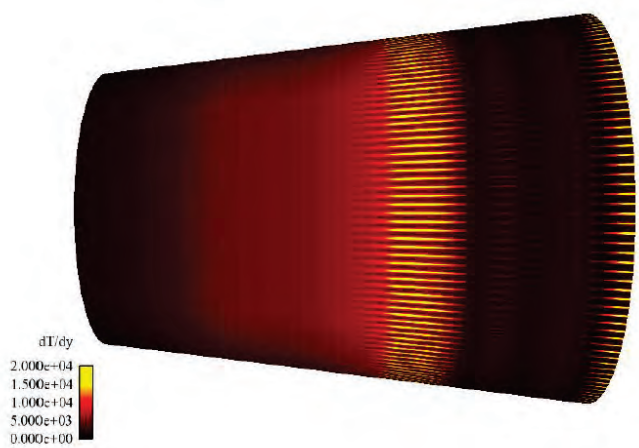


Figure 8. Time averaged heat transfer (dT/dy) at the wall from the fundamental breakdown simulation. The streamwise aligned "hot" streaks observed in our simulations looks qualitatively similar to the streamwise streaks observed in the Purdue experiments using temperature sensitive paints for a flared cone.

4. Oblique Breakdown

In addition to providing evidence for the presence of a fundamental resonance mechanism the wave packet simulations have also indicated a possible presence of an oblique breakdown mechanism in the nonlinear transition regime of a sharp cone boundary layer at Mach 6. Oblique breakdown is not a secondary instability mechanism in the classical sense. Rather than a primary disturbance causing nonlinear growth of a lower amplitude secondary disturbance, oblique breakdown is initiated only by a pair of oblique disturbance waves with identical, but opposite, wave angles - modes (1,±1). The pair of oblique disturbances grows and eventually reaches amplitudes large enough to cause nonlinear amplification of numerous higher modes. In the nonlinear transition region, the primary pair of oblique waves transfer energy to a steady mode (longitudinal vortex) with twice the spanwise wave number of the primary oblique waves. The growth of this steady mode results from a wave-vortex triad ((1, 1), (1,−1) and (0, 2)) and is considered to be the dominant mechanism early in the transition process. As transition progresses, higher harmonics are generated and the spectrum is rapidly broadened, especially in the spanwise direction.

In order to compare with the previously discussed fundamental breakdown simulations and to find out if oblique breakdown can indeed be a viable path to complete breakdown to turbulence, we performed highly-resolved oblique breakdown simulations. The disturbance spectrum obtained from wave packet simulations showed strong speaks for shallow oblique waves with azimuthal wave numbers k_c±20. Consequently, waves with this azimuthal wave numbers were forced in the present simulation in order to initiate the transition process via oblique breakdown. The frequency was chosen to be F=1.071×10^{-5} (f*~210kHz), because it is linearly the most amplified frequency for the azimuthal mode number k_c=20 as shown in Figure 9. Note that the dominant axisymmetric wave (k_c=0) and the dominant oblique wave with k_c=20 have about the same frequency.

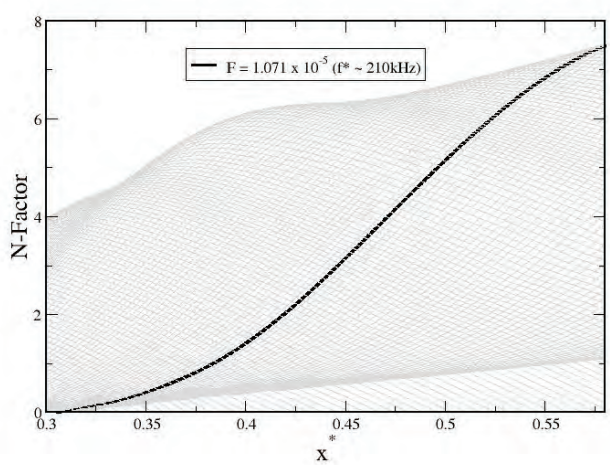


Figure 9. N-factor curves for oblique waves with azimuthal wave number k_c=20. The N-factor curves were obtained from a low-amplitude wave packet simulation.

The downstream development of the wall-normal amplitude maximum of the u-velocity disturbances as obtained from the oblique breakdown simulation is presented in Figure 10. Selected modes that play an important role in the early nonlinear stage of an oblique breakdown mechanism are shown. Highlighted are the signature modes of oblique breakdown: the initially forced mode (1, 1), the nonlinearly generated steady modes (0, 2), (0, 4), (0, 6), and mode (1, 3). Both nonlinearly generated steady modes (0, 2), (0, 4), (0, 6), and mode (1, 3) experience a stronger amplification than the primary wave (1, 1), and eventually reach the same amplitude level. Once certain threshold amplitude is reached, nonlinear mechanisms eventually trigger the transition process. Then all higher modes experienced rapid streamwise growth as the transition process becomes highly nonlinear, which leads to the onset of the final breakdown to turbulence.

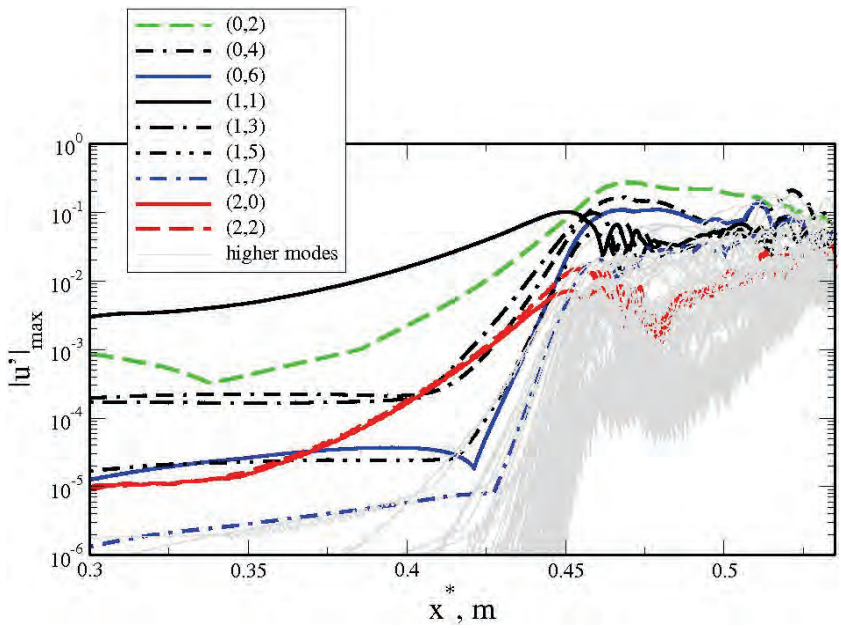


Figure 10. Streamwise development of the maximum u-velocity disturbance amplitude from the oblique breakdown simulation

The downstream development of the time- and azimuthally-averaged skin friction coefficient, C_f is shown in Figure 11. The skin friction initially follows the laminar curve up to x^*~0.44m. Then there is a small initial rise and then a dip in the skin friction coefficient, similar to the initial rise and fall observed in the fundamental breakdown simulations. As in the case of fundamental breakdown the initial rise in skin friction is related to the high amplitude primary wave. However, the rise is much less pronounced for oblique breakdown which could be due to the lower amplitude of the primary wave (1, 1) because the oblique primary wave is not as strongly amplified as axisymmetric disturbances. Farther downstream the skin friction rises again steeply and overshoots the turbulent skin friction curve close to the end of the computational domain.

Flow structures from the oblique breakdown simulation are presented in Figure 12. The linear regime is mainly influenced by the forced modes (1, 1). Thus the flow structures in the linear regime represent the superposition of both waves. Then in the nonlinear regime generation of higher modes, especially the vortex modes (0, 2), (0, 4), (0, 6) alter the flowfield. Finally, as the nonlinear interactions become stronger the flow breaks up into wedge-shaped regions where small-scale flow structures develop.

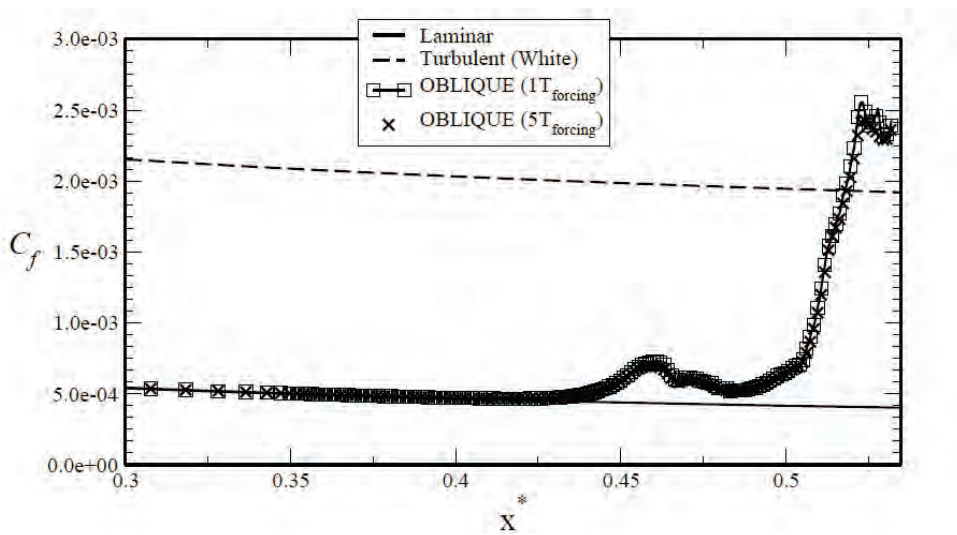


Figure 11. Time- and azimuthal-averaged skin friction coefficient from the oblique breakdown simulation. Note that the interval for time average is indicated by the number of forcing periods.

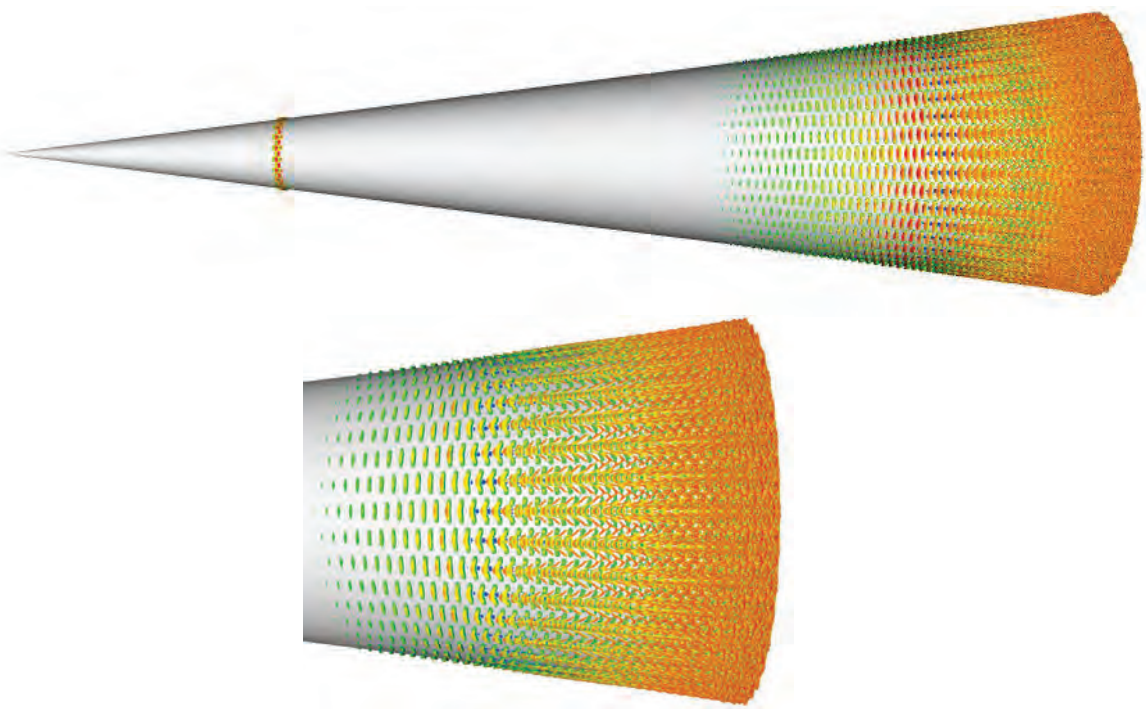


Figure 12. Visualization of flow structures by iso-surface of Q criterion from the oblique breakdown simulation. The iso-surface is colored using the streamwise velocity magnitude.

Iso-contours of the time averaged wall heat flux (dT/dy) from the oblique breakdown simulation is shown in Figure 13. The streamwise arranged "hot" streaks observed in the case of oblique breakdown looks strikingly different from that observed for fundamental breakdown. As in the case of fundamental resonance, these streaks are a consequence of the steady modes (longitudinal vortex). However, in the case of oblique breakdown the nonlinearly generated steady modes are (0, 2), (0, 4), (0, 6), etc. Therefore the azimuthal spacing of these streaks is different for oblique breakdown.

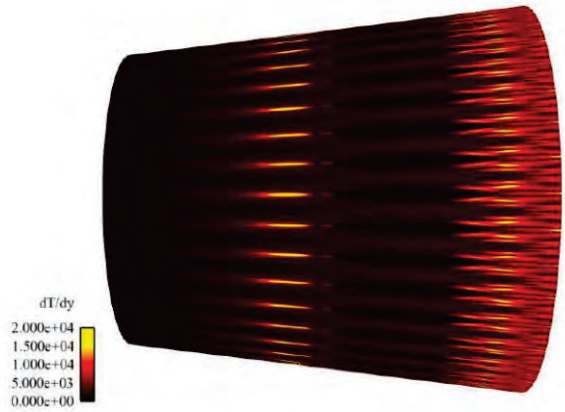


Figure 13. Time-averaged heat transfer (dT/dy) at the wall from the oblique breakdown simulation. The streamwise aligned streaks observed in the case of oblique breakdown looks strikingly different from that observed for fundamental breakdown.

In comparison to fundamental breakdown, the onset of transition, the initial rise and dip in skin friction and the final rise in skin friction in oblique breakdown shows similar trend. However, it is extremely difficult to compare transition onset between these two mechanisms because they are of a completely different nature. The true onset of transition for fundamental breakdown could be considered to occur after the first rise in skin friction because this initial distortion of the mean flow is caused entirely by the axisymmetric primary disturbance wave. With this consideration, the true onset of transition occurs later for fundamental breakdown. At this point, however, the slope of the skin friction is about the same

for both breakdown scenarios. Thus, it is difficult to draw any concrete conclusions regarding the relative dominance of one transition mechanism over the other and it suffices to say that both are likely to be relevant for hypersonic boundary layers.

5. Conclusion

Direct numerical simulations were performed to investigate laminar-turbulent transition for a sharp circular cone at Mach 6. A long term objective of our research is to determine which nonlinear mechanisms are dominant in a broad band disturbance environment and then to gain more insight into the nonlinear development by performing controlled transition simulations. Towards this end, we modeled and investigated a "natural" transition scenario using short duration (localized) pulse disturbances. The pulse forcing leads to a three-dimensional wave packet with a broad band of frequencies and wave numbers. These wave packet simulations provided strong evidence for strong second-mode fundamental and oblique breakdown mechanisms in the nonlinear transition regime.

We performed a set of highly resolved "controlled" fundamental and oblique breakdown simulations. The skin friction coefficient obtained from these simulations initially follows the laminar curve, and then increases from the laminar curve towards the theoretical turbulent one. However, farther downstream the skin friction drops back to nearly the laminar curve before rising again steeply and even overshooting the theoretical turbulent curve. A closer look at the streamwise development of disturbance waves revealed that the initial rise in skin friction is caused by the large amplitude primary wave. The first peak in the skin friction roughly corresponds to the streamwise location where the primary wave saturates. As the primary wave starts to decay following the nonlinear saturation, the skin friction decreases strongly. Then finally, a steep rise in skin friction occurs when all higher modes experience strong nonlinear amplification, and results in an overshoot of the theoretical turbulent curve. Such observations were also made in the Purdue experiments and the streamwise streaks observed in the experiments look qualitatively similar to the "hot" streaks observed in the simulations. This suggests that both second-mode fundamental and oblique breakdown may have played a role in the natural (unforced) transition experiments at Purdue. Moreover, the simulations performed under this project demonstrated that both mechanisms may be viable paths to complete breakdown to turbulence in hypersonic boundary-layers.

6. Significance to DoD

Hypersonic transition research is vitally important for the success of many DoD programs such as global reach bombers and space operations. Boundary-layer transition is associated with very high aerodynamics heating and unsteady aerodynamic loads, and therefore has an enormous impact on the performance of hypersonic flight vehicles. As a consequence of this, reliable transition prediction tools are desperately needed for the development and safe operation of future high-speed flight vehicles. Our highly-resolved DNS of the late stages of transition are part of the research program of the National Center for Hypersonic Laminar-Turbulent transition (jointly funded by AFOSR and NASA) with the goal of advancing the understanding of high-speed transition and hence to improve existing transition prediction methods and to suggest strategies for transition control.

Systems Used

Typical simulations were computed on 512 processors on the Cray XT4 (Jade) and Cray XE6 (Garnet) at the US Army Engineer Research and Development Center (ERDC), and on up to 1,024 processors on the Cray XE6 (Chugach) at the Arctic Region Supercomputing Center (ARSC).

Acknowledgments

This work was funded by the AFOSR/NASA National Center for Hypersonic Laminar-Turbulent Transition Research at the Texas A&M University. Computer time provided by ARSC and ERDC under Challenge Project AFOSR26292C4R is gratefully acknowledged.

References

Berridge, D., Chou, A., Ward, C., Steen, L., Gilbert, P., Juliano, T., Schneider, S., and Gronvall, J., "Hypersonic Boundary-Layer Transition Experiments in a Mach 6 Quiet Tunnel." *AIAA-2010-1061*, 2010.

Gross, A. and Fasel, H.F., "Numerical investigation of supersonic flow for axisymmetric cones." *Mathematics and Computers in Simulation*, 81 (1), pp. 133–142, 2010.

Laible, A., Mayer, C., and Fasel, H., "Numerical Investigation of Supersonic Transition for a Circular Cone at Mach 3.5." *AIAA-2008-4397*, 2008.

Laible, A., Mayer, C., and Fasel, H., "Numerical Investigation of Transition for a Cone at Mach 3.5: Oblique Breakdown." *AIAA-2009-3557*, 2009.

van Leer, B., Flux-Vector Splitting for the Euler Equations, *International Conference on Numerical Methods in Fluid Dynamics*, Springer-Verlag, 170, pp. 507–512, 1982.

Mack, L.M., "Boundary-Layer Stability Theory." *Internal Document 900-277*, Jet Propulsion Laboratory, Pasadena, CA, 1969.

Schneider, S.P., Private Communication, 2009.

Sivasubramanian, J. and Fasel, H.F., "Transition Initiated by a Localized Disturbance in a Hypersonic Flat-Plate Boundary Layer." *AIAA-2011-0374*, 2011.

Sivasubramanian, J. and Fasel, H.F., "Growth and Breakdown of a Wave Packet into a Turbulent Spot in a Cone Boundary Layer at Mach 6." *AIAA-2012-0085*, 2012.

Zhong, X., "High-Order Finite-Difference Schemes for Numerical Simulation of Hypersonic Boundary-Layer Transition." *J. Comp. Phys.*, 144, pp. 662–709, 1998.

Highly Accurate Simulations of Low Reynolds Number Micro Air Vehicle Wing Aerodynamics

Raymond E. Gordnier, Miguel R. Visbal, and Caleb J. Barnes
US Air Force Research Laboratory, Air Vehicles Directorate, Computational Aerophysics Branch (AFRL/RB), Wright-Patterson AFB, OH
{raymond.gordnier, miguel.visbal, caleb.barnes}@wpafb.af.mil

Abstract

In this paper, computations are performed to better understand unsteady, low-Reynolds number aerodynamics, which is critically important in providing insight into the flying characteristics of natural fliers and exploiting natural aerodynamic efficiencies for micro air vehicle (MAV) development. The first computations explore the flow structure over a bio-inspired corrugated airfoil. The flow behavior was found to be strongly affected by the leading-edge geometry, due to interaction between the detached shear-layer emanating from the leading-edge and the first corrugation peak. The second set of simulations explores the aerodynamics of an aspect-ratio two, flexible-membrane wing at angle-of-attack. The impact of membrane flexibility on the aerodynamic performance is shown to come primarily from the development of mean camber, with some further effects arising from the interaction between the dynamic motion of the membrane and the unsteady flowfield above. Finally, the unsteady separation process and induced loading produced by a canonical pitch-and-hold motion for a low-aspect-ratio wing is investigated. The flowfields are computed employing an extensively validated high-fidelity implicit large-eddy simulation (ILES) approach found to be effective for moderate Reynolds number flows exhibiting mixed laminar, transitional and turbulent regions. This high-order method furnishes a unique, scalable computational capability that has been specifically created to address the simulation issues associated with highly-nonlinear, unsteady, transitional flows inherent to MAVs.

1. Introduction

MAVs designed to operate in a complex urban environment will be required to have a number of un-precedented features to successfully carry out their mission. These vehicles will need to be highly-agile in order to avoid the man-made obstacles present in their path, or operate in the interior of buildings. They will be required to hover or perch in a location for extended periods of time to carry out proposed missions and energy-harvesting tasks to remain on station. Operations will also be carried out in an environment where gusts and crosswinds with spatial sizes of the order or greater than the vehicle size and velocity will be present. These vehicles will need to be highly-adaptable to rapidly changing environmental conditions to prevent erratic and poorly-controlled flight.

Improved understanding of the complicated, multidisciplinary physics critical to MAV design demands high-fidelity modeling capabilities. In the present paper, a high-fidelity, multidisciplinary computational solver is employed to accomplish these types of challenging simulations. For the Navier-Stokes solver spatial derivatives are computed using high-order (up to sixth-order) compact schemes and a Pade-type low-pass filter is used to ensure stability. This provides a novel, unified approach to treat mixed laminar/transitional/turbulent flows termed implicit large-eddy simulation (ILES). For the flexible-membrane wings considered, structural modeling of the elastic membrane will be accomplished using a geometrically nonlinear p-version mixed Reissner-Mindlin plate element.

Simulations for three separate canonical problems associated with low-Reynolds number MAV aerodynamics are presented. The first simulations explore the flow structure over a bio-inspired corrugated airfoil. The flow behavior was found to be strongly affected by the leading-edge geometry, due to interaction between the detached shear-layer emanating from the leading-edge and the first corrugation peak. The second set of computations calculates the flow over an aspect ratio two-membrane wing at angle-of-attack. The impact of membrane flexibility on the aerodynamic performance is investigated with emphasis on the development of mean camber, and the interaction between the dynamic motion of the

membrane and the unsteady flowfield above. Finally, simulations which describe the three-dimensional unsteady separation process and induced loading produced by a canonical pitch-and-hold motion for a low-aspect-ratio wing under low-Reynolds number conditions are performed. The effect of pitch-rate and Reynolds number on the flow structure, surface pressure and accompanying aerodynamic loads is explored.

2. Computational Framework

A. Aerodynamic Solver

The computational approach employed for the aerodynamics solves the unsteady, three-dimensional Navier-Stokes equations using a well-validated and robust high-order solver[1–3] (FDL3DI). A number of key features of the FDL3DI code make it most suitable for the present computational challenge. All spatial derivatives are computed through high-order compact or Pade-type[4] difference methods. Schemes ranging from standard 2nd-order to highly-accurate 6th-order methods are possible. To enforce numerical stability, which can be compromised by mesh stretching, boundary conditions and non-linear phenomena, a higher-order, low-pass filter is utilized. This discriminating up to 10th-order low-pass filter preserves the overall high-order accuracy of the spatial discretization while retaining stability for nonlinear applications. Careful attention to the treatment of the metric derivatives and the Geometric Conservation Law (GCL)[5] ensures higher-order accuracy on deforming and moving grids. The solver is embedded in a high-order overset-grid scheme which is utilized to provide flexibility for modeling complex geometries. It also serves as a domain decomposition mechanism for application of the high-order approach on massively-parallel, high performance computing platforms.

B. Implicit Large-Eddy Simulation Methodology

The ILES method to be used in the present computations was first proposed and investigated by Visbal, et al.[6]. The underlying idea behind the approach is to capture with high-accuracy the resolved part of the turbulent scales, while providing for a smooth regularization procedure to dissipate energy at the represented but poorly-resolved, high-wavenumbers of the mesh. In the present computational procedure the 6th-order compact difference scheme provides the high-accuracy while the low-pass spatial filters provide the regularization of the unresolved scales. All this is accomplished with no additional sub-grid scale models as in traditional LES approaches. An attractive feature of this filtering ILES approach is that the governing equations and numerical procedure remain the same in all regions of the flow. In addition, the ILES method requires approximately half the computational resources of a standard dynamic Smagorinsky sub-grid scale LES model. This results in a scheme capable of capturing, with high-order accuracy, the resolved part of the turbulent scales in an extremely efficient and flexible manner.

C. Structural Dynamics Solver

Structural modeling of the elastic membrane will be accomplished using a geometrically-nonlinear p-version mixed Reissner-Mindlin plate element. The kinematical assumptions for this model are based on Mindlins hypothesis that fibers normal to the middle plane of the plate remain straight and do not change length. The kinematic description of the plate deflections (u, v, and w in the x, y, and z directions, respectively) is given by:

$$u = \hat{u} + z\left(\theta - \frac{1}{6}\left(\theta^3 + \theta\not{v}^2\right)\right),\quad v = \hat{v} + z\left(\not{v} - \frac{1}{6}\left(\theta^2\not{v} + \not{v}^3\right)\right),\quad w = \hat{w} - \frac{1}{2}z\left(\theta^2 + \not{v}^2\right) \tag{1}$$

where θ and $\not{v}$ are rotations of fibers orthogonal to the mid-plane and $\hat{u}$, $\hat{v}$, and $\hat{w}$ are the components of the mid-plane deflection vector u_m. In the present formulation, a linear isotropic material is assumed, providing a linear relationship between the stress resultants and the strain vectors. A mixed finite element approach to solve the variational problem is developed where hierarchical shape functions of Szabó and Babŭska[7] are used to interpolate the displacements and rotations, and Legendre polynomials are used to interpolate the shear-stress variables. The resulting discrete equations are integrated in time using a Newmark acceleration method with the resulting linear equations set being solved using a parallel multi-frontal solver in conjunction with a Newton-Raphson predictor-corrector algorithm. A detailed description of the development of this technique may be found in Reference 8.

D. Aerodynamic/Structural Coupling

Coupling of the aerodynamics with the membrane response occurs through the external loads and by the resulting deflection of the membrane, which is returned to the aerodynamic grid. In the present computations a one-to-one match is used between the structural and fluid meshes. When solving fluid/structure interactions, the aerodynamic mesh must be allowed to move in accordance with the motion of the structural surface. A simple algebraic method, described in Reference 9, deforms the aerodynamic mesh to accommodate the changing membrane surface position. This grid-motion strategy has proved adequate for the membrane motions considered in the present work.

Implicit coupling of these two sets of equations is achieved by a global sub-iteration strategy. During each sub-iteration, the aerodynamic forces in the membrane equations are updated and the new surface displacements are provided to the aerodynamic solver. Using this approach the temporal lag between the aerodynamic and membrane equations may be eliminated and a complete synchronization of the aero-dynamic/structural equation set is achieved. Any factorization or linearization errors introduced in the equations may also be eliminated using this global sub-iteration procedure. The resulting coupled procedure retains second-order temporal accuracy.

3. Results

A. Flow Structure Over a Bio-Inspired Corrugated Airfoil

The corrugated airfoil shown in Figure 1a is based on the cross-sections obtained by Kesel[10], and the actual dimensions are provided in Murphy and Hu[11] and meshed using an O-mesh topology shown in Figure 1b. A fine-mesh is generated using a distribution of 853 points circumferentially around the airfoil surface (ξ-direction) in order to capture the flow structure around the sharp corners of the peaks and valleys, and 328 points radially-outward (η-direction) to create a two-dimensional (2D) computational plane. This grid is uniformly extruded along the airfoil span (ζ-direction) to include a total of 125 points and a span-length of 0.2.

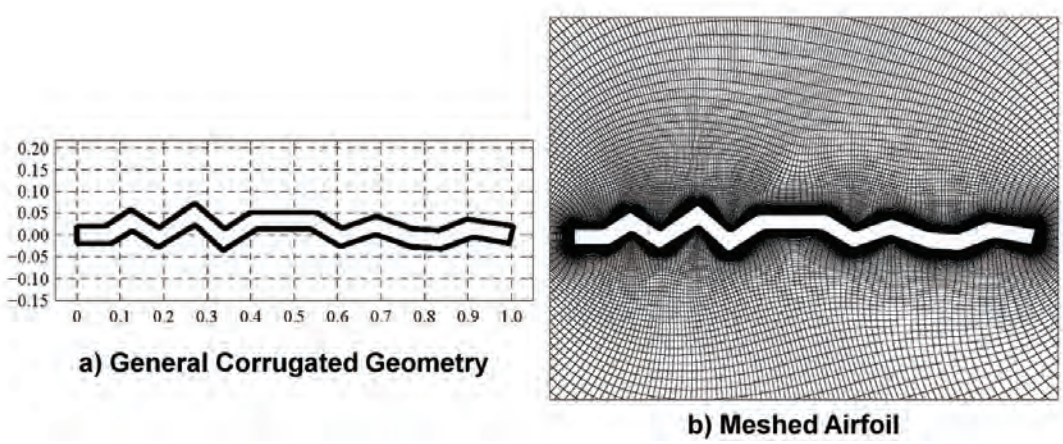


Figure 1. Geometry and computational mesh

1. Effect of Angle-of-Attack at Fixed-Reynolds Number

In order to investigate the effect of angle-of-attack on the aerodynamic forces and flow structure, simulations are obtained over a range of angles-of-attack between 5° and 10° at Re_c=3.4×10^4 . The streamlines and turbulent kinetic energy (TKE) for the time-mean computation of the baseline airfoil are provided in Figure 2. The streamlines at α=5° show a flow structure with slowly rotating vortices trapped in the corrugation valleys. Recirculation also occurs within the detached region leading the first peak. Development of a large separation bubble (LSB) is revealed as the angle-of-attack is increased. Little change is observed in the streamlines as the angle-of-attack is increased to 8°, but the turbulent region of flow grows thicker and the detached shear-layer moves further away from the first corrugation peak. The flow remains attached to the virtual profile formed from the airfoil body and trapped flow in the corrugation valleys, but the high-speed, free-stream flow is pushed further away from the first corrugation peak with the increasing level of detachment at the leading-edge and transition occurs further away from the airfoil body as indicated by the plots of TKE. At α=9°, the detached region leading the first peak becomes so thick the recirculation is unable to pull the free-stream close to the airfoil

body. The first dividing streamline emanating from the leading-edge travels further way from the first peak and the trapped vortices in the first and second valleys begin to merge. At this point, a large-scale separation bubble suddenly spans the entire chord length and rapidly grows as angle-of-attack is increased to 10°. The turbulent region at this angle-of-attack is much larger and further away from the airfoil body.

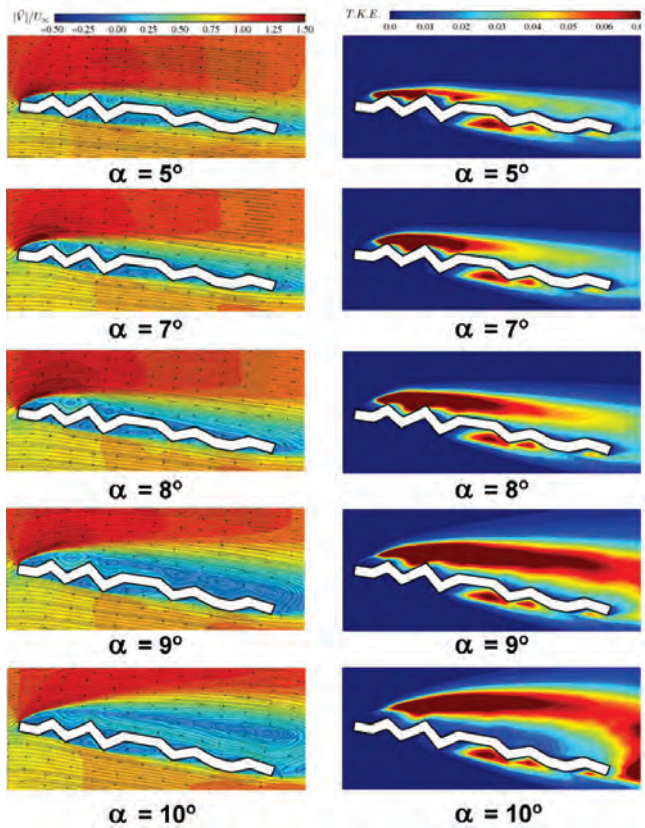


Figure 2. Formation of separation bubble with increasing angle-of-attack

Aerodynamic forces are tabulated for the angles-of-attack tested at Re_c=3.4×10^4 in Table 1. For the baseline geometry, the lift values increase with increasing angle-of-attack until large-scale separation occurs at α=9°, where a large drop in the lift-to-drag ratio is observed. Drag increases due to the growing wake region, while the lift-to-drag ratio decreases with increasing angle-of-attack indicating the drag rises more rapidly than lift. The magnitude of the pitching moment increases with increasing angle-of-attack.

Table 1. Aerodynamic forces with angle-of-attack (Re_c=3.4×10^4)

α	C_L	C_D	L/D	C_M
5°	0.0646	0.0833	7.255	−0.1684
7°	0.7567	0.1113	6.799	−0.2018
8°	0.8242	0.1286	6.409	−0.2300
9°	0.7901	0.1600	4.938	−0.2536
10°	0.7276	0.1757	4.141	−0.2833

2. Effect of Geometry

The baseline corrugated airfoil is compared to a flat plate with identical leading-edge geometry and a modified corrugated geometry with sharper corners and taller corrugation peaks at α=10°. Streamlines and TKE for all cases are shown in Figure 3. For the flat plat and baseline geometries an almost identical detached shear-layer and a very large separation bubble form beginning from the leading-edge. For the modified geometry the steeper first peak penetrates further into the detached region following the leading-edge corner, and reduces the distance between the detached shear-layer and the airfoil body. The taller peak results in a greater level of interaction between the coherent vortices and the

leading-edge recirculation, leading to an earlier onset of turbulence. The more energetic flow is better able to advance against adverse pressure gradients, reducing the degree of separation to a flow structure similar to the baseline geometry at α=8°.

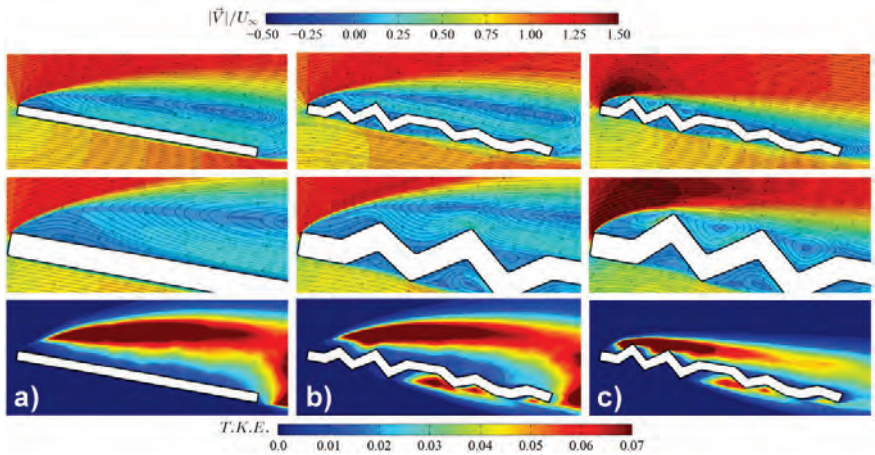


Figure 3. Comparison of the a) flat plate, b) baseline corrugated profile, and c) modified airfoil at α=10° and Re_c=3.4×10^4

The aerodynamic forces for the flat plate baseline and modified corrugated airfoil are compared in Table 2. Compared to the flat plate and baseline geometry, the modified geometry produces a higher lift, with a drag coefficient comparable to the baseline profile resulting in a much higher lift-to-drag ratio. However, the pitching moment increased in magnitude for the modified airfoil, likely resulting from flow interaction with the steeper corrugation peaks at the leading-edge. The increased aerodynamic performance is a result of the reduced flow separation observed in Figure 3. Moving the corrugations further into the separated shear layer appears to help entrain the free-stream flow, and the improved flow entrainment results in a higher lift-coefficient while the drag is not adversely affected. The steeper corrugations only protrude into slow-moving flow, preventing the drag associated with an increased cross-sectional area from overcoming the drag reduced through decreasing the flow separation. These modifications in the geometry suggest the flow behavior is strongly affected by the leading-edge configuration, and geometry can be optimized for different angles-of-attack by contracting or extending the leading- edge corrugations.

Table 2. Aerodynamic forces of alternate geometries compared to baseline (Re_c=3.4×10^4)

	α	C_L	C_D	L/D	C_M
Geometry	10°	0.7053	0.1725	4.0887	−0.2892
Flat plate	10°	0.7276	0.1757	4.141	−0.2833
Modified corrugated	10°	0.8760	0.1755	4.991	−0.3111

B. Aerodynamic Performance of an Aspect Ratio Two-Membrane Wing

The membrane wing to be studied is based on the experimental model of Rojratsirikul, et al.[13]. The wing tested was rectangular with an aspect ratio, *AR*=2. The wing consisted of a rigid frame with an airfoil shaped cross-section made from stainless steel, and was covered with an attached membrane sheet of latex rubber. The latex sheet had a thickness of t=0.2*mm*, Young's modulus of 2.2MPa, and a density, $\rho_m = 1gr/cm^3$. The experiments were performed over an angle-of-attack range from α=0 to 25 degrees, and Reynolds number based on chord length of *Re*=24,300. Computations for an angle-of-attack α=16° are carried out in the present work.

In order to delineate the influence of the development of mean camber and the dynamic motion of the membrane surface for the aspect-ratio two-membrane wing, computational results for a rigid, flat-membrane wing, a rigid, cambered wing with the shape of the mean deflection for the flexible case, and the flexible-membrane wing are compared. To understand the three-dimensional structure of the flow, Figures 4a-c show the pressure coefficient and limiting streamline pattern on the upper-surface of the membrane wing. For the rigid, flat-membrane wing, separated flow covers the majority of the wing with the attachment line reaching the trailing-edge over the mid-section of the wing, Figure 4a. Two stable foci are observed in the separated flow region. Low-pressure associated with the separated flow covers a significant portion of

the wing, with lower pressures located under the two distinct stable foci, Figure 4a. Two narrow low-pressure regions located along the tips of the wing are the low-pressure imprint of the tip vortices on the wing-surface.

Introducing the mean cambering of the flexible-membrane wing results in a decrease in the size of the main separated flow region, with primary flow attachment occurring upstream from the trailing edge, see Figure 4b. The two foci observed in Figure 4a have moved upstream and outboard, and are now located in the upstream corners of the wing. Lower values of the pressure coefficient are observed in the separated flow region. Figure 4c presents the results for the flexible case. A further reduction in the size and extent of the separated region is observed with flow attachment now located near the mid-chord of the wing. Even lower values of the pressure coefficient in the separated flow region are observed, with strong low-pressure in the upstream corners of the wing.

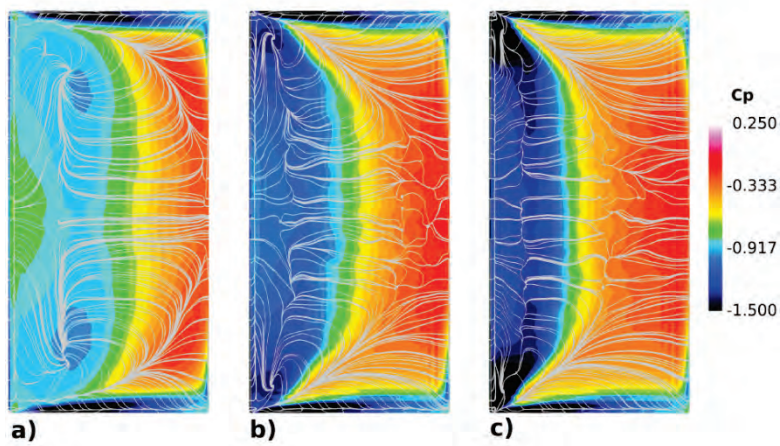


Figure 4. Surface pressure coefficient and limiting streamline pattern: a) rigid, flat-b) rigid, mean cambered, and c) flexible-membrane wing

Figure 5 displays the dynamic response of the membrane surface over approximately one-cycle of the structural motion (a-b-c-d-e-a). The upper-half of Figure 5 shows the total membrane deflection. From these figures the global structural response is seen to be in the first-mode. The region of maximum deflection moves downstream and subsequently back upstream over one-cycle of the membrane motion. If the mean first-mode deflection is subtracted from the total-surface deflection, a second stream-wise structural mode is more clearly discerned in the lower-half of Figure 5. These observations concerning the membrane dynamics are consistent with those made by Rojratsirikul, et al.[14] for an angle-of-attack, α=17°.

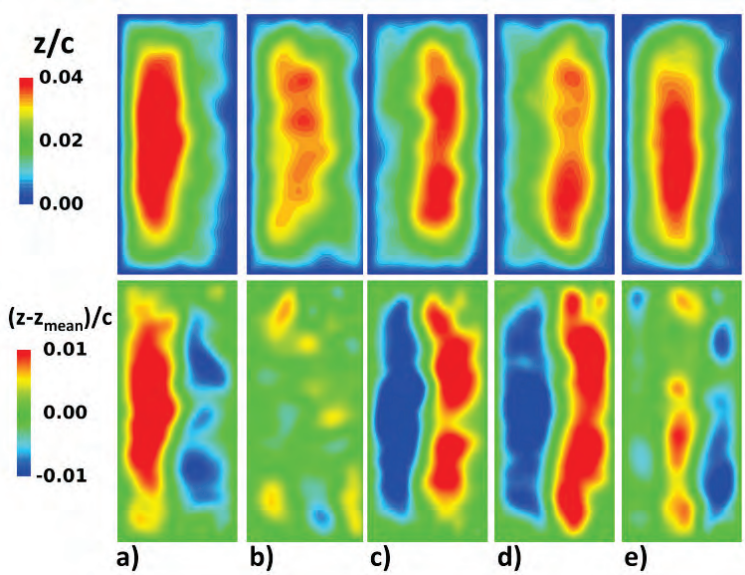


Figure 5. Total surface defection z/c (upper) and difference from the mean deflection ($z-z_{mean}$)/c (lower): a) τ=0.0, b) τ=0.25, c) τ=0.5, d) τ=0.75 e) τ=1.0

The three-dimensional, unsteady flow structure over the flexible-membrane wing is visualized in Figure 6 by plotting an iso-surface of the Q-criterion[15] colored by pressure coefficient. The flow separates from the leading-edge and the separated shear-layer rolls up into a series of longitudinal vortical structures, Figure 6b. A complex amalgamation of these longitudinal structures is formed, Figures 6c-d, and is subsequently shed and convects downstream, Figures 6e-b. These longitudinal vortical structures are subject to span-wise instabilities and breakdown into a complex collection of small-scale structures as they convect downstream. Such a distinct shedding process is not clearly discerned in the rigid, flat-membrane wing case, and results from the influence of the membrane motion on the vortex dynamics of the separated flow.

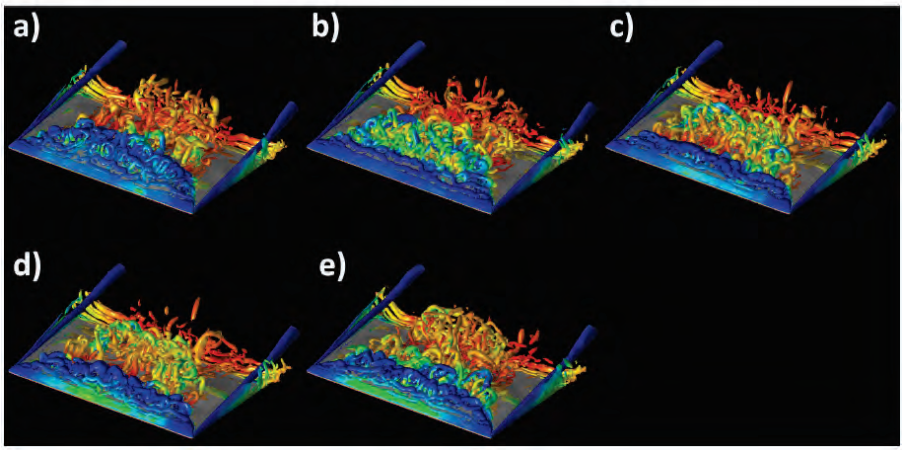


Figure 6. Iso-surfaces of Q-criterion colored by pressure coefficient for the flexible-membrane wing: a) τ=0.0, b) τ=0.25, c) τ=0.5, d) τ=0.75 e) τ=1.0

The modification of the flow structure, due to the flexibility of the membrane wing, impacts the aerodynamic performance of the wing. Table 3 lists the aerodynamic coefficients for each of the cases computed. The rigid, cambered wing demonstrates a slight improvement in the total lift coefficient, albeit at the cost of increased drag and reduced L/D. The flexible case performs slightly better than the mean cambered airfoil, indicating a favorable effect due to the dynamic motion of the membrane. The pitching moment exhibits a continual reduction of the nose-down pitching moment when going from the rigid, flat-membrane wing to the flexible-membrane wing.

Table 3. Aerodynamic coefficients

	CL	CD	L/D	Cm_y
Rigid Flat Wing	0.965	0.235	4.112	−0.160
Rigid Cambered Wing	1.020	0.269	3.794	−0.140
Flexible Wing	1.026	0.263	3.903	−0.123

C. Flow Structure and Unsteady Loading Over a Low-Aspect-Ratio Pitching Wing

Computations are carried out for a low-aspect-ratio plate wing that has an aspect ratio $2S/c$=2, a uniform thickness t/c=0.016 and all edges squared-off. The Reynolds number varies over the range $10^3 \leq Re_c \leq 4\times10^4$.

Computations are performed on a stretched Cartesian mesh containing approximately 30 million grid-points. The grid consisted of five zones, and had overall dimensions of 440×469×146 in the stream-wise, normal and span-wise directions, respectively. Clustering was provided near the plate in order to capture the near-field, complex, vortical structure generated by the pitching motion.

The surface velocity components (u_s, v_s, w_s) were determined from the imposed pitching motion (i.e., as a function of the instantaneous angle-of-attack, rotation-rate and pitch-axis location). In order to avoid discontinuities in the angular acceleration, the angle-of-attack was prescribed employing a modified version of the ramping function of Eldredge, et al.[16]. The angle-of-attack, α, is given by

$$\alpha(t) = \frac{\Omega_0^+}{2a}\ln\left[\frac{\cosh\left(a\left(t-t_1\right)\right)}{\cosh\left(a\left(t-t_2\right)\right)}\right] + \frac{1}{2}\alpha_{max} \tag{2}$$

In this equation, a controls the smoothing of the angular acceleration and deceleration, $\Omega_0^+ = \alpha_{max} / (t_2 - t_1)$ is the nominal non-dimensional pitch rate, α_{max} denotes the maximum angle-of-attack, and t_1 and t_2 define the nominal pitching time interval prior to smoothing. In all of the cases to be presented, t_1=0.5, a=11, and the pitch axis is located at the wing quarter chord. The instantaneous angular rate can be obtained by straightforward differentiation of Equation 2.

Pitching simulations were started from the corresponding computed static solutions at zero-degree angle-of-attack. A very small non-dimensional time-step $\Delta t U_\infty / c = 6.25\times10^{-5}$ was prescribed in order to provide sufficient temporal resolution of the transitional fine-scale flow evolution. This value of Δt corresponded to 64,000 time-steps during the pitching interval $\tau_2 - \tau_1$ for the highest pitch-rate considered.

Results for the highest pitch-rate $\Omega_0^+ = 0.196$ with $Re_c = 10^4$ are shown in Figures 7 and 8. Figure 7 displays the three-dimensional unsteady flow structure at selected instants during the pitch-up motion by means of an iso-surface of constant total pressure. One can observe the formation of a leading-edge vortex, which is pinned at the corners of the wing. The LEV system evolves into the arch-type vortical structure previously found by Visbal[17] for a plunging wing. As the angle-of-attack increases, one can also see the formation, breakdown and collapse of the wing-tip vortex. Comparison of the instantaneous computed flow-structure with recent experiments of Yilmaz and Rockwell[18] is provided in Figure 8 at two angles-of-attack. Good overall agreement is observed between the experiments and computations, in terms of the arch and tip vortices.

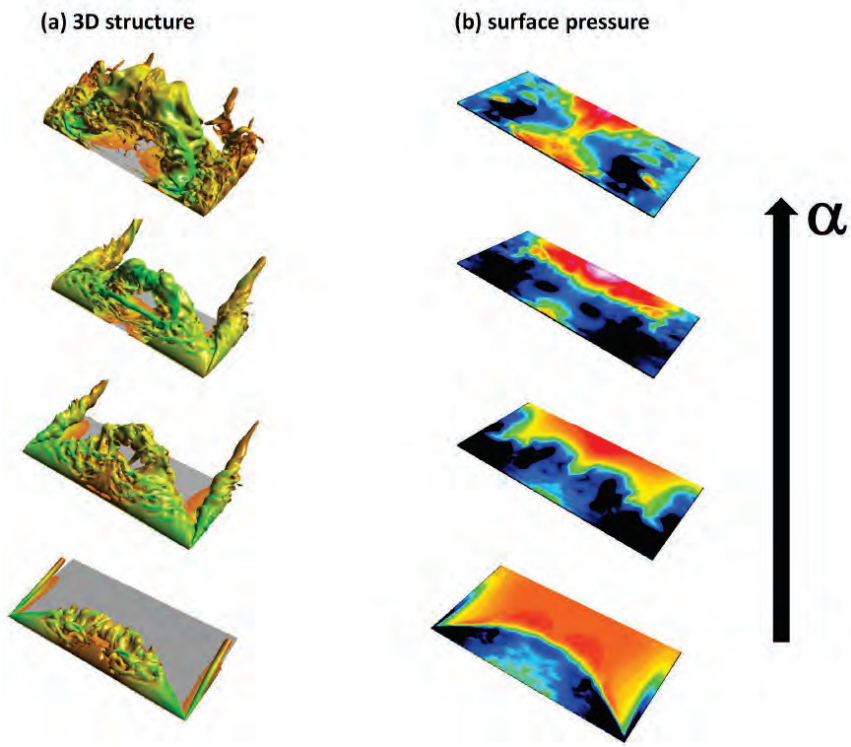


Figure 7. Computed evolution of instantaneous three-dimensional flow structure and surface pressure for $Re_c = 10^4$ and $\Omega_0^+ = 0.196$

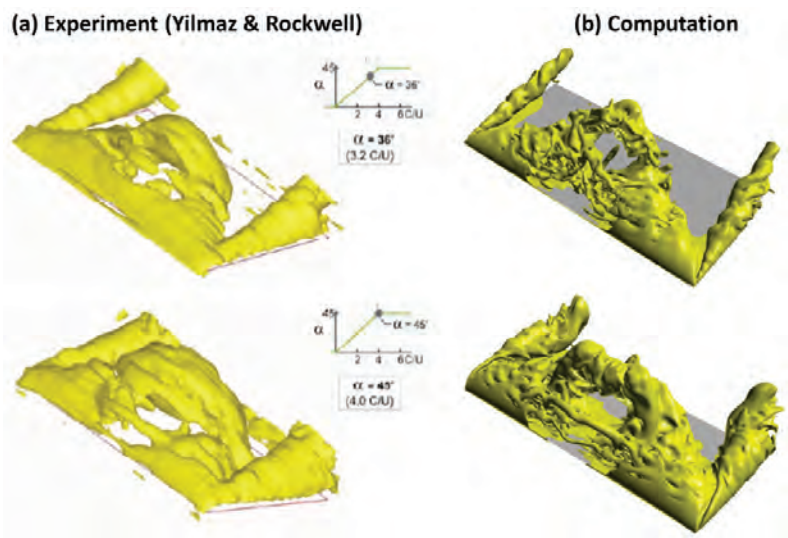


Figure 8. Comparison of computed and experimental[18] three-dimensional flow structure for Re_c=10^4 and Ω_0^+ = 0.196

The evolution of the region of low-pressure on the wing upper-surface is shown in Figure 7. Initially, lower- pressures are observed under the developing leading-edge vortex system, as well as under the tip vortices. Subsequently, concentric regions of low-pressure appear and move towards the wing centerline corresponding to the legs of the evolving arch-type vortex. At high-angles-of-attack, a large-scale circulatory pattern is established over the wing, which results in two well-defined regions of low-pressure.

In order to study the effect of pitch-rate on the flow structure and unsteady loads, three values of Ω_0^+ were considered $(\Omega_0^+ = 0.049,\ 0.098,\ 0.196)$. Only the effects of pitch-rate on the lift and pitching moment coefficient histories are included (Figure 9a,b). Several trends are apparent in the lift distribution as a function of pitch-rate. In all cases, there is a non-circulatory spike during the period of acceleration with a magnitude roughly proportional to Ω_0^+. This effect has also been documented for a two-dimensional pitching airfoil.[19]

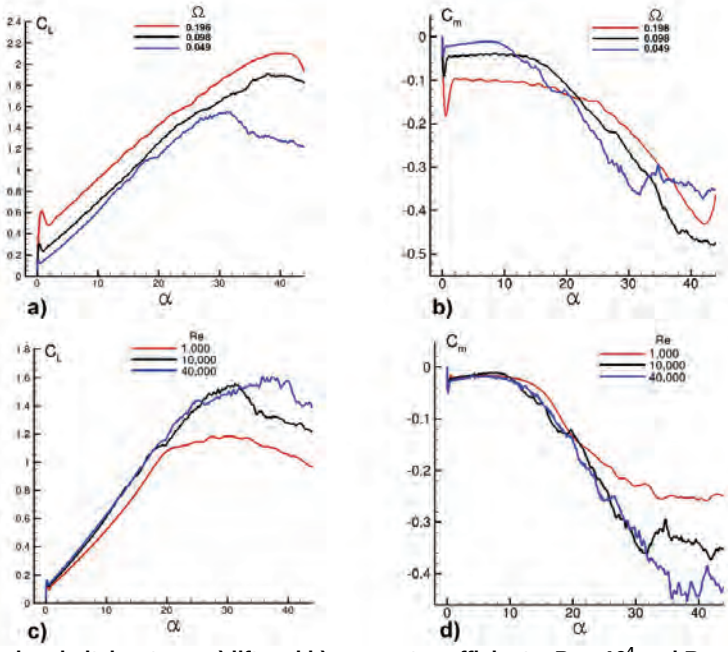


Figure 9. Effect of non-dimensional pitch-rate on a) lift and b) moment coefficients, Re_c=10^4 and Reynolds number on c) lift and d) moment coefficients, Ω_0^+ = 0.049

Following this initial surge, the lift distribution becomes fairly linear, with a slope close to the static value of π but shifted by an offset which is again proportional to pitch-rate. This offset can be attributed to the pitch-induced chord-wise camber.[20] The maximum value of C_L and its angular delay both increase with pitch-rate. The values of $C_{L_{max}}$ corresponding to the three pitch-rates are approximately 1.54, 1.89 and 2.1. Even for the smallest pitch-rate, there is a significant increase in maximum-lift which is almost twice its static value. The moment coefficient exhibits an initial angular acceleration-induced spike, followed by a fairly flat distribution which again is compatible with the rotation-induced camber effect. This stage is followed by a rapid drop in C_M which takes place considerably before the angle of maximum-lift. There is also an angular delay in the onset of pitching moment stall with increasing Ω_0^+.

The effect of Reynolds number on the flow structure and loading was investigated for the lowest pitch-rate $(\Omega_0^+ = 0.049)$ by considering three values of Re_c over the range of interest. Only the behavior of the lift and pitching moment coefficients as a function of Reynolds number is displayed in Figure 9c,d. One can observe that there are significant effects of Reynolds number, not only on maximum-lift but also on its angular position. There appears to be a consistent delay in lift-stall as the Reynolds number increases. The C_L distribution for Re_c=10^3 has a lower slope and subsequently displays a much smoother character. The pitching moment coefficient also exhibits a bigger drop with increasing Re_c.

4. Conclusion

An Implicit Large-Eddy Simulation (ILES) approach was applied to the prediction of transitional flow for canonical problems associated with biologically-inspired MAVs. ILES is found to be particularly attractive for these types of low-Reynolds number flows that exhibit mixed laminar, transitional and turbulent regions for which conventional high-Reynolds number analysis tools may not be adequate. The computational framework employed, FDL3DI (and its extension for aeroelastic problems FDL3DIAE), contains a robust, high-order Navier-Stokes solver which adapts overset-grid technologies to treat the most complex geometries. This framework is highly-scalable and easily-portable to a variety of parallel computing platforms.

Computations for three separate canonical problems have been presented in this work. The first problem computed the flow over a bio-inspired corrugated airfoil. The formation of large-scale separation was studied at Re_c=3.4×10^4 where the flow was found to remain attached to a virtual profile formed by vortices in the corrugation valleys through α=8°. As the angle-of-attack was increased, the detached shear-layer moved further away from the airfoil body and the free-stream flow experienced decreased interaction with the first corrugation peak. Consequently, the effect of entrainment from the valley vortices on the free-stream weakened and a large separation bubble rapidly formed. A modified corrugated airfoil, characterized by sharper corners and steeper corrugation peaks at the leading-edge was also computed. The taller peak resulted in a greater level of interaction between the coherent vortices and the leading-edge recirculation, resulting in an earlier onset of turbulence. The size of the separation bubble was significantly reduced and the aerodynamic forces were more favorable.

For the second problem presented, computations were performed for an aspect-ratio two membrane-wing at an angle-of-attack, α=16° and a Reynolds number, Re=24,300. Three different wings were considered: a rigid, flat membrane-wing; a rigid, cambered wing, with the shape of the mean deflection for the flexible-wing; and a flexible membrane-wing. The flexible membrane-wing produces a reduced separation region due to the fluid-structure interaction between the structural dynamics of the membrane and the unsteady flow over the wing. Enhanced lift is obtained in the flexible case albeit at the cost of increased drag and a lower lift-to-drag ratio. The nose-down pitching moment is reduced. The impact of flexibility on the aerodynamic performance was demonstrated to come primarily from the development of mean camber, with an additional effect arising from the interaction between the dynamic motion of the membrane and the unsteady flowfield above.

The final problem presented addresses the flow structure and unsteady loading arising over an aspect-ratio two- wing subjected to high-amplitude pitching maneuvers. A canonical pitch-and-hold motion with maximum incidence of 45° is prescribed over time intervals ranging from four to sixteen convective time-scales. The Reynolds number based on the wing-chord was 10^4. The highest pitch-rate considered corresponded to recent experiments by Yilmaz and Rockwell. Good overall agreement is observed between the computed and experimental vortical flow features. The maximum value of C_L and its angular delay both increased with pitch-rate. The lift distribution is observed to be predominantly linear, with a slope close to the static value of π, but shifted by an offset proportional to pitch-rate. This offset can be attributed to the pitch-induced chord-wise camber. The moment coefficient exhibits and initial angular acceleration-induced spike followed by a fairly-flat distribution which again is compatible with the rotation-induced camber effect. This stage is followed by a

rapid drop in C_M which takes place considerably before the angle of maximum lift. There is also an angular delay in the onset of pitching moment stall with increasing Ω_0^+.

5. Significance to the DoD

This project harnesses the attributes of recently-developed high-fidelity, multidisciplinary codes to provide accurate, detailed simulations of the aerodynamics associated with bio-inspired fliers. The high-order overset method employed in the present project provides a unique, scalable computational capability that has been specifically created to address the simulation issues associated with highly-nonlinear, unsteady, transitional flows inherent to MAVs. Likewise, computational structural modules that treat large structural deflections and nontraditional structural materials have been developed and incorporated into the context of this computational framework. The computational techniques demonstrated in this project will provide MAV designers with unique tools and understanding to accelerate the development of unconventional, biomimetic MAV concepts.

Acknowledgments

This work was sponsored by AFOSR under a task monitored by Dr. Doug Smith, and supported in part by a grant of HPC time from the DoD Shared Resource Centers at the US Air Force Research Laboratory, Wright-Patterson AFB, OH and the US Army Corps of Engineers Research and Development Center at Vicksburg, MS.

References

1. Gaitonde, D. and M. Visbal, "High-Order Schemes for Navier-Stokes Equations: Algorithm and Implementation into FDL3DI." *Tech. Rep. AFRL-VA-WP-TR-1998-3060*, Air Force Research Laboratory, Wright-Patterson AFB, 1998.

2. Gaitonde, D. and M.Visbal, "Further Development of a Navier-Stokes Solution Procedure Based on Higher-Order Formulas." *AIAA Paper 99-0557*, January 1999.

3. Visbal, M. and D. Gaitonde, "High-Order Accurate Methods for Complex Unsteady Subsonic Flows." *AIAA Journal*, Vol. 37, No. 10, pp. 1231–1239, 1999.

4. Visbal, M.R. and D.V. Gaitonde, "On the Use of Higher-Order Finite-Difference Schemes on Curvilinear and Deforming Meshes." *Journal of Computational Physics*, Vol. 181, pp. 155–185, 2002.

5. Visbal, M. and R. Gordnier, "A High-Order Flow Solver for Deforming and Moving Meshes." *AIAA Paper 2000-2619*, June 2000.

6. Visbal, M. R., P.E. Morgan, and D.P. Rizzetta, "An Implicit LES Approach Based on High-Order Compact Differencing and Filtering Schemes (Invited)." *AIAA-2003-4098*, June 2003.

7. Szabó, B. and I. Babŭska, *Finite Element Analysis*, John Wily & Sons Ltd, 1997.

8. Attar, P.J., "Some results for approximate strain and rotation tensor formulations in geometrically non-linear Reissner-Mindlin plate theory." International Journal of Non-Linear Mechanics, Vol. 43, 2008, pp. 81–99.

9. Melville, R.B., "Dynamic Aeroelastic Simulation of Complex Configurations Using Overset Grid Systems." *AIAA 2000-2341*, June 2000.

10. Kesel, A B., "Aerodynamic Characteristics of the Wings and Body of a Dragonfly." *Journal of Experimental Biology*, Vol. 203, pp. 3125–3135, 2000.

11. Murphy, J.T. and H. Hu, "An Experimental Study of a Bio-Inspired Corrugated Airfoil for Micro Air Vehicle Applications." *Journal of Experimental Fluids*, Vol. 49, pp. 531–546, 2010.

12. Jeong, J. and F. Hussain, "On the Identification of a Vortex." *Journal of Fluid Mechanics*, Vol. 285, pp. 69–94, 1995.

13. Rojratsirikul, P., M.S. Genc, Z. Wang, and I. Gursul, "Flow-induced vibraions of low aspect ratio rectangular membrane wings." *Journal of Fluids and Structures*, Vol. 27, No. 11, pp. 1296–1309, 2011.

14. Rojratsirikul, P., Z. Wang, and I. Gursul, "Unsteady Aerodynamics of Low Aspect Ratio Membrane Wings." *AIAA-2010-0729*, January 2010.

15. Jeong, J. and F. Hussain, "On the identification of a vortex." *Journal of Fluid Mechanics*, Vol. 285, pp. 69–94, 1995.

16. Eldredge, J., C. Wang, and M. Ol, "A computational study of a canonical pitch-up, pitch-down maneuver." *AIAA-2009-3687*, June 2009.

17. Visbal, M.R., "Three-Dimensional Flow Structure on a Heaving Low-Aspect-Ratio Wing." *AIAA-2011-219*, January 2011.

18. Yilmaz, T.O. and D. Rockwell, “Flow structure on finite-span wings due to pitch-up motion.” *Journal of Fluid Mechanics*, 2011(to appear).

19. Visbal, M.R., “Dynamic stall of a constant-rate pitching airfoil.” *Journal of Aircraft*, Vol. 27, No. 5, pp. 400–407, 1990.

20. Visbal, M.R. and R.E. Gordnier, “Pitch-Rate and Pitch-Axis Location Effects on Vortex Breakdown Onset.” *Journal of Aircraft*, Vol. 32, No. 5, pp. 929–935, 1995.

Modeling Optical Distortions in a Turret Shear Layer for Feedback Flow Control

Christopher Porter, Robert Decker, and Jürgen Seidel
US Air Force Academy (USAFA), Colorado Springs, CO
{christopher.porter, robert.decker, jurgen.seidel}@usafa.edu

Abstract

Reduced-order modeling of the optical aberrations caused by density variations in the separated shear layer behind a turret is discussed in this paper. Spatially and temporally well-resolved Delayed Detached-Eddy Simulations (DDES) using Cobalt were performed to investigate the flow-field. Both a turret with a conformal window and a turret with a flat window have been investigated. The flow-field was analyzed using Proper Orthogonal Decomposition and Dynamic Mode Decomposition. The analysis revealed that the optical distortions can be characterized into three main components: static lensing due to the density gradient around the turret, tip/tilt due to the density gradient of the separated shear layer, and the aberrations due to unsteady flow structures. Using these results, a reduced-order model was developed using pressure sensors around the aperture to capture the flow-field. The results show that using ten sensors, the optical distortions can be predicted with an error of less than 0.1%.

1. Introduction

An initially collimated beam passing through a medium is distorted by changes in the refractive index, n, in the medium. For air, the refractive index is primarily a function of the fluid density, described by the Gladstone-Dale relation,

$$n\,(x, y, z, t) = 1 + k_{GD}\rho\,(x, y, z, t), \tag{1}$$

where the proportionality constant is the Gladstone-Dale constant k_{GD}=2.289 · 10^{-4}m^3/kg and ρ is the fluid density.[1–3] A measure of the aberration of a beam is the optical path length,

$$OPL\,(x, y, z, t) = \int n\,(x, y, z, t)dl, \tag{2}$$

i.e., the path integral of the index of refraction along the beam. The wave-front distortions ultimately result in reduced focus and beam intensity in the far-field. This is particularly detrimental for airborne optical systems, where density variations occur close to the beam aperture due to the motion of the aircraft. The distortions away from the ideal diffraction limit can greatly reduce the usefulness of an airborne optical system.

To highlight the wave-front distortions, the optical path difference,

$$OPD(x,z,t;y) = OPL(x,z,t;y) - \overline{OPL\left(x,z,t;y\right)}^{x,z}, \tag{3}$$

is typically used, where it is assumed that the beam propagates in the y-direction and the spatial average is performed over the beam aperture. In addition, it is customary to remove the mean wave-front tip and tilt, which are the inclination of the mean wave-front with respect to the beam propagation direction.

Aero-optical aberrations are typically categorized in two main groups. The first category, associated with fluctuations in the flow-field of the scale of the aperture, include bore-sight and tracking errors. These aberrations are amenable to correction by current adaptive optical systems due to their relatively large length-scales and concomitant slow time-scales. The second category includes errors such as beam spreading, scintillation and reduction of resolution, contrast, etc., which are caused by the smaller scales in the flow, including the coherent motion in the flow close to the aperture and the coherent structures typically associated with turbulent flows. Aberrations caused by these unsteady flow structures cannot currently be controlled using optical systems and new approaches are necessary for their mitigation.

A relatively new approach to understanding and controlling the optical aberrations observed in airborne applications is to directly control the coherent structures in the flow-field over the aperture to minimize their strength. It has been shown that these coherent structures are responsible for the vast majority of the optical distortions in aero-optical systems.[2] Open-loop

flow control, where small amplitude disturbances are introduced into the flow at sensitive locations, utilize the flow's inherent instabilities to amplify the disturbance input and achieve large global changes in the flow-field. Both passive and active flow control system are possible. A well-known example of the former is the delay of separation on airfoils at large angles-of-attack by using vortex generators to introduce small vortices upstream of the separation point to keep the flow attached by energizing the boundary-layer. In active systems, energy is typically added or subtracted from the flow by active introduction of mass using, for example, micro-jets.

Previous research has shown that a flow-state database is required to design an efficient feedback flow control system. In particular, sensor placement, including both the number and location of the sensors, the development of the flow-state estimator, as well as the design of the controller heavily rely on accurate instantaneous information about the flow-field.[4] Such a flow-state database has to contain flow-field data that capture the essential physical processes to be controlled. In addition, the effect of actuation on the flow-field has to be included in the database, since feedback control, if effective, will alter the flow-field significantly. While the ultimate flow-state is unknown before a controller is employed, important insights into the dynamics of the flow can be gleaned from the flow's response to periodic forcing, especially from the transient behavior that occurs when the forcing is turned on or off.

With this database, a reduced-order model (ROM) of the flow dynamics can be constructed using state-of-the-art modeling techniques.[5,6] The main objective of this ROM is to capture the relevant flow dynamics with a small number of variables, reducing the enormous number of degrees-of-freedom in a typical simulation of experiment to a manageable set of data for controller development. In addition, this ROM is also used as the basis for sensor placement studies and development of flow-state estimation schemes.

In this paper, the simulations for the development of such a database of the flow around a three-dimensional turret are described. First, the numerical method is discussed and some representative results of the simulations are shown. Second, the analysis of the data using Proper Orthogonal Decomposition and Dynamic Mode Decomposition is shown. Finally, the development of a reduced-order model based on surface flow measurements is discussed.

2. Numerical Method

The computational fluid dynamic (CFD) simulations were performed using COBALT V5.0 and V5.2 from Cobalt Solutions, LLC, an unstructured finite-volume code developed for the solution of the compressible Navier-Stokes equations. The basic algorithm is described in Strang, et al., although substantial improvements have been made since then.[7] The numerical method is a cell-centered finite-volume approach applicable to arbitrary cell topologies. The spatial operator uses the exact Riemann Solver of Gottlieb and Groth, least squares gradient calculations using QR factorization to provide second-order-accuracy in space, and TVD flux limiters to limit extremes at cell faces.[8] A point implicit method using analytic first-order Jacobians is used for advancement of the discretized system. For time-accurate computations, a second-order-accurate method with Newton sub-iterations is employed. As in earlier investigations, Delayed Detached-Eddy Simulation (DDES) are used to model the effect of turbulence at length scales smaller than the grid-size.[9]

For the flow-field analysis, Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD) have been utilized. POD is designed to capture the most energy in the flow-field with the fewest modes and then compute-time coefficients for these modes to describe their temporal behavior, while DMD captures spatial modes for a given temporal frequency.[18]

2.1 Computational Domain and Discretization

The basic geometry under consideration is a three-dimensional turret. The base cylinder has a diameter of D=30.5cm and is H_c=10cm tall. The radius of the optical aperture is R_A=5cm, and for the flat window turret, the flat has a radius of $R_{A'}$=10.5cm to allow for the introduction of pressure sensors and actuators around the aperture (Figure 1).

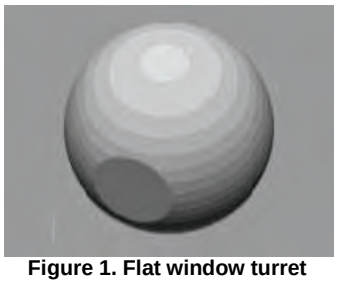


Figure 1. Flat window turret

Figure 2 shows the grid in the center plane and the refinement around the turret. In Figure 3, the grid in the junction plane (x,z-plane) where the cylinder and the hemisphere intersect is shown. For the simulations presented here, the aperture of flat window turret is located at an azimuthal angle of Az=120 and an elevation angle of El=50°.

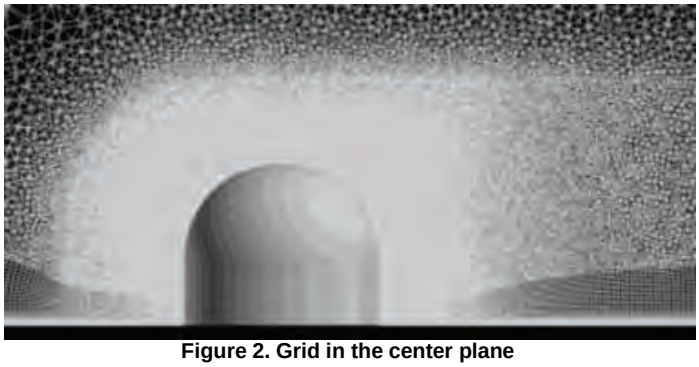


Figure 2. Grid in the center plane

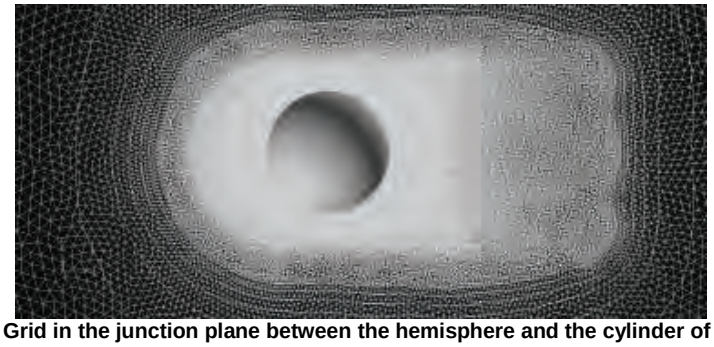


Figure 3. Grid in the junction plane between the hemisphere and the cylinder of the turret

The SimCenter software suite was used to generate the computational domain and grids.[10] The turret was surrounded by a refinement region that extended x/D=1 upstream and laterally of the turret and x/D=3 downstream to capture the shear layer and recirculation region behind the turret. The coordinate system is oriented such that the x-axis points downstream, the y-axis vertically up, and the z-axis is in the span-wise direction. The inflow boundary is x/D=7.5 upstream of the turret, while the downstream boundary is x/D=15 downstream to allow structures to dissipate before reaching the boundary. In the y-and z-directions, the grid extends to y/D=7.5 and z/D=15 to allow for sufficient distance to the boundary to minimize reflections.

The grid-spacing on the turret was ds=1mm on the turret surface and $dn=2\times10^{-4}$ mm in the wall-normal direction, which resulted in an average $y^{+}<1$. The grids had approximately 70M cells.

A time-step study showed that $\Delta t=5\cdot10^{-5}$s was sufficient to capture the relevant flow structures.

To study the optical properties of the flow-field, the flow-field data were written out on beam-grids (created using Cobalt's tap feature) and analyzed using Equations 1–3. For the conformal turret, five different tap-grids could be applied in a single computation (Figure 4). For the flat window turret, the tap-grid is fixed to the flat aperture and separate simulations are necessary for different viewing angles.

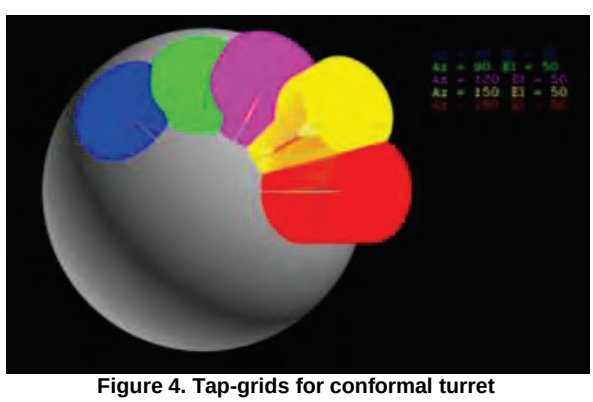


Figure 4. Tap-grids for conformal turret

2.2 Computational Parameters

The inflow Mach number was maintained at M=0.4. Standard atmospheric conditions at an altitude of 15,000 ft were prescribed at the inflow to match flight experiments conducted by the University of Notre Dame. The resulting Reynolds number, based on the turret diameter is approximately Re=900,000.

Compute-times, including the data input/output (I/O), were measured at approximately 92 CPUs/iteration or 1.3μs/cell/iteration.

3. Results

The flow-field around the conformal-windowed turret exhibits several known main features, from the horseshoe vortices in front of the turret to the horn vortices in the wake. An instantaneous Q-iso-surface of the flow-field around the turret is shown in Figure 5. At the front of the turret, several horseshoe vortices can be clearly seen for this Reynolds number.[11] As the flow separates from the turret, the two dominant counter-rotating horn vortices can also be seen. Along the center plane, a small depression in the iso-surface of Q exists, corresponding to the downward component of the flow resulting from these two horn vortices. The other dominant flow feature, especially with respect to aero-optics is the roll-up of a Kelvin-Helmholtz shear layer towards the top of the turret as the flow separates. Figure 6 shows an instantaneous slice of the density field along the centerline of the turret. The existence of a Kelvin-Helmholtz shear layer should be evident by low-density cores associated with the concomitant low-pressure core inside individual coherent vortices that roll-up. However, the roll-up and existence of a Kelvin-Helmholtz type shear layer is not visible in the current data.

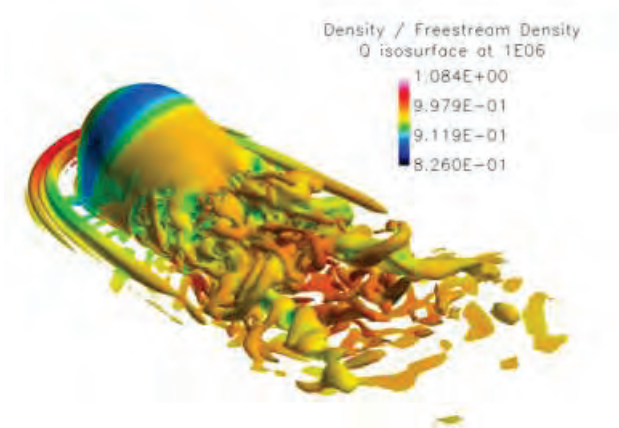


Figure 5. Iso-surface of Q-criteria colored by normalized density around the conformal-windowed turret at Mach 0.4

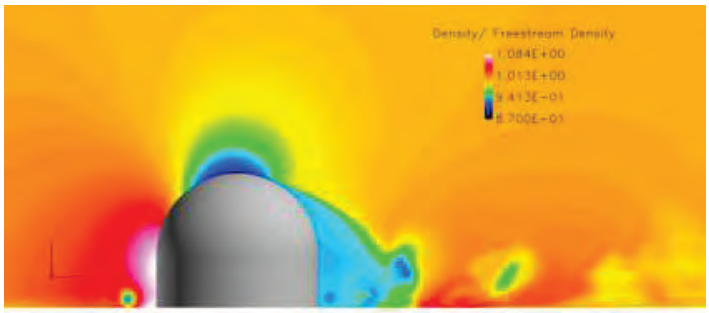


Figure 6. Instantaneous normalized density in the center plane of the turret

To investigate why the characteristic Kelvin-Helmholtz type shear layer is not forming, the pressure coefficient along the centerline for several simulations with different turbulence models and grid-resolutions was compared to experimental data. Figure 7 shows the corresponding pressure coefficient for the highest grid-resolution tested and the two different turbulence models, DDES with a SST model and DDES with a SARC model. Furthermore, the pressure coefficient for a flat-windowed turret at an azimuth angle of 120 degrees and elevation angle of 50 degrees was also calculated and is shown in Figure 7. Finally, the data is compared to experimental measurements.[12,13] For both turbulence models used, the point of separation over the top of the turret occurs just after 120 degrees. However, the experimental measurements suggest that the

flow separates at approximately 115–118 degrees. This delayed separation in the computational model is probably responsible for the lack of the Kelvin-Helmholtz shear layer in the computations. As verification that delayed separation is preventing the formation of the shear layer, the flat-windowed turret provides a discrete surface discontinuity to force the flow to separate and form a shear layer. Although the flat-windowed geometry, using DDES with a SST model, is slightly different than the conformal experimental results, Figure 7 shows the pressure coefficients match very well.

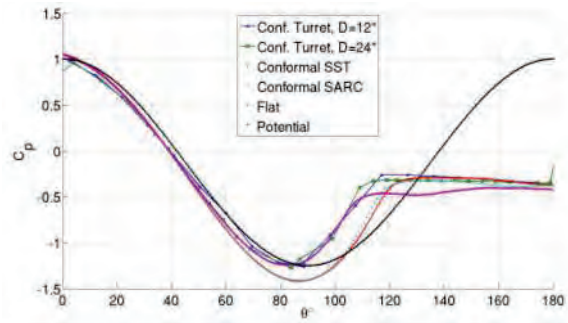


Figure 7. Comparison of the calculated pressure coefficient to experimentally-measured pressure coefficients along the centerline of a turret[12,13,14]

3.1 Conformal Window Aero-Optics

After the initial transient start-up, the resulting flow variables at each beam tap-grid location (see Figure 4) along with the surface taps on the hemisphere were written out for each time-step. To investigate the effect of the resulting flow structures and their corresponding density fluctuations on a notional beam, the density field was integrated along each beam grid to obtain the resulting OPD from Equation 3. Furthermore, three additional datasets were generated from the raw OPD data. The first removed tip/tilt from each wave-front as a function of time. This still left some amount of static optical aberration due to the steady density gradient across each aperture. Therefore, the static lensing, computed as the time-averaged OPD across each wave-front, was removed from the original OPD data. The final dataset removed tip/tilt from the OPD data after static lensing effects were removed. This dataset corresponds to experimental measurements from a Shack-Hartman wave-front sensor, similar to flight test data.[15]

Figure 8 shows the resulting time-averaged OPD_{RMS} of each of the four different OPD datasets at the five different viewing angles where beam grids were placed. For comparison to experimental measurements[15,16], the data are plotted against the viewing angle (α) of each of the beam grids,

$$\alpha = \cos^{-1}(\cos(Az)\cos(El)), \tag{4}$$

and the OPD_{RMS} values have been normalized by the appropriate scaling for subsonic flight regimes,

$$OPD_{Norm} = \frac{OPD_{RMS}}{\frac{\rho}{\rho_{SL}} M^2 D}., \tag{3}$$

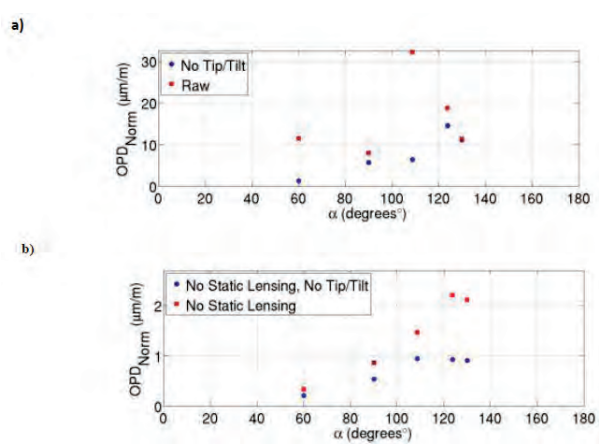


Figure 8. Calculated OPD_{Norm} for the flow around the canonical conformal-windowed turret at five different viewing angles: a) normalized OPD_{RMS} without static lensing effects removed, and b) normalized OPD_{RMS} without static lensing

Figure 8a shows that the normalized OPD_{RMS} values are very large, even when tip/tilt is removed. Furthermore, at the forward looking angle (α=60°), when tip/tilt is not removed, the results show that static lensing effects on a beam can be detrimental to airborne systems. Note that at this angle when tip/tilt is removed the normalized OPD_{RMS} values is reduced by almost an order-of-magnitude, indicating that a large static density gradient is present on the front of the turret, which can also be seen in Figure 9. Furthermore, at a viewing angle of approx. 110 degrees, a very large raw OPD_{Norm} value was obtained. When tip/tilt was removed, the OPD_{Norm} was drastically reduced; this indicates that the beam propagates through a low-density region whose thickness changes drastically over the aperture. Figure 8b, where static lensing was removed, shows that the same general trends seen in experimental flight data are captured in the computations, although at the backward-looking angles, the normalized OPD values are under-predicted, probably due to the lack of a Kevin-Helmholtz-type shear layer structures.[15,16] However, at an azimuth angle of 180 degrees, a reduction in the normalized OPD values is seen, as the beam propagates in between the two horn vortices. Furthermore, with static lensing removed, removing tip/tilt barely reduced the OPD_{RMS} at the forward-looking angles, since density fluctuations in this region originate from small boundary-layer-type structures. However, at the backward-looking angles, removing tip/tilt reduced the OPD_{RMS} values by a factor of two, indicating that the structures in the flow are on the order of the size of the beam aperture, resulting in aperture effects.[17] To see the difference in OPD for the forward- and backward-looking angles, Figure 9 shows instantaneous wave-fronts for the beams at Az=45°, El=45° and Az=120°, El=50°. As shown in Figure 9, the spatial fluctuations in the wave-front looking upstream are much smaller than the spatial fluctuations when looking downstream. The downstream fluctuations also show large, linearly coherent structures across the aperture.

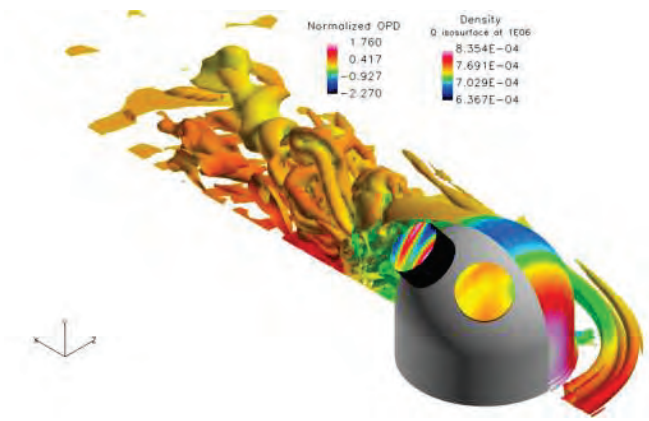


Figure 9. Comparison of the instantaneous wave-fronts (at Az=45°, El=45° and for Az=120°, El=50°) on a canonical conformal turret at Mach 0.4

3.2 Adaptive Optics Feedback Control

Given the current flow state and optical database, the development of a feedback system based on various reduced-order modeling type techniques was initiated. As a first step, the OPD data for each beam as a function of time was decomposed using both Proper Orthogonal Decomposition (POD) and a Dynamic Mode Decomposition (DMD).[18] In both cases, the OPD data is decoupled spatially and temporally. For POD, this decoupling of space and time is optimum in the sense that a truncation in the number of spatial modes to reproduce the original data always minimizes the error in the reconstruction. For DMD, space and time are separated into bases of fixed-frequency, such that the first mode corresponds to the lowest resolved frequency based on the time history of the computation, and the last spatial basis corresponds to the highest resolved frequency in the flow-field. In this case; however, the bases are not ordered by energy, but rather frequency, and more spatial modes must be used to reduce the reconstruction error to values seen in POD.

Once the OPD data was decomposed into spatial and temporal modes (either from POD or DMD), the corresponding time coefficients provided an estimate of what the instantaneous OPD was without directly measuring it. Therefore, equally spaced pressure sensors (using surface tap data in the simulations) were placed around the beam to correlate the OPD POD time coefficients to the measured pressure signals. With the mapping between the pressure and POD time coefficients, predicted OPD across the beam could, for example, be used for feedback flow control or to actuate a deformable mirror (DM) to correct for the optical aberrations resulting from the flow around the turret.

3.3 Proper Orthogonal Decomposition

An example of the POD spatial modes is shown in Figure 10 for all four forms of OPD calculated in the beam reference frame. The first predominant mode for the raw OPD data resembles optical tip/tilt. When compared to Figure 8a, a large reduction in the OPD_{Norm} was seen when tip/tilt was removed, which is verified by the POD in that the mode with the most energy is predominantly tip/tilt. The same effect can be seen in the OPD data with static lensing removed, where Mode 2 resembles optical tip/tilt.

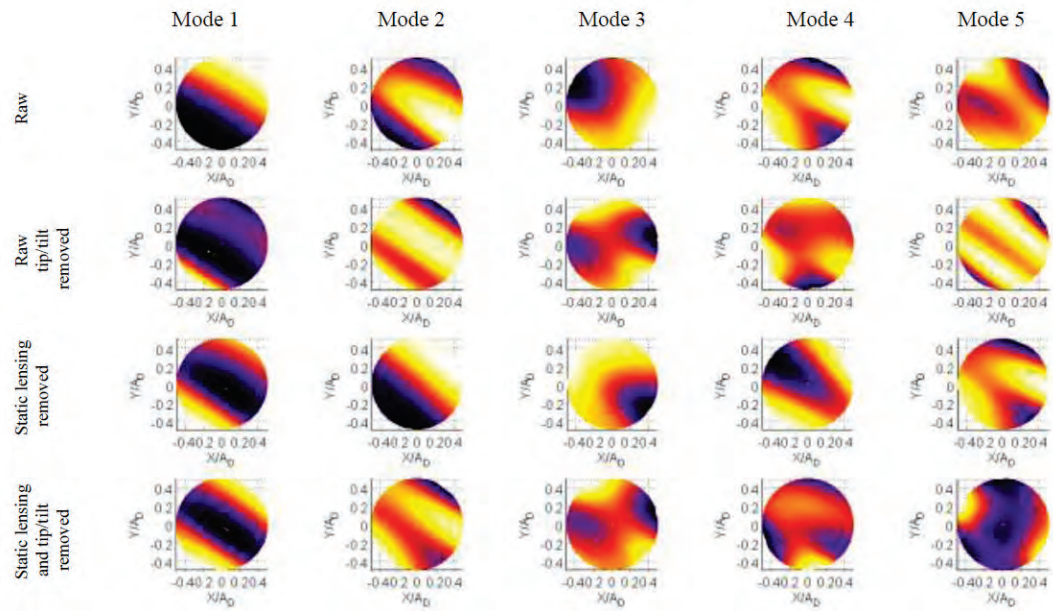


Figure 10. Calculated POD spatial modes for an azimuth angle of 120 degrees and an elevation angle of 50 degrees

With the spatial modes and corresponding time coefficients for each beam viewing angle, a linear mapping between the pressure signals around the beam was determined, where the number of pressure sensors varied from 1 to 30 to determine the optimum number of sensors to use. For the current study, 2,000 time-steps were used to find the mapping between the pressure and POD time coefficients. That mapping was then used to estimate the POD time coefficients over the next 1,000 time-steps to compare the estimated results to the simulation data and determine the error in the estimated OPD based on the mapping and number of sensors. For POD, using the first 10 Modes captured over 98% of the energy (Figure 11).

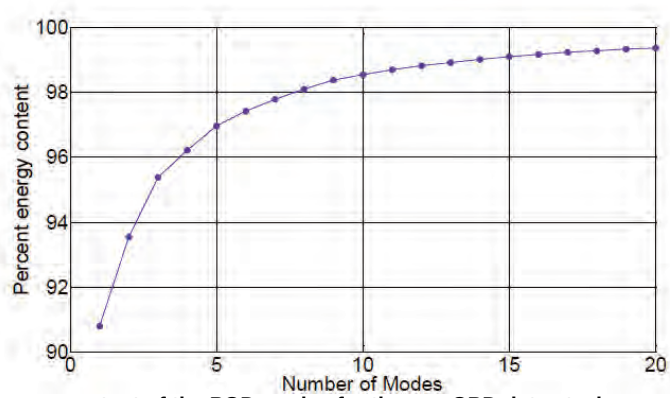


Figure 11. Cumulative energy content of the POD modes for the raw OPD data at a beam angle of Az=120°, El=50°

Figure 12 shows the actual Mode 1 time coefficient for the raw OPD data for the Az=120°, El=50° beam. For the data shown, 10 pressure taps equally spaced around the perimeter of the beam were used to find a linear mapping between the pressure signals and the POD time coefficients. As shown, over the final 50 ms, using the pressure signals on the surface of the turret provides a good estimate of the actual POD time coefficient. Similar results were obtained for the other 10 modes

and for the other beam viewing angles (not shown). Finally, with 10 pressure sensors used to estimate the first 10 POD mode time coefficients, the maximum absolute error of the actual OPD and the estimated OPD remained below 0.1%.

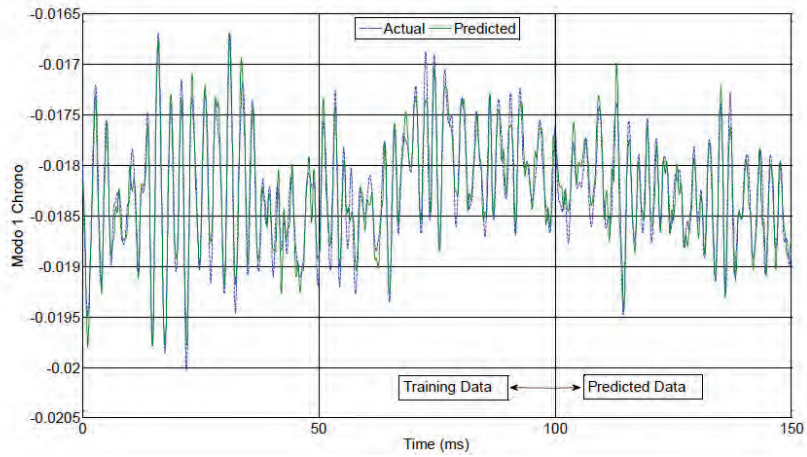


Figure 12. Comparison of the actual Mode 1 time coefficient for the raw OPD data, and the estimated Mode 1 time coefficient based on 10 pressure sensors equally spaced around the outside of the beam at Az=120°, El=50°

3.4 DMD Analysis

The Dynamic Mode Decomposition analysis provides similar results to POD, with the exception that more modes are required to reproduce the same error in the estimated OPD.[18] However, DMD yields valuable information about the data, primarily the frequency content over the whole aperture. Figure 13 shows the frequency content of the OPD for each of the five beam orientations, calculated over a time period of 150 ms. In all the data, the previously noted "aero-optic hump" can clearly be seen, starting around a frequency of 100 Hz and extending to a frequency around 1 kHz.[19] Furthermore, as the azimuth angle is increased, the energy content also increases, although not as drastically as expected. This might also be attributed to the weaker formation of a Kevin-Helmholtz-type shear layer at the larger angles when compared to experimental data.

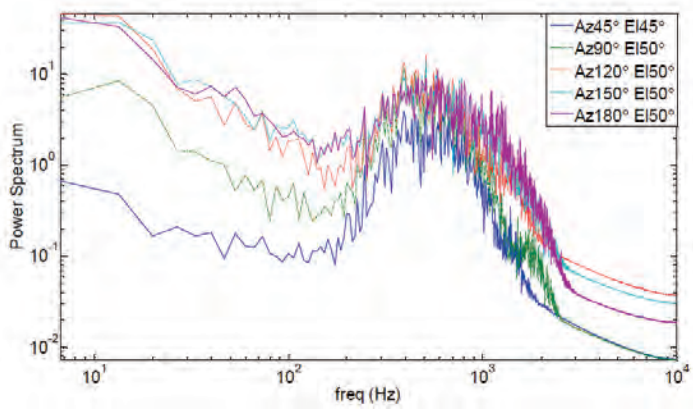


Figure 13. Dynamic Mode Decomposition power spectrum of the OPD without static lensing or tip/tilt for the five different beam viewing angles

4. Conclusion

In this paper, the flow-field of a three-dimensional turret is analyzed for its aero-optical implications. Two turret geometries, one with a conformal window and one with a flat window, were considered. Using High Performance Computing Modernization Program (HPCMP) resources under the Challenge Project C4L, Delayed Detached-Eddy Simulations using Cobalt were performed to study the flow-field. The simulation results show significant optical distortions for all viewing angles under consideration.

Comparing the simulation results with experimental findings shows that the separation of the boundary-layer over the top of the turret occurs later in the simulations than in the experiments. Nevertheless, a detailed optical analysis of the simulation data using Proper Orthogonal Decomposition, as well as Dynamic Mode Decomposition, allowed for the development of an accurate flow model. Using as few as ten pressure sensors equally spaced around the optical aperture, the OPD of the flow-field could be predicted with an error of less than 0.1%. Therefore, such a configuration is a viable candidate for implementing feedback flow control or for feedback control of the emitted beam using a deformable mirror.

Acknowledgements

The DoD HPCMP supported this project by supplying supercomputer time under Challenge Project C4L at the DoD Distributed Resource Center at Maui and Alaska. We would like to thank Dr. Charles Wisniewski, Director, Modeling and Simulation Research Center, US Air Force Academy, for providing additional computing, storage and visualization resources.

References

1. McMackin, L., Hugo, R.J., Pierson, R.E., and Truman, C.R., "High speed optical tomography system for imaging dynamic transparent media." *Optics Express*, 1(11), pp. 302–311, 1997.

2. Jumper, E.J. and Fitzgerald, E.J., "Recent advances in aero-optics." *Prog. Aero. Sci.*, 37, pp. 299–339, 2001.

3. Settles, *Schlieren & Shadowgraph Techniques*, Springer, 2001.

4. Siegel, S., Seidel, J., Fagley, C., Luchtenburg, D., Cohen, K., and McLaughlin, T., "Low-dimensional modeling of a transient cylinder wake using double proper orthogonal decomposition." *J. Fluid Mech.*, 610, pp. 1–42, 2008.

5. Fagley, C., Balas, M., Siegel, S., Seidel, J., and McLaughlin, T., "Reduced Order Model of Cylinder Wake with Direct Adaptive Feedback Control." *AIAA Paper 2009-5856*, 2009.

6. Seidel, J., Siegel, S., McLaughlin, T., and Fagley, C., "Feedback Flow Control of a Shear Layer for Aero-Optic Applications." *AIAA Paper 2010-0356*, 2010.

7. Strang, W., Tomaro, R., and Grismer, M., "The defining methods of Cobalt60: A parallel, implicit, unstructured Euler/Navier-Stokes flow solver." *AIAA Paper 1999-0786*, 1999.

8. Gottlieb, J.J. and Groth, C.P.T., "Assessment of Riemann solvers for unsteady one-dimensional inviscid flows for perfect gases." *J. Comp. Phys.*, 78(2), pp. 437–458, 1988.

9. Seidel, J., Fagley, C., and Decker, R., "Mitigation of Optical Distortions in a Free Shear Layer using Feedback Flow Control." *DoD HPC User Group Conference*, 2010

10. Marcum, D. and Weatherill, N., "Unstructured grid generation using iterative point insertion and local reconnection." *AIAA J.*, 33(9), pp. 1619–1625, 1995.

11. Jelic, R., Sherer, S., and Greendyke, R., "Simulation of Various Turret Configurations at subsonic and Transonic flight Conditions Using OVERFLOW." *AIAA-2012-464*.

12. Gordeyev, S. and E. Jumper, "Fluid Dynamics and Aero-Optics of Turrets." *Progress in Aerospace Sciences*, 46, pp. 388–400, 2010.

13. Gordeyev, S., M. Post, T. MacLaughlin, J. Ceniceros, and E. Jumper, "Aero-Optical Environment Around a Conformal-Window Turret." *AIAA Journal*, Vol. 45, No. 7, pp. 1514–1524, 2007.

14. Gordeyev, S., T. Hayden, and E. Jumper, "Aero-Optical and Flow Measurements Over a Flat-Windowed Turret." *AIAA Journal*, Vol. 45, No. 2, pp. 347–357, 2007.

15. Porter, C., S. Gordeyev, M. Zenk, and E. Jumper, "Flight Measurements of Aero-Optical Distortions from a Flat-Windowed Turret on the Airborne Aero-Optics Laboratory (AAOL)." *AIAA Paper 2011-3280*.

16. De Lucca, N., S. Gordeyev, and E. Jumper, "The Airborne Aero-Optics Laboratory, Recent Data." *Acquisition, Tracking, Pointing, and Laser Systems Technologies XXVI, Proceedings of SPIE*, Volume 8395, Paper 8395-7, June 2012.

17. Siegenthaler, J., Gordeyev, S., and Jumper, E., "Shear Layers and Aperture effects for Aero-Optics." *AIAA-2005-4772*, 36th AIAA Plasmadynamics and Lasers Conference, Toronto, Ontario, June 6–9, 2005.

18. Schmid, P.J., "Dynamic mode decomposition of numerical and experimental data." *J. Fluid Mech.*, Vol. 656, pp. 5–28, 2010

19. De Lucca, N., S. Gordeyev, and E. Jumper, "The Study of Aero-Optical and Mechanical Jitter for Flat Window Turrets." *AIAA Paper 2012-0623*.

Numerical Investigation of Boundary-Layer Separation at Different Reynolds Numbers

A. Gross, S. Hosseinverdi, and H.F. Fasel
Department of Aerospace and Mechanical Engineering, University of Arizona, Tucson, AZ
{agross, shirzadh, faselh}@email.arizona.edu

Abstract

Earlier research has shown that flow control that exploits shear-layer instability is very effective for controlling laminar separation at low-Reynolds numbers. If such active flow control strategies work for free-flight conditions that are characterized by free-stream turbulence and/or larger-Reynolds numbers remains an open question. Using highly-resolved direct numerical simulations, we are investigating the effect of free-stream turbulence on laminar separation bubbles. The simulations demonstrate that free-stream turbulence has a strong effect on the bubble length and height. A detailed analysis did; however, reveal that the shear-layer instability remained the dominant mechanism. Simulations were also carried out for an entire NACA 64(3)-618 airfoil section at a chord-Reynolds number of 1 Million. For this Reynolds number, direct numerical simulations where all scales of fluid motion are resolved are very expensive, and a one-equation hybrid turbulence model was employed. Three different angles-of-attack with turbulent trailing-edge separation, α=10, 20, and 30deg, were considered. For α=10 separation control by harmonic blowing through a span-wise slot was employed. For the chosen flow control parameters, the actuation had little effect on lift and drag.

1. Introduction

For many Air Force applications, successful active control of boundary-layer separation for lifting surfaces could lead to significant performance gains. This is especially true for Unmanned Aerial Vehicles (UAVs), which have become increasingly important for military operations. Due to small wing dimensions and low air-speeds, UAVs often operate within a Reynolds-number flight regime, for which a strong interaction exists between separation and transition. Laminar separation is also a concern for laminar airfoils which offer greatly improved aerodynamic performance compared to older more conventional airfoils. Active Flow Control (AFC) of separation is also of interest for airfoils operating at or beyond stall. For example, active control of turbulent trailing-edge separation may increase the maximum angle-of-attack or allow for post-stall maneuvering.

A well-founded understanding of the relevant physical mechanisms governing separation and transition for controlled "clean" conditions (numerical simulations and laboratory experiments) and during free-flight is required for an effective control. Of particular relevance is the consideration of the "real-life" operating environment, which is characterized by roughness and Free-Stream Turbulence (FST) among others. Experiments have shown that for sufficiently high FST levels, transition to turbulence is accelerated. Clear air turbulence, which is characterized by large length-scales, can reach high-turbulence kinetic energy levels and thus has to be considered when investigating separation and separation control.

The mean aerodynamic chord-Reynolds number for general aviation aircraft at cruise is in the order of several million. At such Reynolds numbers, separation is typically turbulent. Separation control strategies that are successful for controlling laminar separation at low-Reynolds numbers (e.g., pulsed-vortex generator jets) may also be feasible for controlling separation at flight-Reynolds numbers. The question how the relevant flow physics scale with Reynolds number can be addressed with numerical simulations. For example, the Reynolds number scalability can be investigated by choosing appropriate dimensionless parameters (e.g., dimensionless frequency and momentum coefficient), keeping them constant, and then determining if the dimensionless performance parameters (e.g., lift and drag coefficient) remain constant when the Reynolds number is altered. Analogously, similarity of the flow physics would mean that similar physical mechanisms are exploited (the corresponding amplitudes may scale with the Reynolds number, etc.).

Several years ago, a research project was initiated at the University of Arizona involving dynamically-scaled models of general aviation aircraft, such as the Aeromot 200S Super Ximango motor glider, which has a modified NACA 64_3-618 airfoil (the modified airfoil has a higher maximum-lift coefficient than the original NACA 64_3-618) [Gross, et al., 2008]. Two-dimensional (2D) wing sections were tested in wind and water tunnel experiments for Reynolds numbers between 64,200 and Re=322,000 [Mack, et al., 2008; Heine, et al., 2008; Plogmann, et al., 2009]. The experiments were supported by simulations for Re=64,200 [Brehm, et al., 2006; Gross and Fasel, 2010; Balzer and Fasel, 2010a], and Re=322,000 [Gross and Fasel, 2011]. The full-size cruise-Reynolds number is 3.2 Million. Thake, et al. [2011] investigated separation control by steady vortex generator jets for the original NACA 64_3-618 airfoil at four different Reynolds numbers, Re=64,200, 180,000, 1 Million, and 4 Million. When the blowing ratio was kept constant, the flow control effectiveness varied with Reynolds number.

We are employing computational fluid dynamics (CFD) for investigating separation and separation control. The underlying flow physics are highly-complex, as both unsteady separation and transition are at work simultaneously. In particular, we are employing high-resolution Direct Numerical Simulations (DNS) for investigating canonical separation bubbles on a flat-plate. For these simulations, we accurately model FST as would be seen, for example, in a wind tunnel [Balzer and Fasel, 2010b]. Free-stream turbulence is often neglected in CFD but is always a factor in real-world applications. The considerable computational expense of fully-resolved DNS can be avoided when all or parts of the turbulence kinetic energy spectrum are modeled. For example, in Reynolds-averaged Navier-Stokes (RANS) all unsteady motion is modeled, while in unsteady RANS (URANS) all but the dominant large scale structures are modeled. In Large Eddy Simulations (LES), the filter cutoff is located in the inertial sub-range and only the statistically more random turbulence is modeled. Hybrid models blend between RANS and LES according to the local grid-resolution and flow properties. For example, near walls they revert to pure RANS models. This property is advantageous, as modern RANS models are well-suited for the computation of attached boundary-layer flows. Away from walls, a typical LES or DNS is recovered.

In this paper, we first review the computational methods. Next, results from direct numerical simulations that we carried out to investigate the effect of free-stream turbulence on laminar separation are discussed. Then, we show results from high-Reynolds number simulations for a NACA 64_3-618 airfoil section. Finally, we provide a summary and conclusion.

2. Computational Method

We employ a two-pronged computational approach. An incompressible finite-difference code is employed for the simulations where we investigate the effects of free-stream turbulence for a separation bubble on a flat-plate. A compressible finite-volume code is employed for the simulations of an entire airfoil section.

The higher-order-accurate <u>incompressible finite-difference code</u> was developed in our laboratory [Meitz and Fasel, 2000]. Derivatives in the stream-wise and wall-normal directions are discretized with fourth-order-accurate compact differences. A pseudo-spectral method is employed in the span-wise direction and a fourth-order-accurate Runge-Kutta scheme is employed for time-integration. A hybrid parallelization approach, using both Message Passing Interface (MPI) and shared memory parallel programming (OpenMP) was adopted to make optimal use of the available computational resources..

For simulations of the entire airfoil geometry, we are employing a higher-order-accurate <u>compressible finite-volume code</u> that was also developed in our laboratory [Gross and Fasel, 2008]. The Navier-Stokes equations are discretized with a ninth-order-accurate upwind scheme for the convective terms and a fourth-order-accurate discretization for the viscous terms. The turbulence model transport equations are discretized with a second-order-accurate upwind scheme for the convective terms and a second-order-accurate discretization for the viscous terms. The code was parallelized using MPI.

3. Investigation of Effect of Free-Stream Turbulence

The setup of the simulations was such that a comparison with water tunnel experiments at the Hydrodynamics Laboratory of the University of Arizona was possible [Radi and Fasel, 2010; Jagadeesh and Fasel, 2012]. In the experiments, a laminar separation bubble is generated on a flat-plate through the close proximity of a modified NACA 64_3-618 airfoil (Figure 1). The airfoil has 30cm chord and is placed at a distance of 12cm from the flat-plate. Suction is applied in the aft part of the airfoil to prevent flow-separation and velocity data is obtained using Particle Image Velocimetry (PIV). For our DNS, a wall-normal velocity distribution is specified at the free-stream boundary such that the measured velocity data at a distance of 11cm from the flat-plate is matched. Length-scales were made dimensionless with

a reference length of 0.1m, and velocities were made dimensionless with a reference velocity of 0.028m/s. The computational grid is shown in Figure 1. The number of grid-points in the stream-wise and wall-normal directions were 2,001×210. The span-wise domain extent was L_z=2 ($16\delta_{99,0}$) and was resolved using 140 collocation points (84 Fourier modes). The Reynolds number based on displacement thickness and stream-wise velocity at the inflow boundary, $Re_{\delta^*,in}$, was 140.

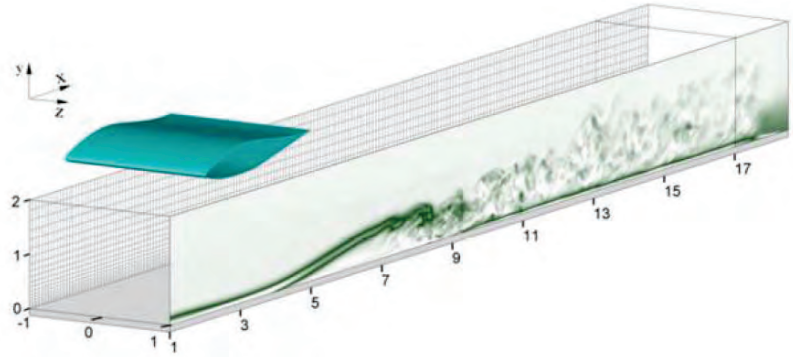


Figure 1. Computational grid for DNS (every 10th grid line is shown). Also shown are instantaneous gray-scale contours of the span-wise-averaged span-wise vorticity, ω_z.

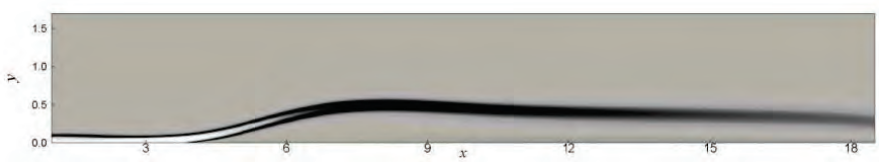


Figure 2. Instantaneous gray-scale contours of the span-wise vorticity component, ω_z, obtained from 2D DNS

Instantaneous contours of the span-wise vorticity component, ω_z, obtained from a 2D simulation are presented in Figure 2. Without free-stream turbulence, the flow remains steady. The separation bubble is much longer than in the experiments. Of course, laminar-turbulent transition is suppressed in the 2D simulation. However, subsequent three-dimensional (3D) simulations without free-stream turbulence also predicted a steady bubble that was far too long. The fact that the flow does not transition in the 3D DNS is an indication of the very low background noise level of our research DNS code. In the experiment, transition occurs via the amplification of naturally-occurring background disturbances such as, for example, tunnel free-stream turbulence. For very strong adverse pressure gradients, and/or when the flow is absolutely/globally unstable, the "numerical noise" originating from round-off and/or truncation errors in numerical simulations is sufficient to reproduce "realistic" separation bubbles. However, this is not the case here.

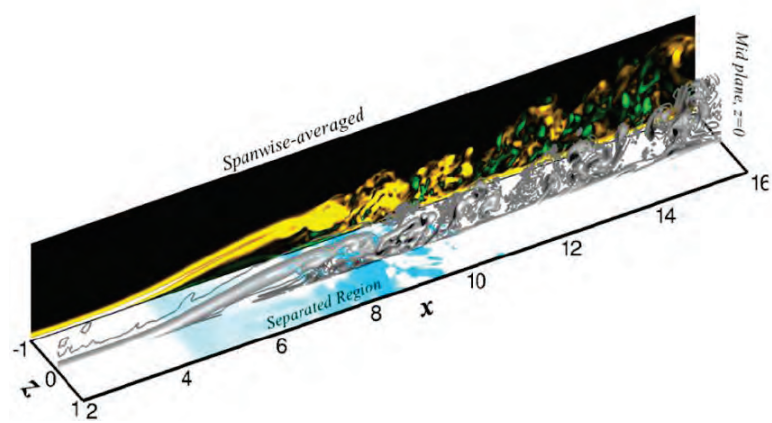


Figure 3. Visualization of instantaneous contours of the span-wise vorticity component, ω_z obtained from 3D DNS. Shown are contours $\omega_z \leq 0$ at the wall (blue shades represent reverse flow), span-wise-averaged (color) and contours at mid plane, z=0 (gray-scale and solid lines).

The picture changes completely when controlled isotropic free-stream disturbances with a free-stream turbulence intensity of 0.1% (same as in experiment) and an integral length-scale of 0.065 are introduced at the inflow boundary. Instantaneous contours of the span-wise vorticity, ω_z, are presented in Figure 3. The boundary- layer separates at x_S=4.27. Near x=7, the separated boundary-layer "rolls-up" and the bubble is "shedding" span-wise coherent structures (often referred to as "rollers"). Near x=7.5, intense small-scale structures with pronounced three-dimensionality appear in the separated region indicating transition onset. Shortly thereafter the flow reattaches to the wall. Instantaneous side views of λ_2=−2 [Jeong and Hussain, 1995] taken over the course of one shedding cycle are shown in Figure 4. The small-scale turbulence is organized in lumps (coherent structures) which grow in size as they travel downstream. The dashed line in Figure 4 illustrates that the structures travel at an almost-constant speed (Figure 4).

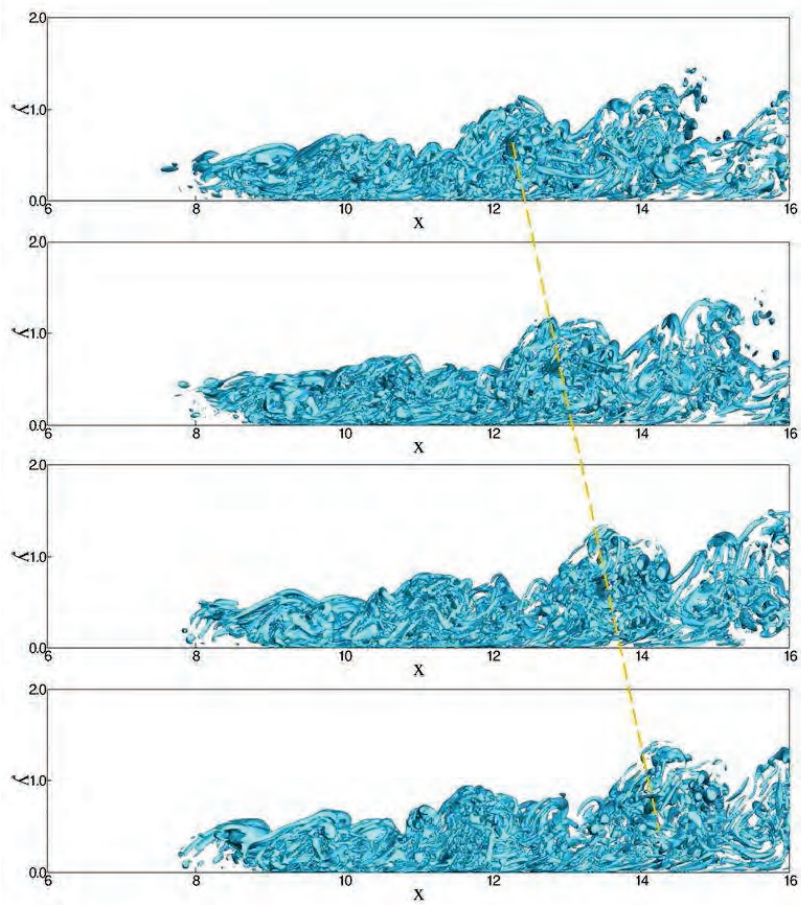


Figure 4. Side views of iso-surfaces of λ_2=−2 for four time instances, t/T_{sh}=0.1, 0.4, 0.7, 1.0 (shedding period, T_{sh}). The dashed line illustrates the time evolution of one coherent flow structure.

A Proper Orthogonal Decomposition (POD) [Sirovich, 1987] was carried out using 120 snapshots of the velocity field equidistantly distributed over 12 periods of the periodic vortex shedding. Perspective views of the v-velocity component for POD modes *i*=1 & 3 and top-down views for modes *i*=1…4 are presented in Figure 5. Modes 1 and 2 are similar except for a phase shift of 1/4 of the stream-wise wavelength indicating that this mode pair captures a downstream travelling structure/wave. The POD results reveal that the most energetic flow structures are predominantly 2D. Modes 3 and 4 are stream-wise-oriented "arrow-head"-shaped structures with large-span-wise wavelength. Figure 5(a) also indicates that the stream-wise wavelength of the energetic 2D structures increases in downstream direction.

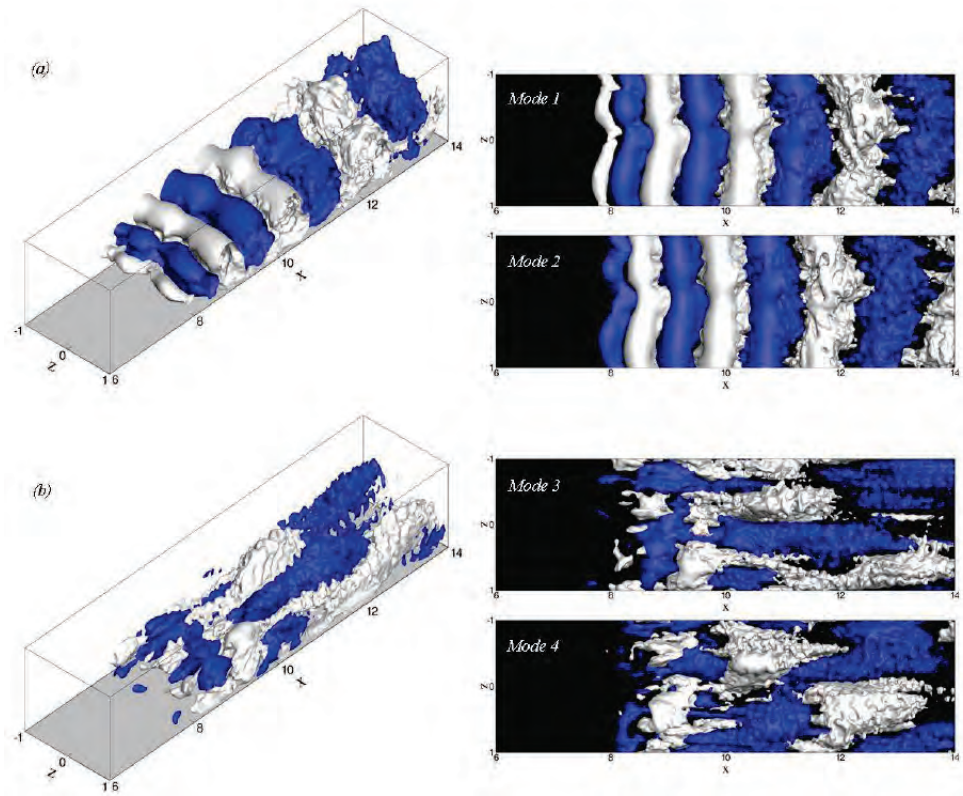


Figure 5. POD eigenfunctions for a) mode i=1 (perspective view, left) and modes i=1, 2 (top-down view, right) and b) mode i=3 (perspective view, left) and modes i=3, 4 (top-down view, right). Shown are positive (blue) and negative (white) iso-surfaces of the wall-normal velocity component: |v|=0.02.

The time coefficients, A_i, associated with POD modes i=1, 2 are plotted in Figure 6 (left). The frequency of the time-coefficients matches the natural shedding frequency of the separation bubble. The POD eigenvalue spectrum is shown in Figure 6 (right). The POD-mode eigenvalue magnitude corresponds to twice the kinetic energy content of the respective POD modes. Most of the kinetic energy is contained in the mean-flow (mode 0). Eigenvalues 1 & 2 have almost identical magnitude which is typical for traveling waves.

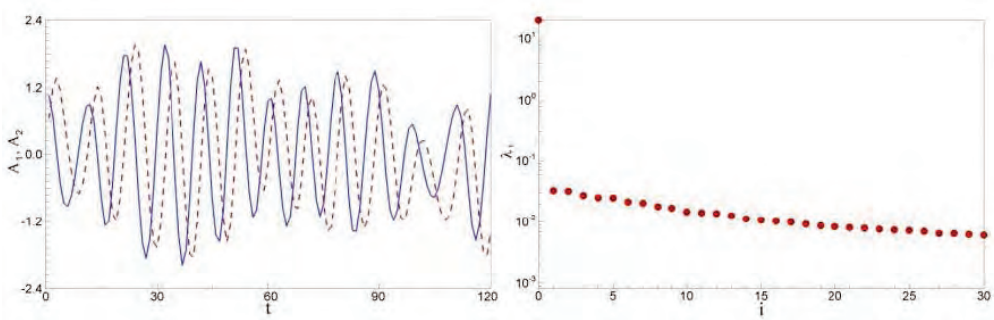


Figure 6. Left: POD time-coefficients, $A_i(t)$, associated with modes i=1 (---) and i=2(—); Right: POD eigenvalues

To detect the dominant frequencies in the separation bubble, the time-signal of the wall-normal disturbance velocity component $v' = v - \overline{v}$ at the center of the domain, z=0, and at a distance y=δ^*(x) away from the wall (δ^* is the local, time- and span-wise-averaged displacement thickness) was analyzed. Figure 7 (left) shows gray-scale contours of **ln (|v′|)** in an x-t-diagram. For x<6 downstream growth of v′ is observed with very little variation in time. Such low-frequency oscillations are a common characteristic of realistic separation bubbles. For x>6 a strongly time-periodic variation of the velocity signal can be observed. These oscillations can be associated with the fundamental shedding frequency of the separation bubble. The traces show a positive slope in the x-t-diagram, indicating that the disturbances are travelling in the downstream direction. Time signals of v′ at three different locations are shown in Figure 7 (right).

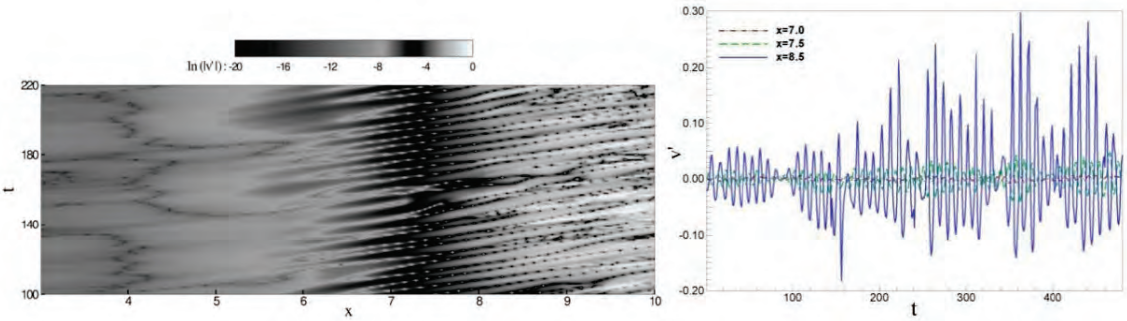


Figure 7. Left: Contours of wall-normal disturbance velocity, v′, in t/x diagram; Right: Time-signals of v′ at three different locations

Both discrete Fourier transforms and the maximum entropy method [Ghil, et al., 2002] were employed for obtaining velocity spectra for four different downstream locations (Figure 8). All spectra indicate distinct peaks close to f^*=0.1 Hz. This frequency corresponds to the shedding or fundamental frequency of the separation bubble.

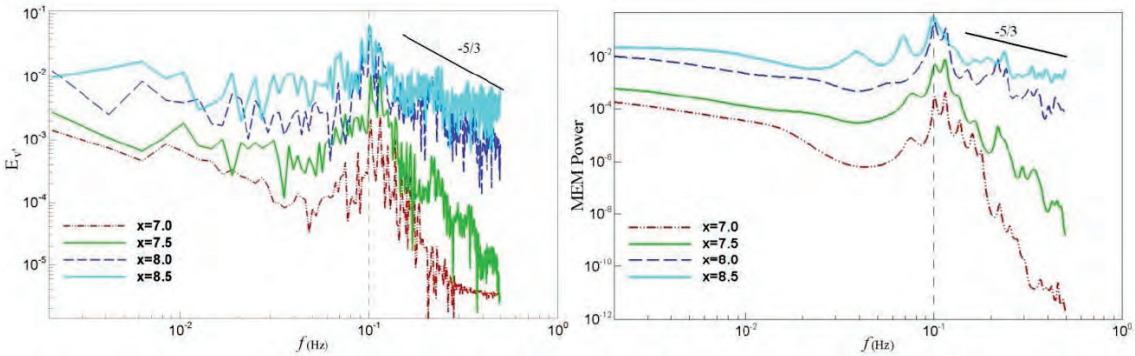


Figure 8. Spectra of wall-normal disturbance velocity, v′, at y=δ*(x) & z=0 obtained using discrete Fourier transforms (left) and maximum entropy method of order M=40 (right)

To obtain insight into the nature of the underlying instability mechanisms, the downstream development of the u′-disturbance velocity for the fundamental frequency, f^*, was compared with results from Linear Stability Theory (LST) calculations (Figure 9). Downstream of separation, x_S=4.27, the disturbances grow exponentially (linear in the log-plot) until they saturate at a finite-amplitude level. The agreement with the LST result is very good indicating that the primary instability is linear. Wall-normal profiles of the u′-disturbance velocity taken inside the region of exponential growth are shown in Figure 8 (right). The amplitude distribution from the DNS agrees remarkably well with the LST result. This good agreement with the LST results indicates that the onset of unsteadiness and transition is due to inviscid (inflectional) shear-layer instability.

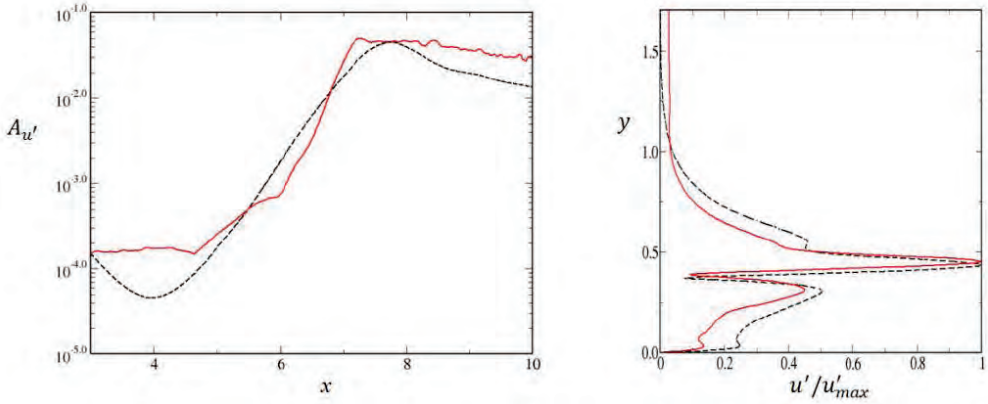


Figure 9. Left: Downstream development of u′-disturbance amplitude (max. over y) for the fundamental frequency f=0.1 Hz. Right: Wall-normal profiles of u′ disturbance (normalized by its maximum) taken at the downstream location where $A_{u'}=10^{-2}$. (---) LST; (—) DNS.

4. Simulations of Entire Airfoil

The Reynolds number based on mean-aerodynamic chord for full-size general aviation aircraft is of order 1 Million or more. To investigate the Reynolds number effect on the effectiveness of flow control devices, we are gradually increasing the Reynolds number of our simulations. Earlier simulations demonstrated the effectiveness of separation control using time-periodic blowing and suction. For example, low-Reynolds number performance losses were successfully eliminated for Re=64,200 [Gross and Fasel, 2010], and stall could be delayed for Re=320,000 [Gross and Fasel, 2011]. Direct Numerical Simulations (DNS) which do not employ any modeling become prohibitively expensive as the flight Reynolds numbers of general aviation aircraft are approached. The required number of grid-points for direct numerical simulations scales with $Re^{9/4}$. Therefore, simulations at Re=320,000 require roughly $(320/64.2)^{9/4}$=37 times more grid-points than simulations at Re=64,200. In addition, when keeping the dimensionless time-step (Courant-Friedrichs-Lewis number) constant, the number of time-steps increases by $(37)^{1/3}$=3.3. An earlier DNS for Re=320,000 [Gross and Fasel, 2011] demonstrated that simulations at higher-Reynolds numbers such as Re=1 Million would not be feasible.

The present simulations are for a modified NACA 64_3-618 airfoil. This is the airfoil of the Aeromot 200S SuperXimango motor glider which is subject of a comprehensive flight research program at the University of Arizona [Gross, et al., 2008]. Experiments by Heine, et al. [2008] and Plogmann, et al. [2009] for Re=320,000 (the scaled cruise-Reynolds number for a 1:5 scale model) showed stall onset for α>8deg. For α=13.2deg a laminar separation bubble was observed near the leading-edge and the flow separated turbulence near the trailing-edge. The cruise-Reynolds number of the full-size is about 3 Million. Because wind tunnel measurements are available for the original NACA 64_3-618 airfoil for Re=1 Million [Thake, et al., 2011] the decision was made to choose the same Reynolds number for the present simulations. Hybrid simulations based on a one-equation hybrid model by De Langhe, et al. [2008] were carried out for angles-of-attack of α=10, 20, and 30 deg. A stochastic model was added to "seed" small-scale turbulent velocity fluctuations for flow from coarse- into fine-grid regions [Gross and Fasel, 2011].

A Poisson grid-generator was employed to generate an O-grid with 600×200 cells. This 2D grid was then extruded in the span-wise direction (Figure 10). The span-wise grid extent and the number of cells in the span-wise direction were 0.15c (where c is the chord-length) and 32 for α=10deg, and was doubled and quadrupled for α=20deg and α=30deg. The wall-normal grid extent was 20c. A characteristics-based non-reflecting boundary condition was applied at the free-stream boundary and periodicity was assumed in the span-wise direction. In addition to simulations of the uncontrolled flow, simulations were carried out with Active Flow Control (AFC) using wall-normal harmonic blowing through a span-wise slot. The slot was located at 61% chord (slightly upstream of the separation location of the uncontrolled flow) and had a downstream extent of 1.5% chord-lengths. The maximum blowing amplitude was 10% of the free-stream velocity. The slot was modeled by a wall boundary condition for the wall-normal velocity.

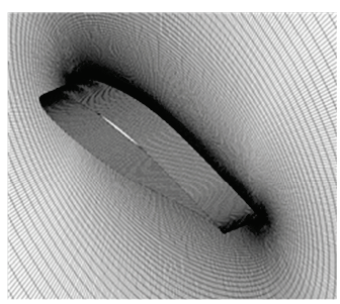


Figure 10. Computational grid for α=10 deg

Instantaneous flow visualizations obtained from the hybrid simulations are provided in Figure 11. Iso-surfaces of the Q vortex identification criterion by Hunt [1988] indicate areas where rotation dominates strain. Also shown are iso-contours of the unresolved-eddy viscosity, μ_{Tu}. Starting shortly downstream of the leading-edge the suction- side boundary-layer is turbulent. This behavior was verified by XFoil calculations, which showed a small laminar separation bubble near the leading-edge [Drela and Giles, 1987]. As the angle-of-attack, α, is increased, the trailing- edge separation becomes more and more severe and the separation line moves upstream. In all instances, following the initial shear-layer "roll-up", energy is quickly transferred into smaller-scale motion. The small-scale structures can be interpreted as coherent structures within a turbulent flow. The smallest dissipating scales are modeled in the hybrid RANS/LES approach.

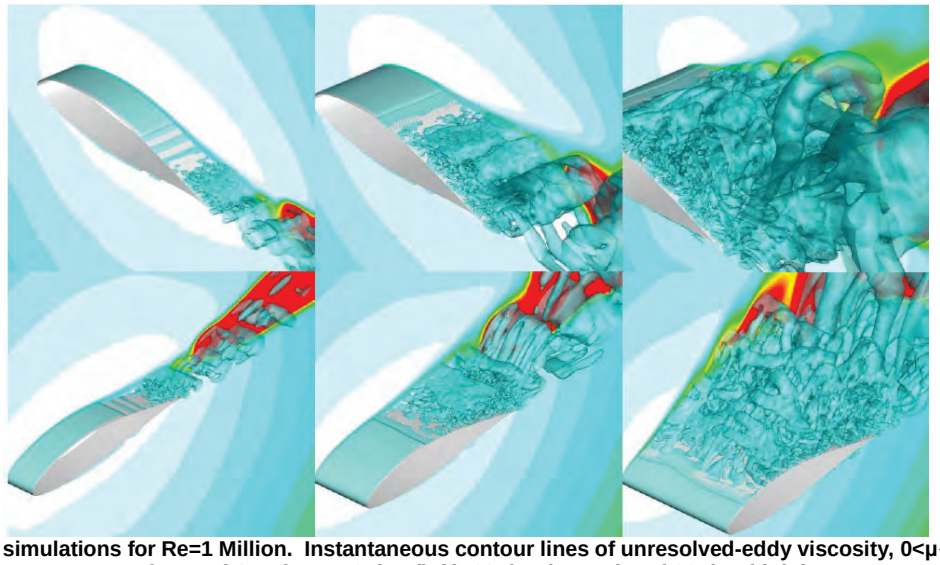


Figure 11. Airfoil simulations for Re=1 Million. Instantaneous contour lines of unresolved-eddy viscosity, $0<\mu_{Tu}<200$, and iso-surfaces of Q=1 for α=10 deg (left), 20 deg (center) and 30 deg (right).

Top-down views are provided in Figure 12. In all instances the flow structures that are evolving in the separated boundary-layer display a distinct stream-wise wavelength. In addition, especially for α=30deg a considerable modulation of the dominant flow structures in the span-wise direction becomes apparent. Our earlier research revealed that low-Reynolds number flow control becomes highly-efficient when hydrodynamic instabilities of the flow are exploited. In particular when span-wise disturbances were introduced with a frequency that excited the inviscid shear-layer instability, span-wise vortices were generated even for very low-blowing ratios.

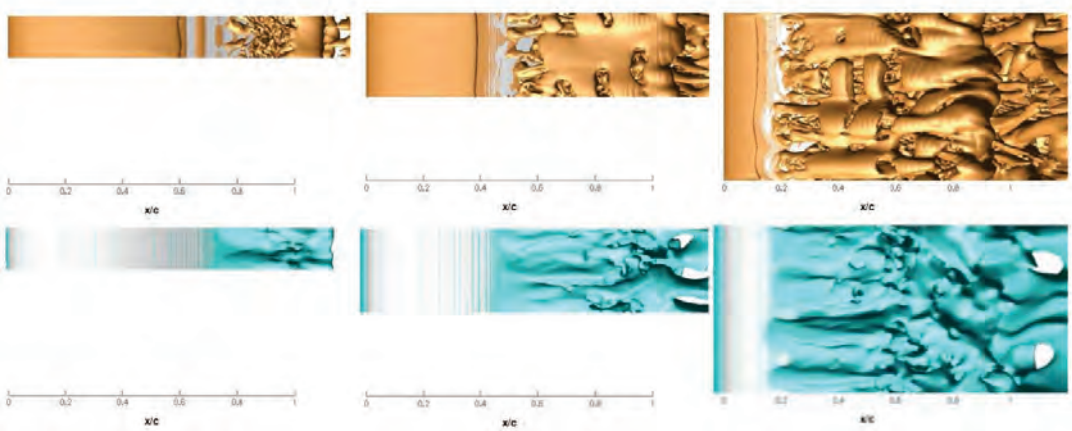


Figure 12. Airfoil simulations for Re=1 Million. Instantaneous iso-surfaces of Q=1 (top) and u=0.05 (bottom) for α=10 deg (left), 20 deg (center) and 30 deg (right).

Figure 13 provides iso-contours of the stream function computed from temporal and span-wise averages. The streamlines indicate reattachment (in the mean) upstream of the leading-edge for α=10 and 20 deg and a bluff body-wake for α=30 deg. Also, the leading-edge stagnation point is seen to move downstream as the angle-of-attack is increased.

Figure 13. Airfoil simulations for Re=1 Million. Iso-contours of stream function for α=10 deg (purple lines), 20 deg (blue lines), and 30 deg (orange lines).

A comparison of the wall pressure and skin friction coefficient distributions with Xfoil predictions [Drela and Giles, 1987] is shown in Figure 14. The wall pressure distributions extracted from the hybrid simulations match the turbulent XFoil data well. The XFoil calculations with e^N transition prediction show a laminar leading-edge separation bubble that is not captured in the hybrid simulations. The one-equation hybrid turbulence model predicts turbulent flow starting from the leading-edge. For the angles-of-attack considered here, this flow scenario is fairly accurate because the laminar separation bubble is located very close to the leading- edge. For cases with a considerable laminar run, the transition-point would have to be set (i.e., based on the e^N method) or the hybrid simulation strategy would have to be based on a more elaborate turbulence model that captures transition.

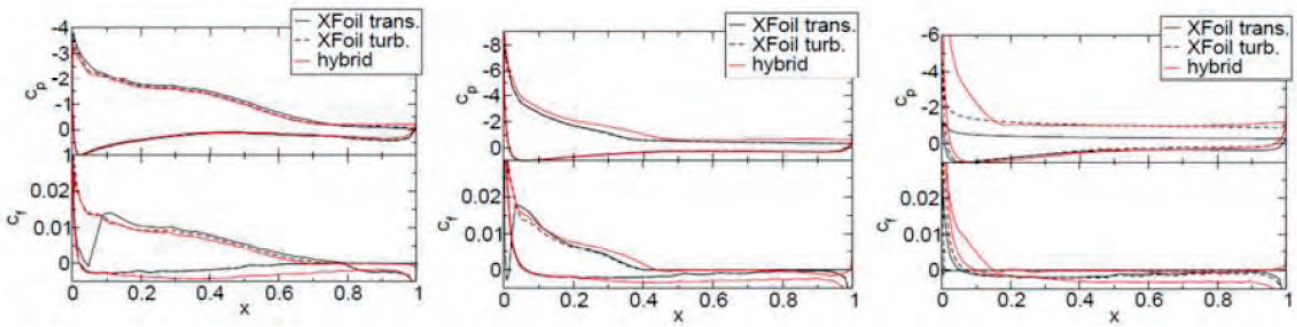


Figure 14. Airfoil simulations for Re=1 Million. Wall pressure and skin riction distributions for α=10 deg (left), 20 deg (center) and, 30 deg (right). Also shown are XFoil results obtained with e^N transition model ("trans.") and assuming fully-turbulent flow ("turb.").

Since the present results again provide indications of underlying shear-layer instability, the decision was made to investigate separation control by harmonic blowing through a span-wise slot for α=10 deg. The dimensionless blowing velocity, A, was 10% of the free-stream velocity. On purpose, a purely 2D disturbance was introduced to exploit the shear-layer instability of the separated boundary-layer. Instantaneous flow visualizations are provided in Figures 15 and 16. For a dimensionless frequency of f=1 (normalized with free-stream velocity and chord-length) the flow appears only minimally affected by the control. For f=4 span-wise vortices are forming downstream of the forcing slot.

Figure 15. Airfoil simulations for Re=1 Million and α=10 deg. Instantaneous contour lines of unresolved-eddy viscosity, $0<\mu_{Tu}<200$, and iso-surfaces of Q=1 for uncontrolled flow (left) and harmonic blowing and suction with A=0.1 and f=1 (center) and f=4 (right).

Figure 16. Airfoil simulations for Re=1 Million and α=10 deg. Instantaneous iso-surfaces of Q=1 (top) and u=0.05 (bottom) for uncontrolled flow (left) and harmonic blowing and suction with A=0.1 and f=1 (center) and f=4 (right).

Iso-contours of the stream function (Figure 17) indicate that the control does not have a significant effect on the separation location or the leading-edge stagnation point. With flow control, the thickness of the recirculation region of the trailing-edge separation is increased slightly compared to the uncontrolled flow. Flow control has a negligible effect on the wall pressure and skin friction coefficient distributions (Figure 18).

Figure 17. Airfoil simulations for Re=1 Million and α=10 deg. Iso-contours of stream function for uncontrolled flow (purple lines) and harmonic blowing and suction with A=0.1 and f=1 (green lines) and f=4 (orange lines). Arrow indicates slot location.

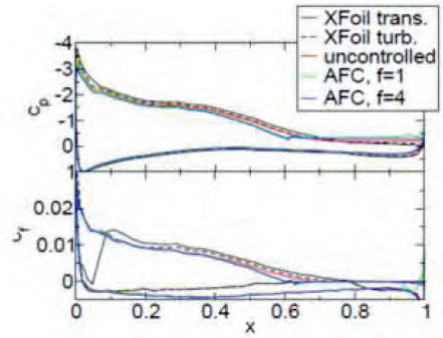


Figure 18. Airfoil simulations for Re=1 Million and α=10 deg. Wall pressure and skin friction distributions.

Finally, lift curves, drag polars, and drag curves are provided in Figure 19. The modified NACA 64_3-618 has a higher maximum-lift coefficient than the original NACA airfoil. This is confirmed by the XFoil calculations. The experiments at Ohio State University (OSU) were carried out for the original airfoil [Thake, et al., 2011]. The experimental lift and drag data for the uncontrolled flow are in fairly good agreement with XFoil data. In the OSU experiments, separation control by steady blowing through a span-wise row of holes was investigated. Some post-stall lift improvement was achieved for α=16deg with a blowing ratio of B=2.02 and for α=20deg with B=2.87. Surprisingly flow control for α=10deg with B=0.29 resulted in a 7% lift reduction. The XFoil results (both, fully-turbulent and based on the e^N transition prediction) for the modified NACA 64_3-618 airfoil indicate a maximum-lift coefficient near α=15deg. The present hybrid simulations provide accurate lift and drag data for α=10deg, but over-predict lift for α=20 and 30deg. The drag prediction remains in reasonable agreement with the XFoil data. Similar to the OSU experiment for α=10deg with flow control, lift is reduced and drag is increased. Further research is needed to corroborate this finding. First and most importantly, simulations based on other, more complex hybrid turbulence models have to be carried out to confirm the validity of the present hybrid turbulence model approach. Then, a larger range of actuator amplitudes has to be considered. The OSU experiments show that blowing velocities as high as 300% of the free-stream velocity are required for a successful trailing-edge separation control. Finally, pure 2D disturbances may not be optimal for trailing-edge separation control. Figure 12 illustrates the considerable 3D character of the flow field, hinting at 3D instabilities that might be exploited for a more efficient flow control.

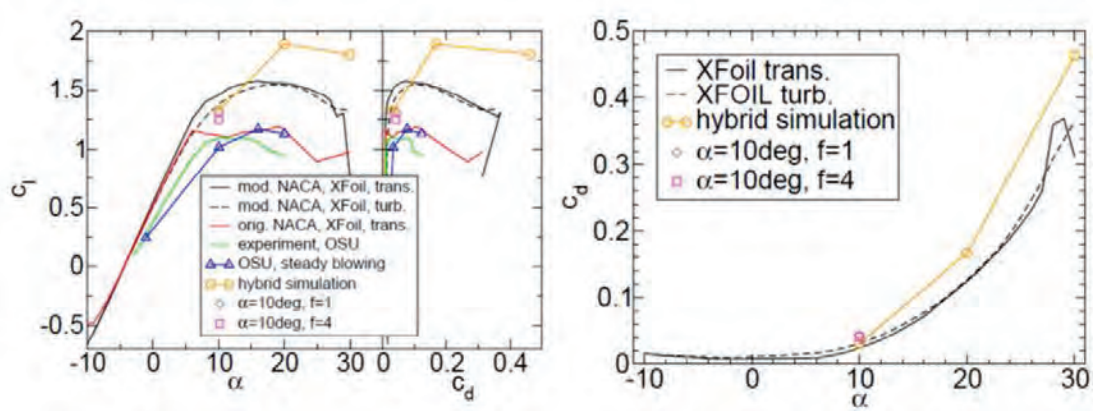


Figure 19. Airfoil simulations and experiments for Re=1 Million. Lift curve, drag polar, and drag curve. OSU data [Thake, et al., 2011] is for original NACA 64_3-618 airfoil.

5. Conclusion

High-resolution Direct Numerical Simulations (DNS) were carried out for a laminar separation bubble on a flat-plate. When the inflow and free-stream mean flow conditions were set to match the respective data from an accompanying water tunnel experiment, the bubble size and characteristics did not match the experimental observations. In particular, the bubble was not shedding. The introduction of weak free-stream turbulence with an intensity that matched the conditions in the experiment resulted in a much-improved agreement with the experimental data. Clearly, "clean" simulations where the free-stream turbulence intensity is zero are not always realistic when separation and transition are investigated. The bubble was found to shed span-wise coherent structures. A detailed analysis revealed that the shedding was the consequence of linear inviscid (inflectional) shear-layer instability. This leads to the conclusion that flow control that exploits the shear-layer instability for amplifying the disturbance input remains a feasible approach in the presence of moderate free-stream turbulence.

Not only the free-stream turbulence, but also the Reynolds numbers are different in full-size free-flight applications when compared to typical laboratory experiments. Fully-resolved DNS at flight-Reynolds numbers are prohibitively expensive. A one-equation hybrid turbulence model was employed for simulations of a modified NACA 64_3-618 airfoil section at a chord-Reynolds number of 1 Million. The agreement of the lift and drag data with panel method data was fair, but leaves room for improvement. Preliminary simulations with harmonic blowing through a slot for 10deg angle-of-attack did not show any performance improvement. Further research is needed to develop more accurate hybrid turbulence models, and to devise flow control strategies that are viable at flight-Reynolds numbers.

6. Significance to DoD

The role of Unmanned Aerial Vehicles (UAVs) in military operations has become crucial in the combined effort to unman the front lines. Small UAV dimensions or high flight-altitudes for larger flight-vehicles result in low-operating Reynolds numbers, which can lead to laminar separation. Laminar separation reduces the usable lift and increases drag. Therefore, laminar separation negatively affects performance (e.g., range, loitering-time, payload). In worst-case scenarios, separation can lead to complete stall and loss of the vehicle. We are developing the necessary tools for investigating separation control strategies for lifting surfaces of small and large UAVs, as well as manned aircraft with the goal of greatly improving aerodynamic performance. We are also investigating how "real-world effects" such as free-stream turbulence and surface roughness are affecting the operation of such vehicles.

Acknowledgments

This work was supported by the US Air Force Office of Scientific Research (AFOSR) under grant number FA9550-09-1-0214 with Dr. Douglas Smith serving as Program Manager. Compute time provided by Challenge Project AFOSR13312C4A is highly appreciated. This work was supported in part by a grant of HPC resources from the Arctic Region Supercomputing Center.

References

Balzer, W., and Fasel, H.F., "Direct Numerical Simulation of Laminar Boundary-Layer Separation and Separation Control on the Suction Side of an Airfoil at Low Reynolds Number Conditions." *AIAA 2010-4866*, 2010a.

Balzer, W., and Fasel, H.F., "Numerical investigation of the effect of free-stream turbulence on laminar boundary-layer separation." *AIAA-2010-4600*, 2010b.

Brehm, C., Mack, S., Gross, A., and Fasel, H., "Investigation of an Airfoil at Low Reynolds Number Conditions." *AIAA-2006-3765*, 2006.

De Langhe, C., Bigda, J., Lodefier, K., and Dick, E., "One-equation RG hybrid RANS/LES computation of a turbulent impinging jet." *Journal of Turbulence*, Vol. 9, No. 16, pp. 1–19, 2008.

Drela, M., and Giles, M.B., "Viscous-inviscid analysis of transonic and low Reynolds number airfoils." *AIAA Journal*, Vol. 25, No. 10, pp. 1347–1355, 1987.

Ghil, M., Allen, R.M., Dettinger, M.D., Ide, K., Kondrashov, D., Mann, M.E., Robertson, A., Saunders, A., Tian, Y., Varadi, F., and Yiou, P., "Advanced spectral methods for climatic time series." *Rev. Geophys.*, Vol. 40, No. 1, pp. 3.1–3.41, 2002.

Gross, A., Pearman, C., Kremer, R., Napier, B., Gosla, C., Kurz, A., Mack, S., Brehm, C., Heine, B., Radi, A., Marovic, B., Retzko, S., Feindler, N., Zickler, B., Fasel, H.F., and Osbrink, A., "1/5 Scale Model of Aeromot 200S SuperXimango for Scaled Flight Research." *AIAA-2008-6416*, 2008

Gross, A., and Fasel, H., "High-Order Accurate Numerical Method for Complex Flows." *AIAA J.*, Vol. 46, No. 1, pp. 204–214, 2008.

Gross, A., and Fasel, H.F., "Active flow control for NACA 6-series airfoil at Re=64,200." *AIAA Journal*, Vol. 48, No. 9, pp. 1889–1902, 2010.

Gross, A., and Fasel, H.F., "Numerical Investigation of Separation for Airfoils." *AIAA-2011-3438*, 2011.

Heine, B., Mack, S., Kurz, A., Gross, A., and Fasel, H.F., "Aerodynamic Scaling of General Aviation Airfoil for Low Reynolds Number Application." *AIAA-2008-4410*, 2008.

Hunt, J.C.R., Wray, A.A., and Moin, P., "Eddies, stream, and convergence zones in turbulent flows." *Report CTR-S88*, Center for Turbulence Research, Stanford, CA, 1988

Jagadeesh, C.S., and Fasel, H.F., "Experimental investigation of the structure and dynamics of laminar separation bubbles." *AIAA-2012-0755*, 2012.

Jeong, J., and Hussain, F., "On the Identification of a Vortex." *J. Fluid Mech.*, Vol. 285, pp. 69–94, 1995

Mack, S., Brehm, C., Heine, B., Kurz, A., and Fasel, H., "Experimental Investigation of Separation and Separation Control on a Laminar Airfoil." *AIAA-2008-3766*, 2008.

Meitz, H.L. and Fasel, H., "A Compact-Difference Scheme for the Navier-Stokes Equations in Vorticity-Velocity Formulation." *J. Comp. Phys.*, Vol. 57, No. 1, pp. 371–403, 2000.

Plogmann, B., Mack, S., and Fasel, H.F., "Experimental Investigation of Open-and Closed-Loop Control for Airfoil under Low Reynolds Number Conditions." *AIAA 2009-4282*, 2009.

Radi, A., and Fasel, H.F., "Experimental investigation of laminar separation bubbles on a flat-plate." *AIAA-2010-4482*, 2010.

Sirovich, L., "Turbulence and the Dynamics of Coherent Structures." *Quarterly of Applied Mathematics*, Vol. XLV, No. 3, pp. 561–590, 1987.

Thake, M., Packard, N., Bonilla, C., Gompertz, K., and Bons, J., "Reynolds Number Scalability for Separation Control on a Laminar Airfoil." *AIAA-2011-1182*, 2011.

6.1 Basic Research

Army

3D Computational Analysis of the Turbulent Mixing of an Angled Jet in Crossflow

Luke Grant
Department of Civil & Mechanical Engineering, United States Military Academy, West Point, NY
luke.grant@us.army.mil

Abstract

Computational results are compared to experimental data for the scalar mixing of two fluids in an incompressible turbulent flow. The apparatus investigated consists of an angled jet entering a crossflow with a 49.3 mm square cross-section. The jet ejects from a single elliptical hole with a length-to-diameter ratio of 6.0 into a crossflow, at 30 degrees pitch and zero skew angles, with a jet to mainstream velocity ratio of 1.0. The elliptical hole is of minor diameter 6 mm in the span-wise direction, and of major diameter 12mm in the stream-wise direction. A counter-rotating vortex pair emerges from the feed-hole, and is sustained through the end of the measurement region over 10 hole-diameters downstream. The three-dimensional, three-component velocity and scalar concentration fields are compared to Reynolds-Averaged Navier-Stokes simulations of the tested geometry using an unstructured grid. Detailed analysis of the similarities and differences between the computational results and experimental data is presented for the flow in the film hole, the crossflow velocity, and the crossflow concentration of the converged flows. A discussion of improvement of the simulation is presented through the implementation of a modified constant turbulent Schmidt number, improved turbulence intensity, or improved boundary condition of the inlet plane of the water channel. The simulations are developed using computational fluid dynamics (CFD)++11.1.1® with support from the US Army Research Laboratory (ARL) at Aberdeen Proving Grounds, MD. High performance computing capabilities facilitated by a United States Military Academy-ARL partnership are required for the computer-intensive Reynolds-averaged Navier-Stokes simulations. One salient, focused application of this research is the model improvement of fluid flow over gas turbine engine blades. Continued subsequent research provides immense potential in improving simulations of complex turbulent mixing flowfields.

1. Introduction

1.1 Project Overview

Gas turbine engines reach very high-temperature levels that endanger their blades by exceeding their natural melting points. These blades are successfully cooled through jets of air derived from film-cooling holes within the blade. The interaction of the cooling jet through the holes and across the surface of the blade demonstrates mixing of two converging flows. Bunker (2005) provides an extensive survey of shaped film-cooling technology used to cool gas turbine airfoils. He concludes that shaped film- cooling can have many benefits including: low-sensitivity to free-stream turbulence intensity variations, low-susceptibility to film blow-off under typical conditions, and little variation of effectiveness given a variety of blowing ratios. An optimization in the design of shaped film-cooling holes can only strengthen the technology and lead to better performance when applied to high-temperature airfoils. A greater understanding of the inherent complexities of the mixing of fluids in turbulent flow can greatly improve computer modeling capabilities. These enhanced simulation capabilities will only further strengthen industrial design and manufacturing abilities.

A computational analysis of turbulent mixing of an angled jet in crossflow is performed using the three-dimensional (3D) geometry displayed in Figure 1. This geometry displays an elliptical red tube dispersing a fluid into a highly-turbulent, larger crossflow with a 49.3 mm square cross-section. The coordinate system has its origin at the center of the elliptical tube exit along the floor of the square cross-section channel. The direction of stream-wise flow is designated the X-direction, span-wise movement is designated the Z-direction, and movement normal to the floor of the channel is designated the Y-direction. The fluid dispersed from the tube is referred to as the coolant flow. A more thorough understanding of the turbulent mixing will improve the capabilities of computer simulations and subsequently benefit the

design capabilities of the engineering community. Improved modeling capabilities reduce the need for long and costly fluid dynamic experimentation. The value of computer simulations is judged upon their ability to replicate validated experimental results.

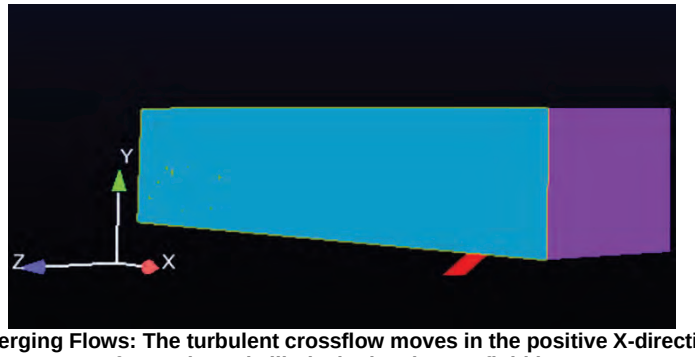


Figure 1. 3D Geometry of Converging Flows: The turbulent crossflow moves in the positive X-direction exiting through the purple face. The red elliptical tube ejects a fluid jet.

The emergence of modern computational fluid dynamics (CFD) has provided greater opportunities for scientists and engineers to conduct significant research into turbulent mixing. These CFD capabilities are aided by modern high-performance computing capabilities. A fundamental understanding of flow physics allows for model improvement. Research into various CFD simulation techniques provides an understanding of typical modeling techniques.

A thorough discussion of diverse modeling techniques is provided by the work of Acharya and Hoda (2000). They analyze seven different turbulence models against a contemporarily valued experimental standard. They state that many models give overly simplistic conclusions to the highly-complex fluid field being considered. They also explain that the most common approach is the use of the Reynolds-averaged Navier-Stokes (RANS) Solver, which requires a turbulence model. They suggest that Claus and Vanka fail to predict proper vortex nature due to the inability of the high-RE model (HRE) to solve complex turbulence fields. The HRE over-predicts the span-wise jet penetration. Additionally, the HRE is said to be unsuited for the experimental flow situation since it does not resolve the near-wall region properly.

Leylek and Zerkle (1994) model the uses of gas turbine cooling parameters for airfoil applications. Their work is a beneficial resource, because the authors explain the details of the computational model, including the method of applying a grid-mesh in three-dimensions. They also explain the use of applying boundary conditions for the six boundary surfaces to most closely replicate the experimental work. Their simulation inputs also included a turbulence intensity of 0.2 percent and a velocity profile using “plug-flow”, since the inlet plane was placed at the leading-edge of the test plate where a new boundary layer was triggered. They finally explain that the pressure level was also specified at the exit plane, along with zero stream-wise gradient for all other dependant variables. Leylek and Zerkle also mention that their cross-stream and film-hole were initialized separately and as accurately as possible. These topics served as a beneficial means to better understand the requirements of completing a comprehensive list of CFD inputs into numerical simulation software programs.

The 2011 work of Bai, Jiang-tao, et al. is also of immense benefit. Although they address new developments regarding the turbulent Prandtl number, their explanation of CFD methodology was quite beneficial in the following experimental work. They showed that a body-fitted, multi-block grid allowed the highest quality in all regions with the fewest number of cells. This supports the use of unstructured meshing to better fit the desired needs of precision for various locations within the flow’s geometry. The domain was partitioned into subsections as follows: the cooling-hole, the jet plenum, and three subsections of the main channel. A large amount of information regarding the construction of mesh data from previous work was then referenced from six unique publications. One of the listed tests utilized a row of cylindrical holes with compound angle orientation, and another utilizes a row of double-fan-shaped holes. Two others that were more closely related to the simulated geometry were those completed by Goldstein, et al. (1968) and Sinha, et al. (1991). Goldstein measured the film cooling effectiveness downstream of a long cylindrical hole inclined at 35° in the streamwise direction. He completed computations at the three blowing ratios of 0.5, 1.0, and 1.5. Bai explains that Sinha conducted similar tests utilizing a row of cylindrical holes, which inclined stream-wise with angle of 35° with a blowing ratio of 0.5. Experimental setup strongly validates the testing methodology of the chosen geometry to undergo simulation in this report.

Upon completion of research into the construction of a CFD simulation utilizing a RANS simulation with a k-ε turbulence model, research into the turbulent Schmidt number is conducted. Stathopoulos and Yoshihide (2007) complete an extensive literature review of the various turbulent Schmidt numbers used within CFD analysis for varying flowfields. They explain that no universal value of the turbulent Schmidt number has been established, and many different values are

utilized given the scenario. These values used ranges as widely as 0.2–1.3, even though they can have a significant impact on computational predictive results. For the specific case of predicting the turbulent scalar fields in jet-in-cross-flow with a standard k-ε model, Yimer, et al. (2002) concludes that a turbulent Schmidt number of 0.7 should be recommended for use in CFD applications that, "involve axisymmetric free-jet flows by considering the consistency with the prediction accuracy of the velocity field." Stathopoulos and Yoshihide suggest that the work of Yimer, et al. maintains a much stronger argument than the 1999 work of He, et al. (1999) that recommends a turbulent Schmidt number of 0.2. Although this value best matches the experimental data set used by He, et al., this low value may very well be a product of a large underestimation of the predicted turbulence intensity around the jet. To further support the work of Yimer, et al., the later work of Huai, et al. (2004) displays the use of a constant turbulent Schmidt number of 0.7 between the turbulent diffusion coefficient and the turbulent viscosity.

Literature review of the jet-in-cross-flow scenario strongly shows that the turbulent Schmidt number is often chosen to be 0.7 for CFD applications, yet highly-optimized simulations may better replicate experimental data if the turbulent Schmidt number may be found to be a more precisely defined constant, or possibly even spatially-variable throughout the 3D geometry to be modeled. Use of highly-precise flow-dependent constant values is seen in the 2002 work of Baurle and Eklund that recommends Sc_t= 0.5 for their simulation of reacting scramjet flows. Sturgess and McManus (1984) display the range of variation of the Sc_t from 0.2 to 0.7 for in a diffusion flame combustor. Batten, et al. explains that in the case of large variations, the Sc_t may be modeled as a function of the k-ε turbulence model, which is "modified by sensitizing the parameter, Cμ, to the ratio P_k/ε (ratio of turbulence kinetic energy production and its dissipation rate) with a similar treatment of the eddy diffusivity." This source is also beneficial because it explains the benefits of using a spatially-varying Schmidt number to increase model precision; yet its explanation of implementation did not cover the case of incompressible fluids. Such a limitation would prove to greatly limit the capabilities of simulations of incompressible flow given the use of software from Metacomp Technologies.

1.2 Nomenclature

Cμ	coefficient in eddy viscosity formula
Sc_t	turbulent Schmidt number
D	mass diffusivity [m^2/s]
$\dot{m}$	mass flow rate [kg/s]
Q	volumetric flow rate [LPM]
Re	Reynold's Number $= \frac{\bar{V}D}{\mu}$
M	molecular weight [kg/kmol]
ρ	density [kg/m^3]
μ	dynamic viscosity [kg/ms]

Subscripts

main	cross-flow
cool	coolant
vel,fluid	fluid used in the velocity simulation
con,main	water, used as crossflow in concentration simulation
con, cool	fluid used for coolant in concentration simulation

1.3 Objectives of Work

1.3.1 Completion of a 3D velocity and concentration field, accounting for the fluid movement of the cross-flow jet, assuming steady flow and no slip condition on the apparatus boundary.

1.3.2 Improvement upon the initial completed simulation described in the initial objective, and attempted through the implementation of a more accurate turbulence intensity, or the use of an improved boundary condition of the inlet plane of the water channel geometry. Detailed explanation can be found within Simulation Procedures.

1.3.3 Further improvement through the determination of the scalar isotropic diffusivity to a much higher level of precision for an angled jet in cross-flow, ultimately allowing for needed improvements with respect to the film-cooling of turbo jet engines.

This final objective may be completed through further use of the Reynolds-averaged Navier-Stokes equation modeling to gain a greater understanding of the turbulent Schmidt number (Sc_t).

$$Sc_t = \frac{\mu}{\rho D} \tag{1}$$

Equation 1 displays the turbulent Schmidt number, which is simply a ratio of the viscous diffusion rate to the molecular diffusion rate. This parameter is useful to be defined when considering a conservation of momentum as done in the following simulations. The value of turbulent mass diffusivity can be calculated throughout the 3D flowfield from experimental results using partial differentiation of the scalar concentration throughout the experimental geometry. Additionally, the momentum diffusivity can be determined by obtaining the eddy diffusivity, at the same discrete points, as a function of the k-ε turbulence model used within CFD++11.1.1®. Use of these values may lead to the determination of a more accurate, constant Schmidt number to be used in CFD simulation. This may allow an improvement of the simulation capabilities to replicate the experimental results (Benson, 2011).

2. Simulation Procedures

The simulations of this study were compared to experiments that used magnetic resonance velocimetry (MRV) and magnetic resonance concentration (MRC) measurement techniques at 0.24 mm^3 resolution. The MRV experiment utilized a single-fluid of copper sulfate mixture and was used to obtain velocity data of the fluid within the tested region. The MRC experiment used two fluids, and was used to obtain concentration data of the fluids within the tested region. The mixed fluids for the concentration measurements were plain de-aerated water in the mainstream, and water with copper sulfate at a concentration of 0.015 mol/L for the jet (Benson, 2011). Data used within these simulations is provided in Table 1.

Table 1. Simulation parameters for velocity, concentration

Q_{main}	71LPM
Q_{cool}	0.79LPM
$\dot{m}_{main}$	1.18 kg/s
$\dot{m}_{cool}$	0.0126 kg/s
$Re_{main,x}$	98,400
$Re_{cool,d}$	2,900
$\mu_{vel,fluid}$	0.00091 kg/ms
$\rho_{vel,fluid}$	1008 kg/m^3
$M_{vel,fluid}$	41.2 kg/kmol
$\mu_{con,main}$	0.00111 kg/ms
$\mu_{con,cool}$	0.0009 kg/ms
$\rho_{con,main}$	998 kg/m^3
$\rho_{con,cool}$	1002 kg/m^3
$M_{con,main}$	18.0 kg/kmol
$M_{con,cool}$	26.7 kg/kmol

The CFD simulations were done using the fundamental Navier-Stokes Equations which represent the conservation of Linear Momentum. An in-depth analysis of this law is not intended to be undertaken in this analysis; rather, a fundamental understanding of its concepts validate the testing for conservation of linear momentum used to support the simulation results.

$$\rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \rho \bar{f}_i + \frac{\partial}{\partial x_j}[-\bar{\rho}\delta_{ij} + \mu(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}) - \rho \overline{u_i' u_j'}] \tag{2}$$

Equation 2, known as the Reynolds-averaged Navier-Stokes equation (RANS), is a time-averaged solution to the Navier-Stokes equations, and is applied to the turbulent, incompressible flow, which is characteristic of the experimental work. The RANS equation is a powerful tool because it allows for the unknown Reynold's stresses to be modeled in terms of the known mean-flow. The simulated steady-state flow is thus able to gain "closure". RANS requires the implementation of a turbulence model. This simulation used the HRE, High-Reynolds number model.

The step-by-step procedures for the creation of the simulations are as follows.

2.1 Geometric Orientation

The geometric dimensions were provided by LTC Michael Benson to model previously highly-validated experimental work (2011). The interior of the channel was modeled as a solid and was meshed for computer calculation of the fluid behavior. A quick orientation to the coordinate system, first displayed in Figure 1, will reveal the crossflow direction of travel in the positive X-axis. The positive Y-axis moves normally from the floor of the channel. Finally, the X-axis moves in the span-wise direction, from side-to- side of the water channel. It is a body-fixed coordinate system with the origin at the center of the coolant exit tube.

2.2 Grid Mesh

An anisotropic discretization of the water channel and feed pipe containing 1.3 million control volumes resulted from the meshing process. An unstructured mesh was utilized to allow greater precision in the discretization of the coolant tube and the portion of the main channel where the fluids would converge. The benefit of unstructured meshing is apparent in separation of the entire geometry into six components for designation of grid shape and resolution. The mesh shape chosen for the walls parallel to the fluid flow was quadrilaterals; whereas the mesh shape chosen for inlet and exit planes orthogonal to fluid flow were triangles. The literature supported the use of the finest discretization of the geometry near and directly downstream of the hole exit where the mixing would occur.

2.3 Boundary Conditions

Boundary conditions were assigned for each component of the complete geometry. The portions of the geometry that had the fluids flow across their surface were designated as wall-boundary conditions and boundary-layers of the proper height were assigned. Those portions of the geometry that had fluid flow through their surfaces were assigned conditions characteristics to two-dimensional (2D) inlet and exit planes.

2.4 Initial Conditions

The initial conditions required the listing of the fluid properties for the one fluid within the velocity simulation and the two fluids within the concentration simulation. These properties are listed in Table 1. This initialization was also important because it determined the starting character of the fluids as they entered the mesh. The simulation utilized pressure parameters during the input of the inlet plane characteristics.

2.5 Turbulence Initialization and Modeling

This turbulence initialization used the RANS model, as previously mentioned. As an HRE model, the turbulence of the freestream velocity was set at 0.01, a value suggested for internal flows using the k-ϵ model chosen, where k is the turbulent kinetic energy and ϵ is the turbulent energy dissipation rate. Such input is likely compatible with the inherent nature of the k-ϵ model providing isotropic diffusivity. Turbulence initialization proves to be largely a characteristic of the nature of the flow and the dimensions of the geometry.

2.6 Convergence

Convergence of the simulation occurs after an iterative input of up to 500 time-steps achieves a pre-determined negligible residual value of less than $1.0e^{-8}$. This requirement is met when the combined error for a given variable changes to such a small amount between iterations that further iteration is deemed unnecessary.

2.7 Comparison of Simulation to Experiment

The simulation will be compared to the experimental data in regard to the nature of the flow in the film hole, as well as, the velocity and concentration of the cross-flow after convergence of the two fluids. This analysis will be focused along the Z-plane in the middle of the channel, and on the X-plane at distances of one diameter and ten diameters downstream of the jet exit of the film-hole.

2.8 Post Simulation Improvements

Improvements upon the initial completed simulation were attempted in the three areas of implementation of an improved turbulence intensity value, an improved boundary condition of the inlet plane of the water channel, and a more accurate, modified constant turbulent Schmidt number.

The turbulence level can be adjusted to more closely model the experimental results. The freetream turbulence level typically varies from 0.01 to 0.05 for incompressible flow. Such an input is needed in CFD unless there is experimentally provided data for the turbulent kinetic energy and its dissipation rate. The initial simulation utilized a free-stream turbulence level of 0.01, and all levels to a maximum of 0.05 were utilized in simulation.

The boundary condition for the inlet of the fluid flow through the simulation geometry was a function of the mass flow rate in the initial completed simulations. The implementation of a 2D velocity profile at this boundary condition would have potential to improve the accuracy of the simulation. This velocity profile would be of high-potential, because it was experimentally obtained from a highly-precise experimental work. Implementation of this 2D velocity profile was completed as an inlet boundary condition at 64.5mm upstream of the film-hole, causing the use of a shortened geometry for simulation. This alternative boundary condition was implemented for all test cases used, yet verifiable high-quality results could not be repeatedly obtained in this study.

Provided the inability of Metacomp software to utilize a spatially-varying turbulent Schmidt number in incompressible flows, various constant turbulent Schmidt numbers were implemented from a range of Sc_t=0.5-0.9, utilizing a cubic k-ϵ turbulence model (Stathopoulos, 2007) (Batten). Verifiable, high-quality results could not be repeatedly obtained in this study.

Inability to achieve verifiable results utilizing the entire range of 2D velocity profile inlet plane boundary conditions and the entire range of constant turbulent Schmidt numbers during simulation provided an improved simulation through optimization of the free-stream turbulence level within the CFD simulations. This improvement is referred to as Improved Simulation within the subsequent discussion of results.

3. Results & Discussion

The simulation resulted in the completion of a 3D velocity and concentration field for the entire geometry. Use of the post-processing software Ensight 9.2® provides clear visualization of the required fluid properties throughout the geometry, allowing for accurate analysis of the validity of the simulation results. Results are also presented for an Improved Simulation utilizing a free-stream turbulence level of 0.04, to more closely model the behavior of the experimental fluid convergence.

3.1 Details of Flow in the Film-Hole

The flow within the film-hole displayed the jetting effect repeatedly seen in the literature for the 30 degree angle-of-discharge. Figure 2 displays this phenomenon through the coolant with the highest streamwise velocity moving toward the cross-flow at the top of the tube. The method of feeding was parallel to the center-line of the tube, a stark difference from the method of side-feed from a plenum as in the experimental data. A predicted sharp transition of coolant flow direction upon interaction with the cross-flow was also validated within simulation results.

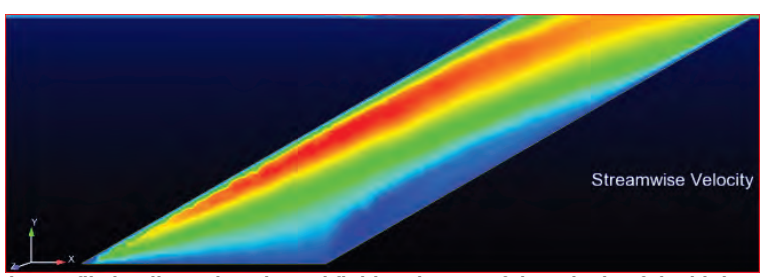


Figure 2. Streamwise Velocity Profile in Film Hole. The red fluid at the top of the tube is of the highest velocity, representing the jetting effect.

3.2 Details of Cross-flow Velocity Analysis

The streamwise velocity field was analyzed by examining the 2D streamwise profiles of the velocity from the floor of the channel to a height of 20mm. This was done along the X-Y plane at Z=0. This profile was taken at a distance of 1 hole diameter downstream in Figure 3, and at a distance of 10 hole diameters downstream in Figure 4. Both axis measurements are non-dimensional. The streamwise velocity is presented as (U/V_{mean}), a ratio of streamwise velocity to average velocity over the plotted domain. The channel height is presented as (Y/D), a ratio of the height to the diameter of the film-hole. At one hole diameter downstream, Figure 3 shows that the Improved Simulation more closely models the fluid mixing behavior than the initial simulation that has a large velocity difference near the floor of the channel. At a distance ten diameters downstream as seen in Figure 4, the improved simulation appears to over-predict the fluid velocity within the first 5mm from the floor of the channel. The improved simulations results appear to be reasonably valid, yet improvement is possible.

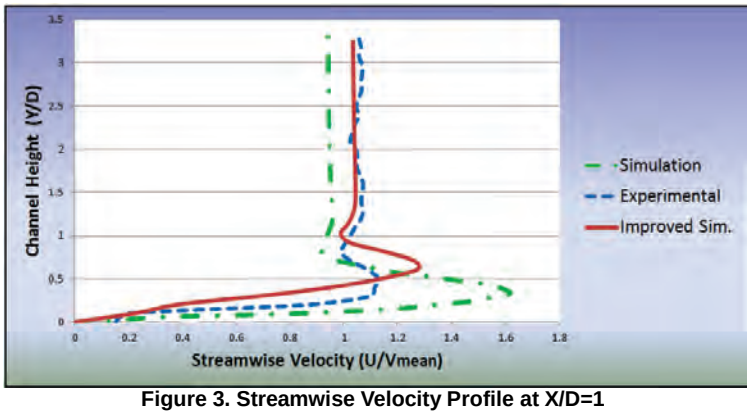


Figure 3. Streamwise Velocity Profile at X/D=1

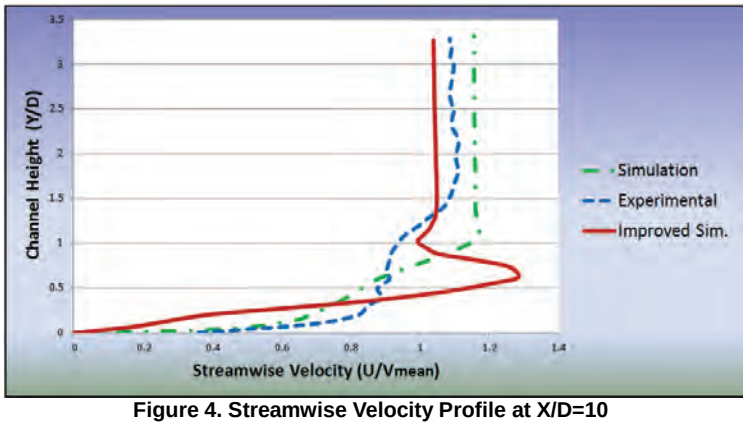


Figure 4. Streamwise Velocity Profile at X/D=10

3.3 Details of Cross-flow Concentration

The concentration analysis may best be understood by an initial look at a 2D profile of the length of the cross-flow downstream of the coolant tube in the stream-wise direction when Z=0. This reveals that the flow in the simulation did not reattach to the floor of the channel like the experimental results appeared to do successfully. Figure 5 and Figure 7 display this disparity in coolant travel downstream. It is also apparent that the simulation's coolant requires a longer distance of travel downstream before it mixes to the same extent as that witnessed in experimental work, yet the higher turbulence level of the improved simulation promotes a higher level of mixing that more closely replicates the experimental data. Closer analysis of the concentration is conducted through analysis in the span-wise direction.

Analysis of a 2D profile of the span-wise direction when viewing the X-plane provides the most analytically profitable information within comparison of data sets. Further regional analysis occurs a distance of one diameter downstream of the film-hole as displayed in Figures 8, 9, and 10, as well as ten diameters downstream of the film-hole, as displayed in Figures 11, 12, and 13. A key note is the isotropic nature of mixing of the experiment compared to the largely anisotropic mixing of the coolant in the initial simulation. The Improved Simulation provides a more even distribution of mixing that closely

replicated the nature of the experimental data. This improvement is much better than the complete lack of isotropic mixing shown displayed ten diameters downstream in the initial simulations. This isotropic nature of mixing is not directly sought, but it is desired given the characteristic of the experimental mixing appearing generally isotropic. The isotropic nature of the RANS k-ϵ model is successfully displayed given the optimum turbulence level. An insufficient turbulence level appears to more closely model the excessive span-wise spreading mentioned in the work of Acharya and Hoda (2000). A removal of any excessive span-wise spreading in the Improved Simulation is indicative that the simulation successfully displays a properly-oriented counter-rotating vortex pair at lengths of at least ten diameters downstream.

The Improved Simulation, shown in Figure 9 and Figure 12, also appears to provide a higher-level of mixing than the initial simulations, although it still does not meet the higher-levels of mixing displayed in the experimental profiles of Figure 10 and Figure 13. At a distance of X/D=1, the experimental data displays only a small-level of pure coolant, yet the improved simulation still maintains a larger section that has yet to mix with the cross-flow. At a distance of X/D=10, this phenomenon is again witnessed. The highest level of coolant concentration of the experimental data is approximately 30%, yet the Improved Simulation maintains a coolant fluid concentration as high as approximately 50%.

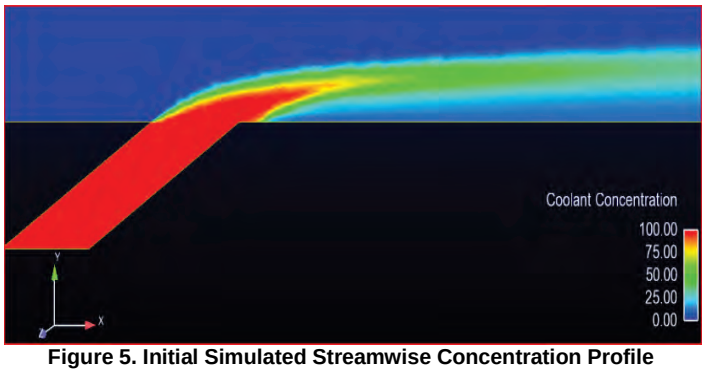


Figure 5. Initial Simulated Streamwise Concentration Profile

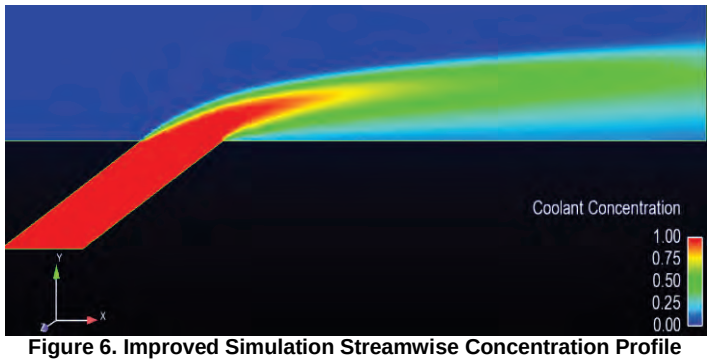


Figure 6. Improved Simulation Streamwise Concentration Profile

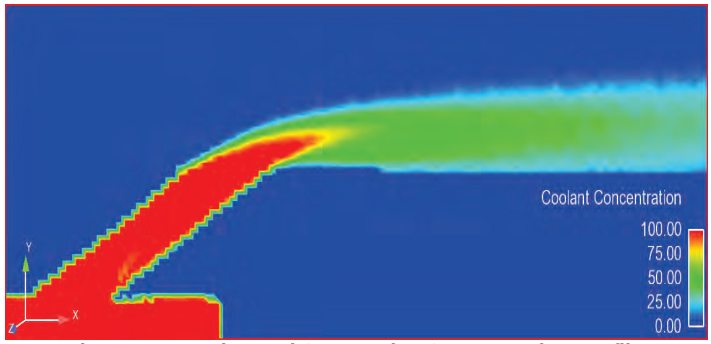


Figure 7. Experimental Streamwise Concentration Profile

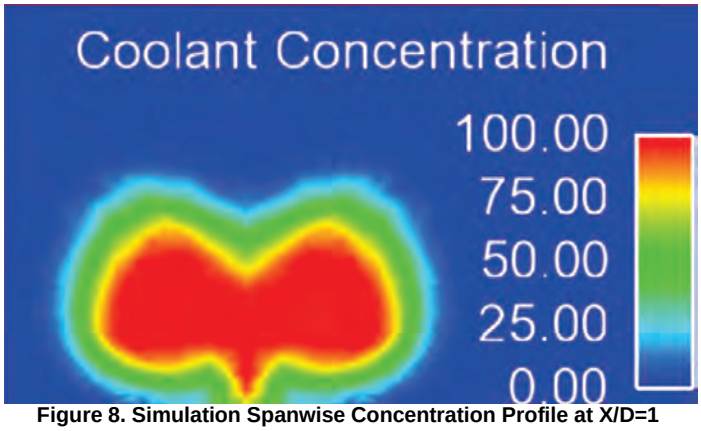


Figure 8. Simulation Spanwise Concentration Profile at X/D=1

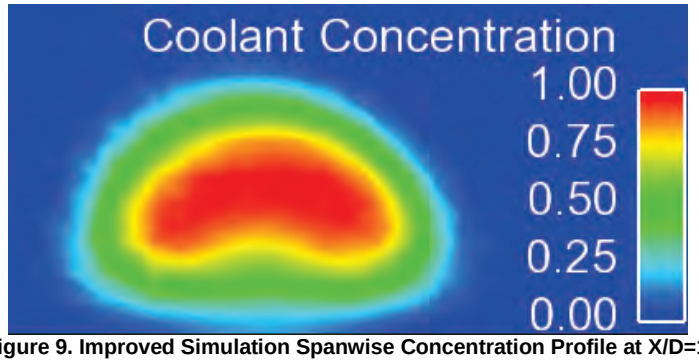


Figure 9. Improved Simulation Spanwise Concentration Profile at X/D=1

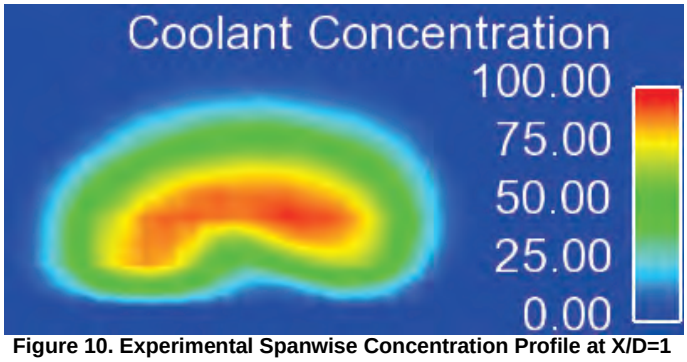


Figure 10. Experimental Spanwise Concentration Profile at X/D=1

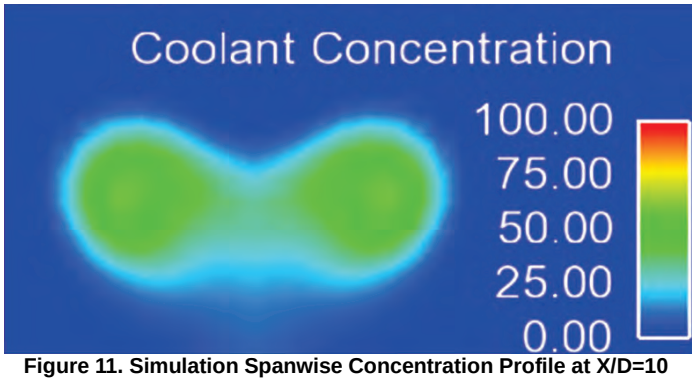


Figure 11. Simulation Spanwise Concentration Profile at X/D=10

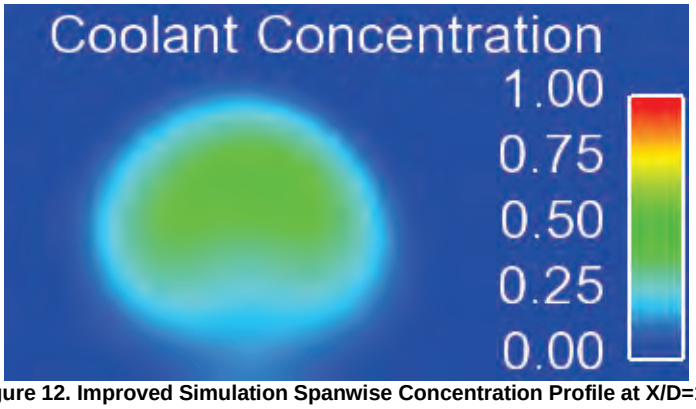


Figure 12. Improved Simulation Spanwise Concentration Profile at X/D=10

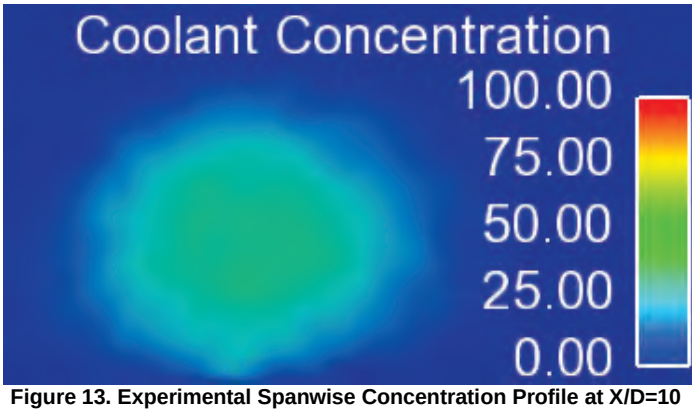


Figure 13. Experimental Spanwise Concentration Profile at X/D=10

3.4 Implications

The Improved Simulation proved to be a much stronger match to the experimental data set. Implementation of a free-stream turbulence level of 0.04 was much more appropriate to replicate the necessary higher level of mixing inherent in the experiment. This Improved Simulation proves to be a positive gain, because it more closely resembles the isotropic nature of the experiment and more closely mirrors its level of mixing than the initially-completed RANS realizable k-ϵ turbulence model simulations. There is room for improvement in the area of simulating a greater-level of mixing of the converging fluids. It would also be very beneficial to create simulated results that allow the coolant jet to reattach to the floor of the channel. This additional mixing and reattachment of the coolant jet to the channel floor are both necessary tools to more closely simulate the useful industrial application of cooling air inciting heat transfer away from the surface of very hot gas turbine engine blades.

4. Conclusion

The simulated results made gains in replication of experimental results throughout the period of CFD study and implementation. The HRE turbulence model chosen may not have had the level of documented historical success as a more computationally-intensive Large-Eddy Simulation, yet it is most commonly used and has provided reasonable results (Acharya, 2000). Velocity analysis along the X-plane demonstrates that the simulation is not greatly different than the experimental work downstream of the coolant-hole. Improved simulation replicates the isotropic nature of fluid mixing, as displayed in the Span-wise Concentration profiles of Figure 9 and Figure 12. Improvements must be made to more closely replicate the reattachment of the coolant jet to the channel floor, as well as, promote more mixing of the fluids downstream of the film-hole. Additional work may be continued to properly implement a 2D velocity profile as the boundary condition of the inlet plane of the channel or determine the optimal Sc_t for the flow properties. Use of the magnetic resonance velocimetry and magnetic resonance concentration measurements (Benson, 2011) to obtain the turbulent diffusivity to generate a regionally-specific averaged optimal Schmidt number represents a powerful tool to improve simulations of complex turbulent mixing flow fields.

Acknowledgements

The author would like to thank the following individuals for their contributions to this research: LTC Michael Benson, Department of Civil and Mechanical Engineering, West Point, New York; and Dr. Sidra Silton, US Army Research Laboratory, Aberdeen Proving Ground, Maryland. Your assistance was very helpful and greatly appreciated.

References

Acharya, Sumanta and Asif Hoda, "Prediction of a film Coolant Jet in Crossflow with Different Turbulence Models." *Journal of Turbomachinery*, 122, pp. 558–568, 2000.

Bai, Jiang-tao, et al., "New development of the turbulent Prandtl number models for the computation of film cooling effectiveness." *International Journal of Heat and Mass Transfer*, 54, pp. 874–886, 2011.

Batten, Paul et al., *Variable Turbulent Schmidt and Prandtl Number Modeling*, Metacomp Technologies, Inc., Agoura Hills, CA.

Baurle, R.A. and D.R. Eklund, "Analysis of dual-mode hydrocarbon scramjet operation at Mach 4–6.5." *Journal of Propulsion and Power*, 18, pp. 990–1002, 2002.

Benson, Michael, and May, Experimental work completed during research at Stanford University, provided to author through Ensight9.2® data files of flow characteristics and oral communication, October 2011.

Bunker, Ronald S., "A Review of Shaped Hole Turbine Film-Cooling Technology." *Journal of Heat Transfer*, 127, pp. 441–453, 2005.

Goldstein, R.J., et al., "Film Cooling with injection through holes: adiabatic wall temperature downstream of a circular hole, *Journal of Engineering Power*, 90, pp. 384–3951968.

He, G., Andrew, Y.G., and Hsu, T., "The effect of Schmidt number on turbulent scalar mixing in a jet-in-crossflow." *International Journal of Heat and Mass Transfer*, 42, pp. 3727–3738, 1999.

Huai, Y., et al., "Comparative study of turbulent mixing in jet in cross-flow configurations using LES." *International Journal of Heat and Fluid Flow*, 25, pp. 767–775, 2004.

Leylek, J.H. and R.D. Zerkle, "Discrete Jet Film Cooling: A Comparison of Computational Results with Experiments." *Journal of Turbomachinery*, 116, No. 3, pp. 358–367, 1994.

Sinha, A.K., et al., "Film cooling effectiveness downstream of a single row of holes with variable density ratio." *ASME Journal of Turbomachinery*, 113, pp. 442–449, 1991.

Stathopoulos, Ted and Yoshihide Tominaga, "Turbulent Schmidt Numbers for CFD analysis with various type of flow field." *Atmospheric Environment*, 41, pp. 8091–8099, 2007.

Sturgess, G.J. and K.R. McManus, "Calculation of turbulent mass transport in a bluff-body diffusion-flame combustor." *AIAA Paper 84-0372*, 1984.

Yimer, I., Campbell, I., and Jiang, L.Y., "Estimation of the turbulent Schmidt number from experimental profiles of axial velocity and concentration for High-Reynolds-Number Jet Flows." *Canadian Aeronautics and Space Journal*, 48, pp. 195–200, 2002.

Aerodynamic Performance of Perforated Wings for Micro-Aerial Vehicle Application

Shawn Aram, Rajneesh Singh, Surya Dinavahi, Aaron Harrington, and Christopher Kroninger
US Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
{shawn.aram.ctr, rajneesh.k.singh.civ, surya.p.dinavahi.ctr, aaron.m.harrington2.civ, christopher.m.kroninger.civ}@mail.mil

Abstract

Numerical computations with companion experiment are conducted to examine the aerodynamic performance of flapping wings at ultra-low Reynolds numbers for micro-aerial vehicle application. Two wings with the same plan-form and thickness are considered for the current study—Solid wing and Comb wing). Two different flow solvers are used to conduct the numerical study of the flapping motion of both the wings at three different flow conditions. Simulations are carried out to compute aerodynamic forces and flowfield. Experimental studies were also conducted for the same wings in oil to measure aerodynamic forces at the lowest Reynolds number used in the computational simulations. Both solvers demonstrated good agreement with the experimental data. The aerodynamic forces for Solid wing and the Comb wing were quite similar at lower Reynolds number; however, the detailed examination of the forces showed that the force generation mechanisms on the two wings are quite different.

1. Introduction

The ongoing development of millimeter scale flapping wings (Pulskamp, et al. (2010)) has engendered an interest in ultra-low Reynolds number flow physics. The biologically-inspired concept explored here is drawn from the Thrips insect, which flies at a Reynolds number sufficiently low to exploit a wing composed of hairs arranged in a comb-like distribution emanating from a spanwise "spar" rather than a solid membrane. Creeping flow is experienced in the region of the comb hairs and minimal flow has been observed to penetrate the hair "membrane" (Kuthe, 1975). Engineered wings of this type have been exploited for micro-aerodynamic decelerators (Zussman, et al. (2002)) which result in small, slowly descending platforms similar in behavior to dandelion seeds. More recent work applied the comb concept to millimeter-scaled flapping wings (Rozen, et al. (2011)).

In application to millimeter-scaled flapping wings, the performance of comb design is compared relative to the performance of traditional solid plan-form designs of Bronson, et al. (2009). These devices are MEMS coupled wing/actuator systems. The actuators for the wings are thin film PZT with later designs incorporating two orthogonally-oriented actuators which control the wing flap and pitch degrees-of-freedom. The flapping degree-of-freedom is designed to operate at the primary system resonant frequency which corresponds to a flapping motion, while pitch is controlled quasi-statically (Pulskamp, et al. (2010)). The introduction of comb design facilitates tailoring mechanical system properties independent of the aerodynamic performance of the wing assuming the Comb wings demonstrate an aerodynamic response similar to the Solid wing. This could facilitate lower operating resonance frequencies or larger wing areas for the same actuator authority, and the incorporation of orthogonal wing elastic properties which might, for example, facilitate passive twisting while resisting span-wise bending to eliminate the need for a pitch actuator. Achieving these benefits relies on the functioning of the creeping flow mechanism at chord-Reynolds numbers in the tens to low hundreds.

Studies of the concept wing design in a Reynolds-scaled experiment in an oil-medium have suggested that the mechanism is extensible to tens of Reynolds numbers, based on wing chord, but proper scaling of all geometric properties is not achieved in the current experiment. Computational studies are performed to explore the aerodynamic performance of perforated wings and compared with conventional solid wing of the same planform. This paper will first perform direct comparisons between experimentally-obtained results and the computational fluid dynamics (CFD) simulations. Detailed

investigations are also carried out into the origins of forces for various cases and flowfield information is utilized to explain the observed behavior. Additional computational studies will consider additional wing kinematics.

2. Computational Setup

Figure 1 shows a schematic of two wings considered in the current study: a) Solid wing and b) Comb wing with the same planform and thickness. The dimensions of the wings, as well as width and length of the slits of the Comb wing, are given in Figure 1 which matches the dimensions of the wings used in the experiments. The performance of the wings is examined for flapping motion of the wing around the axis parallel to the z direction. Sinusoidal variation of angle with time is considered for this motion: $\theta=\theta_{max}\sin(2\pi ft)$, where f is the frequency of the motion. θ_{max} of 15 degrees is used in the current study.

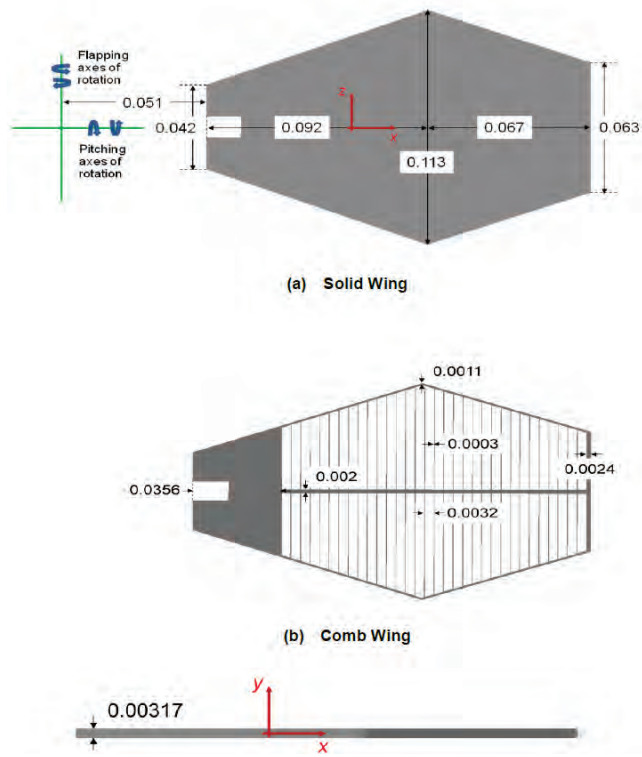


Figure 1. Top (*x-z*) and side (*x-y*) views of the wing used in the study; all dimensions in meters

Two different flow solvers are used to conduct the numerical modeling of motion of the wing: AERO-F and CFD++. AERO-F is a domain decomposition-based, three-dimensional, compressible, steady/unsteady Euler/Navier-Stokes, laminar/turbulent solver, based on finite-volume and finite-element type discretizations on unstructured meshes. This solver has been developed at the US Army High Performance Computing Research Center (AHPCRC) at Stanford University. It operates on unstructured body-fitted meshes, or on fixed meshes that can embed discrete representations of surfaces of obstacles around or within the flow to be computed. The spatial discretization combines a second-order-accurate upwind scheme for the advective fluxes and a Galerkin-centered approximation for the viscous fluxes. Time-integration can be performed with first- and second-order implicit, and first-, second-, and fourth-order explicit algorithm which, when performing in the ALE setting, satisfy their discrete geometric conservation laws (DGCLs) (AERO-F Manual (2011)). CFD++ is a commercial software package that can be used for steady/unsteady, compressible/incompressible, Euler/Navier-Stokes and laminar/turbulent flow problems. It is a finite-volume-based solver with spatial 2nd-order and temporal 2nd-order-implicit/explicit descretization.

The following flow conditions are used for the AERO-F solver: M=0, T=300°K, P=1.05×10^5Pa, ρ=50,000kg/m^3, and v=0 (no-slip boundary condition in the case of viscous flow simulation) on the surface of the wing, where M, T, P, and v are the Mach number, temperature, pressure, and velocity of the flow. Since the flow condition is in the incompressible range, a pre-conditioner Mach number of 0.05 is used for all simulations to accelerate the convergence of the system of equations. An incompressible flow solver algorithm is used for CFD++. The boundary conditions for the CFD++ solver consist of

pressure and velocity at the far-field and no-slip on the wing surface. For both solvers, the same unstructured mesh with 3.2 million tetrahedral elements is used and the entire fluid mesh is moved with the wing. The size of each time-step is 2×10^{-4} seconds for the AERO-F solver, whereas a bigger time-step size of 2×10^{-3} seconds is applied for the CFD++ solver.

3. Experimental Setup

Figure 2 depicts the experimental setup. The facility consists of a tank which holds 70 gallons when filled to capacity. The tank is 0.76 m in height and has a regular hexagonal cross-section with rounded corners with an area of 0.35 m^2. The testing medium is mineral oil which has a density of 860 kg/m^3 and a kinematic viscosity of 0.979 centistokes at the testing temperature of 70°F. Two motors drive a coaxial transmission which controls the wing flap and pitch angles. The flap angle is prescribed by the angular orientation of the inner-shaft while the angular difference between the inner- and outer-shafts controls the pitch. For a fixed pitch angle, both shafts rotate at a common angular velocity. There is a rigid connection with a 6:1 step down between the stepper motor angular inputs and the shaft outputs. The motors are controlled open loop with the actual position tracked by encoders on the motor shafts. The wing model extends from a custom 5-component sting-style balance (Gerdes, 2011). The models were fabricated of an Acrylonitrile butadiene styrene (ABS) & Polycarbonates (PC) blended plastic through a layer deposition processes and conform to the dimensions in Figure 1.

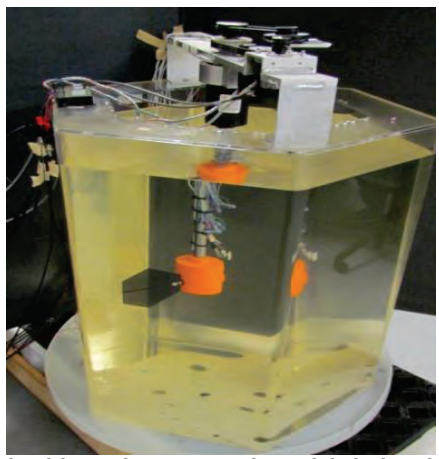


Figure 2. Experimental set-up depicting the drive train on top and coaxial shafts with a Comb wing mounted on the balance

The testing procedure requires the measure of the balance voltage prior to executing the wing motion to be applied as a zero reference. The prescribed motions are executed and voltages are sampled at a rate of 100 Hz. The period of wing motion in all cases is 0.3 Hz and the data is collected for 5 strokes. The change in voltages is converted to changes in force applying a previously determined calibration.

The resulting forces are filtered at 3 Hz to eliminate the effects of some higher harmonic structural motion resulting from gear teeth meshing and an impulsive wobble in the drive shafts at each flap reversal.

4. Results and Discussion

This section consists of three parts. First, the computational results of the compressible flow solver (AERO-F) and incompressible flow solver (CFD++) for the flapping motion are compared at different flow conditions. Then, the computational results at the flow condition of interest are validated against the experimental data. Detailed analysis of the computational results for the flapping motion is presented in the third part.

The frequency of the motion is 0.3 Hz for all flow conditions. The Reynolds number ($Re=(\rho V L_c)/\mu$ in this study is based on the maximum chord length (L_c) of 0.113m shown in Figure 1 and the time-averaged velocity (V) of 0.043m/sec. Also of note for all graphs shown in this paper, the forces obtained from the AERO-F solver are scaled by the density ratio of ρ_1/ρ_2=58.13, where ρ_1=50,000kg/m^3 used in AERO-F to accelerate convergence, and ρ_2=860kg/m^3 is the actual density of the fluid which is applied in CFD++. The wing is assumed to be rigid.

Figure 3 shows the time-history of aerodynamic forces on both wings along the *x* and *y* directions in a fixed frame of reference obtained from the two CFD solvers. The three flow conditions consist of an inviscid flow and two different Reynolds numbers of 18.26 and 250, where the viscosity of the fluid is the only parameter that differs in these three cases. Time-accurate simulations were started from the initial location of wing span along the *x*-axis and wing surface normal to the *y*-direction. The wing initially moves in the positive *y*-direction. Initial transient forces on the wings were observed to reach statistically stationary within 1 sec of simulation time. For the wing flapping at a frequency of 0.3 Hz (time period of oscillation of 3.33 sec), this transient state corresponds to wing motion of little over a quarter of the period.

As seen in Figure 3(a), there is a very close agreement between the results of the two solvers for the Comb wing in the inviscid flow condition. At the medium viscosity (*Re*=250), the results of the both solvers match with each other reasonably, whereas at the highest viscosity (*Re*=18.26) of the range of study, 25% and 12% discrepancy in the peak magnitude of the *x*-force and *y*-force is observed, respectively, between the two simulations.

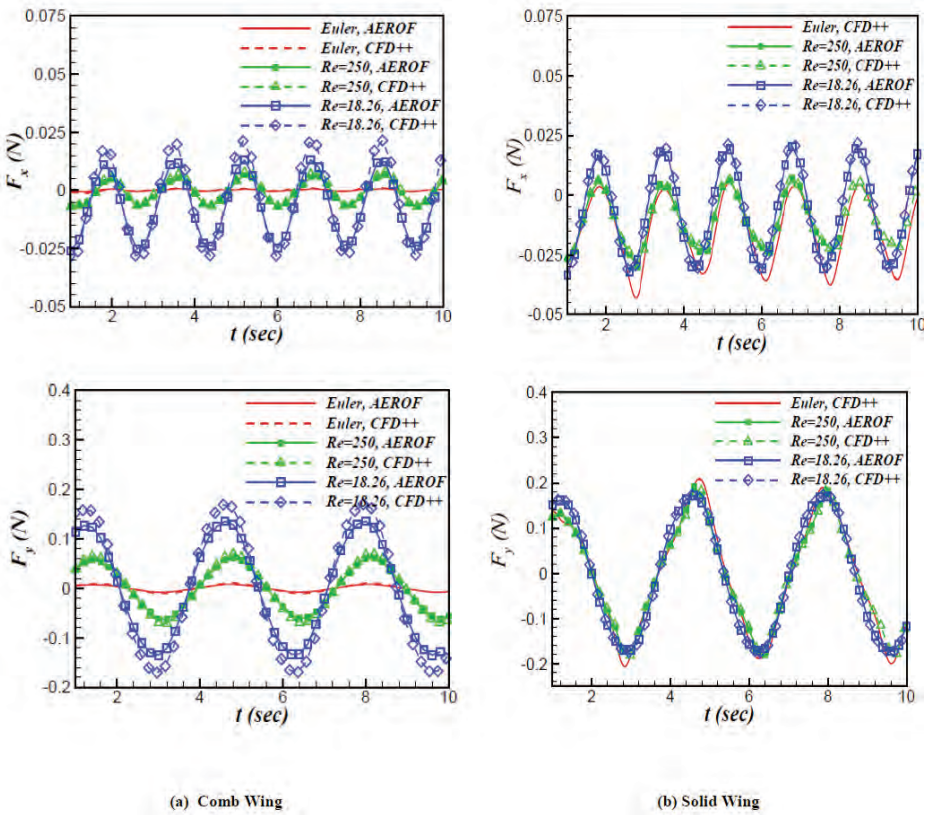


Figure 3. Comparison between the results of AERO-F and CFD++ solvers at different flow conditions

In the case of the Solid wing shown in Figure 3(b), the results of both solvers for two different Reynolds number are in good agreement. It is also observed that while increasing the viscosity of the fluid augments the *x*- and *y*-force for the Comb wing dramatically, it only changes the *x*-force on the Solid wing slightly and does not cause any noticeable effect on y-force of this wing. This observation will be discussed in detail in the third part of this section. It is also seen that the frequency of the *x*-force is almost twice of the frequency of *y*-force, and the time average of the *x*-force is non-zero for both wings. These observations will also be discussed in details later.

The aerodynamic force along the *y*-axis obtained from the numerical computations and experiment is compared in Figure 4 for the flapping motion of both wings at the lowest Reynolds number of the study (18.26). As mentioned earlier, 3 Hz filter was applied to experimental data. Wing was fairly stiff and no noticeable deflection was observed in the experiments. Reasonable agreement is seen between the three sets of data during one cycle with a slight difference in the peak magnitude of the force.

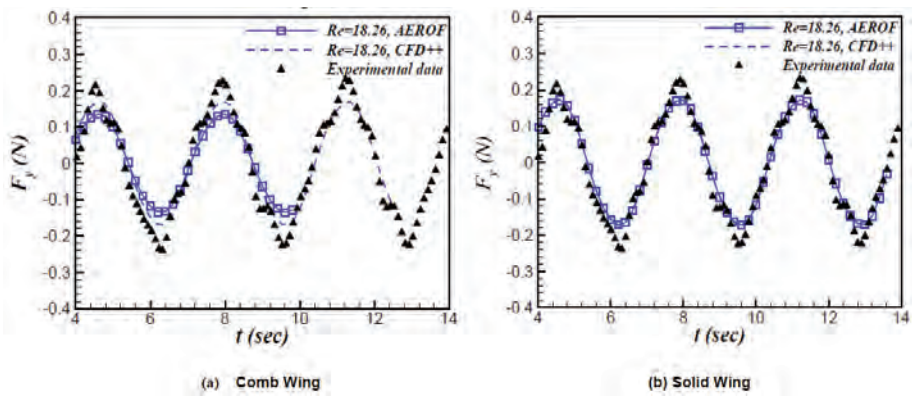


Figure 4. Comparison between the computational results and experimental data at *Re*=18.26

The experiment also shows agreement between the solid and comb wing result as similar force measurements are observed. Although, the pressure and viscous forces cannot be measured separately in the experiment, it is clear that, at these low-Reynolds numbers, creeping flow is resulting in an increased viscous force such that the two wings experience the same overall aerodynamic forces.

The *y*-force from the computations on the two wings at different Reynolds numbers is compared in Figure 5. For the inviscid flow condition, the *y*-force on the Comb wing is negligible in comparison with the Solid wing. However, interestingly at the highest viscosity (lowest *Re*) of the study, the total *y*-force on the Comb wing is similar to the Solid wing. In order to obtain more insight into the flow physics of the problem, the pressure and viscous forces of both wings are extracted and compared at Reynolds numbers of 18.26 and 250. As seen from Figure 6, pressure force is dominant for the Solid wing at both the Reynolds numbers and changing the viscosity of the fluid does not have a significant effect on the total force. In the case of the Comb wing, since the projected area for this wing is smaller than the Solid wing, the pressure force is small in at both the Reynolds numbers. At higher viscosity the contribution due to viscous force increases to the same level as the pressure force on the Solid wing at *Re*=18.26.

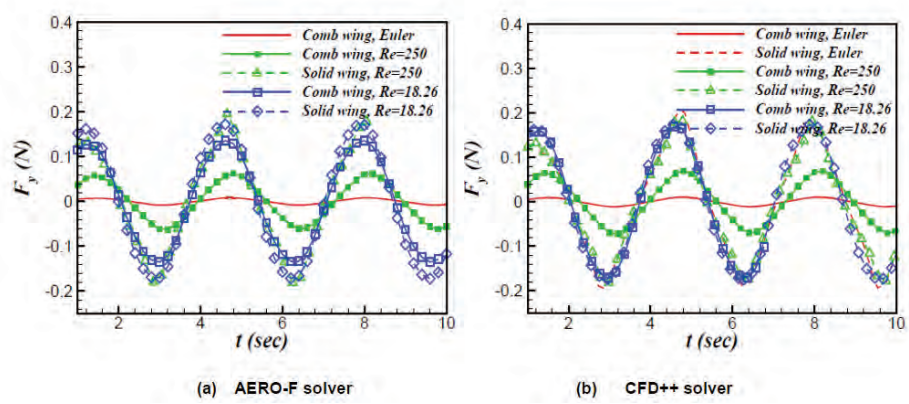


Figure 5. Comparison of *y*-force at different Reynolds numbers

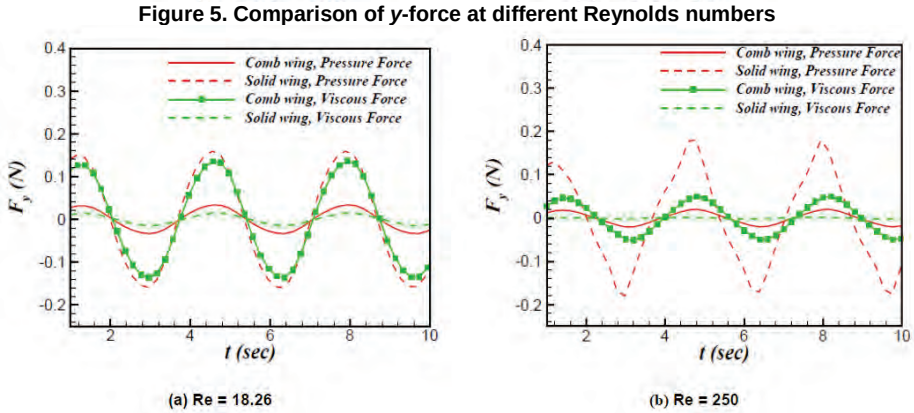


Figure 6. Comparison between the Comb and Solid wings for the pressure and viscous forces

In order to gain further insight into the flow physics of the flapping motion, the pressure contours on the surface of the wings are examined at different time instances given in Figure 7 and 8 for the Solid and Comb wings, respectively. As mentioned before, the flapping angle varies sinusoidally with time (see Figure 9). From 0 to $T/4$ of the cycle, the wing rotates from the center (θ=0°) to one end (θ=15°) of the rotation and during this part of the cycle, the angular velocity which is in the positive direction of the z-axis decreases from its maximum to zero (deceleration interval). As a result, pressure on the top side (first row in Figure 7 and Figure 8) of the wing is higher than the bottom side (second row of Figures 7 and 8) at $T/4$. Note that the top side corresponds to the surface of the wing with its outward normal in the negative direction of y axis at θ=0° and bottom side represents the surface of the wing with an outward normal along the positive y axis at θ=0°. From $T/4$ to $T/2$, the wing rotates in the opposite direction toward the center (From θ=15° to θ=0°) and the angular velocity increases (acceleration interval). Therefore the fluid pressure on the top side of the wing is still higher than the bottom side. During the third quarter of the cycle (from $T/2$ to $3T/4$), deceleration of the wing motion occurs and fluid pressure starts to increase on the bottom side and decrease on the top side and somewhere between θ=0o and θ=-15o pressure on the bottom surface becomes higher than the top surface of the wing. That is one of the main reasons for observing the phase difference between the wing motion and pressure variation in Figure 9. During the final quarter of the cycle (from $3T/4$ to T), the wing rotates from the other end toward the center of rotation (θ=−15° to 0°) and acceleration of the wing causes the fluid pressure on the bottom side of the wing to be still higher than the top side.

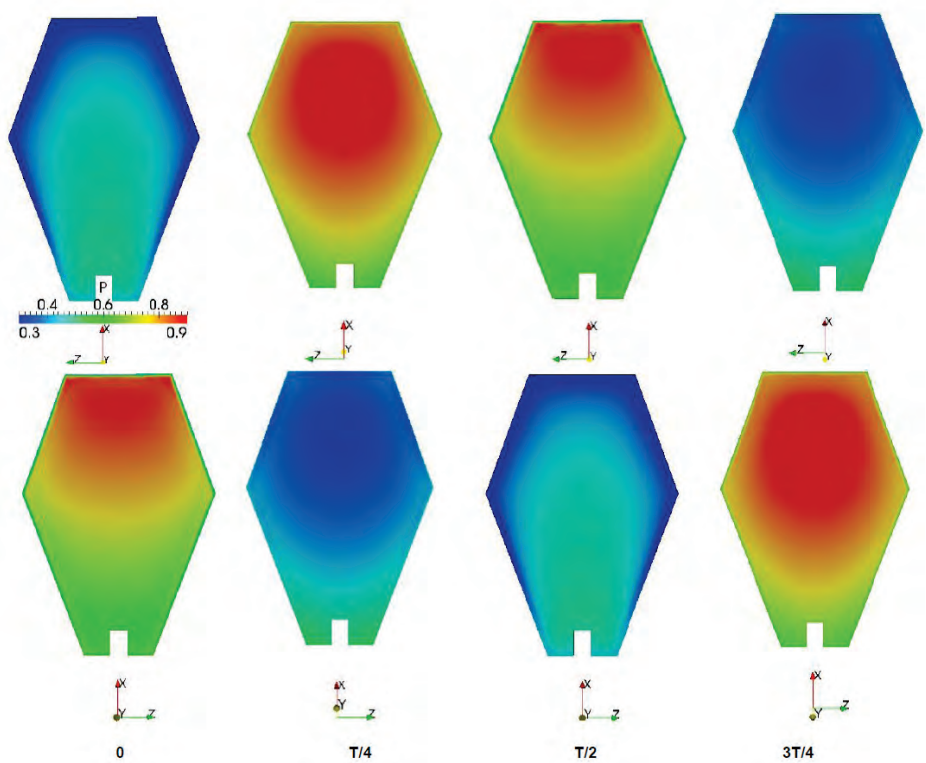


Figure 7. Contours of fluid pressure on the surface of Solid wing at four different time instances. Top row is negative surface normal side; bottom row is positive surface normal side.

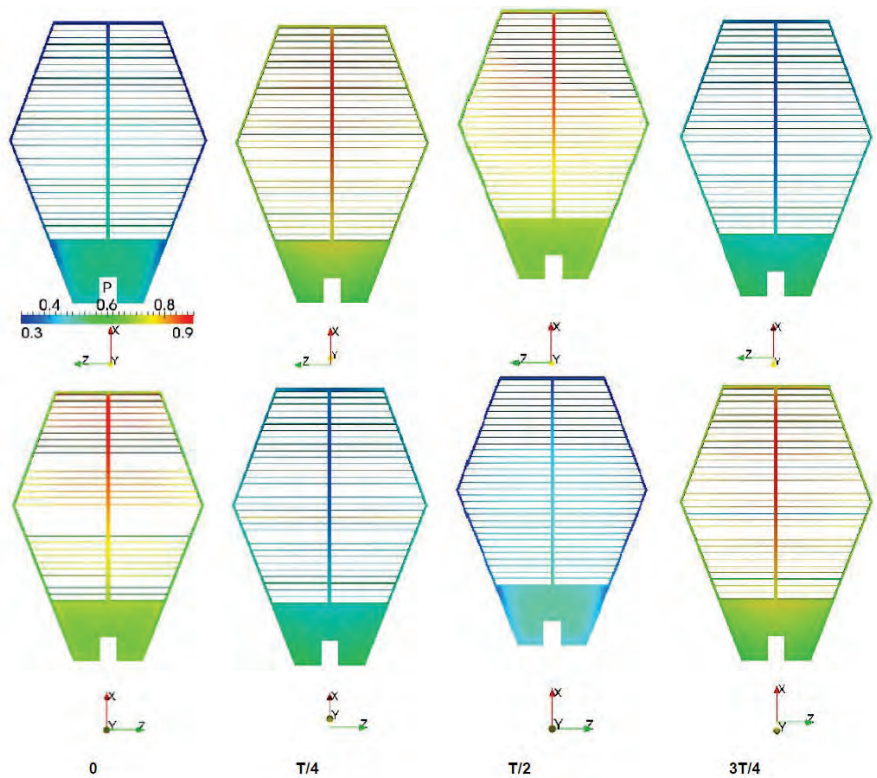


Figure 8. Contours of fluid pressure on the surface of Comb wing at four different time instances. Top row is negative surface normal side; bottom row is positive surface normal side.

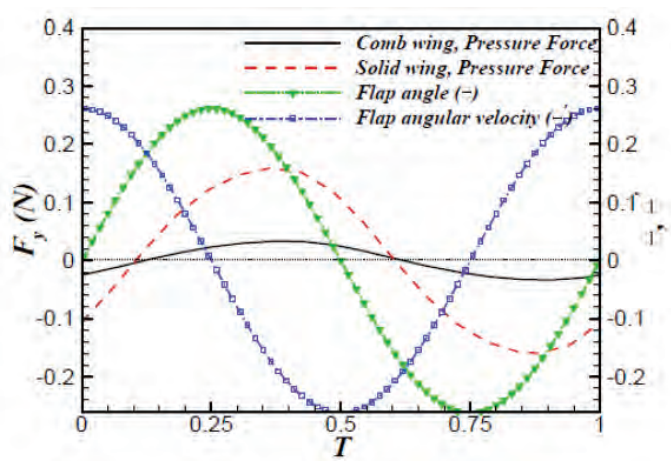


Figure 9. Variation of angle of flapping, derivative of the angle of flapping and pressure force on the wing surface along the *y*-direction over one cycle of motion

Comparing Figures 7 and 8 shows that the pressure contours near the root of both wings are qualitatively similar and as seen in Figure 9 due to the fact that the effective area for the Solid wing is bigger than the Comb wing, the integral pressure force is higher on the former wing.

5. Significance to DoD

Over the past decade, Micro Aerial Vehicles (MAV) has attracted much interest due to their wide range of civil and military applications including intelligence, surveillance and reconnaissance missions. Flapping of the wings is a commonly used mechanism to generate forces required to fly such vehicles. The US Army Research Laboratory has several

ongoing research initiatives towards developing efficient flapping wing systems. The light-weight and small-size of these vehicles increase the physical and technical challenges in modeling these vehicles. From the aerodynamic view point, this small-scale, low-flight-speed condition leads to low Reynolds number flow (on the 10^2).

Most of the past numerical and experimental studies regarding flapping wings have focused on solid (non-perforated) wings, in which the effect of oscillation mode, wing aspect ratio, and type of motion have been investigated. Perforated wings have shown evidence of similar aerodynamic performance in experimental measurements at low-Reynolds number flows. Such wings have the advantage of reduced weight compared with the solid wing of same plan-form and a means of spatially tailoring the elastic properties of the wings. These reasons motivated us to conduct numerical simulations and companion experimental study of the Solid and Comb wings. Insights gained from computational simulations provide valuable guidance in choosing configurations and operational parameters for designing and manufacturing mechanical wings. The numerical study of this problem essentially involves the computational fluid dynamics simulations to capture the flow physics associated with the motion of the wing. These simulations are numerically-expensive and highly time-consuming, and need to be performed on high performance computers. Current simulations are run on Harold (one of ARL DSRC Facilities) with the following specifications: Intel Xeon quad-core Nehalem with 2.8GHZ core speed, 24Gbytes memory/node shared memory on nodes, and total compute nodes of 1,344. The number of compute nodes used for simulations are 32 and 128 for the CFD++ and AERO-F solvers, respectively. The CPU time required for marching in time over one cycle of the flapping motion of the Solid wing is 534 hours for CFD++ and 1,583 hours for AERO-F.

6. Conclusion

Numerical simulations and experimental measurements were conducted to compare the performance of Comb and Solid wings for micro-aerial vehicles application at ultra-low Reynolds number flow condition. Numerical study was performed for flapping motion of both the wings with maximum flapping angle of 15° using two different flow solvers including AEROF and CFD++ at three different flow conditions (Euler flow and two different Reynolds numbers of 18.26 and 250 based on the maximum chord length). Following conclusions can be drawn from the study:

- Good agreement is seen between the results of both solvers at different flow conditions.
- Reasonable agreement between the computational results and experimental data is seen during a cycle of motion with a slight difference in the peak magnitude of the *y*-force.
- Force magnitude for the Solid wing is almost the same as the force magnitude on the Comb wing at the lowest Reynolds number. The forces on the Comb wing decrease with the Reynolds number. For Euler simulations (infinite Reynolds number), forces on the Comb wing are significantly smaller than the Solid wing.
- A detailed analysis of the computational results shows that the pressure force is dominant for the Solid wing, whereas the viscous force provides more contribution to the total force along the *y* direction than the pressure force in the case of the Comb wing. Interestingly the total force in all directions produced by the Comb wing is very close to those of the Solid wing at the highest viscosity of the study. Slight phase difference is observed between the wing motion and pressure force variation.

Future work includes increasing the maximum angle of flapping as well as the flapping frequency, examining the perforated wing with a different slit size, performing a scaling analysis of the wings at different Reynolds number based on the length scale, velocity scale, and fluid properties, and studying the combination of flapping and pitching motions of the wings at different operational and flow conditions.

Acknowledgments

The authors gratefully acknowledge Kevin Guanyuan Wang, postdoctoral fellow at Dr. Farhat Charbel's Research Group at Stanford University for his support and advice on the AEROF solver. The authors would like to thank John Gerdes of ARL/VTD and Howard Carpenter of Bowhead Science and Technology for their efforts in the design and fabrication of the oil tank and balance. We would also like to acknowledge a prior collaboration with Dr. Daniel Weihs of the Technion University with whom the study of the comb mechanism was initially considered. This collaboration was conducted under the US-Israel Memorandum of Agreement on Rotorcraft Aeromechanics and Man-Machine Integration Technologies, TPOC Mark Tischler AMRDEC/AFDD.

References

AEROF Manual, Farhat Research Group (FRG), Stanford University, Edition 0.1/13, July 2011.

Bronson, J.R., Pulskamp, J.S., Polcawich R.G., Kroninger, C.M., and Wetzel E.D., "PZT MEMS Actuated Flapping Wings for Insect-Inspired Robotics." *IEEE Conf. MEMS*, Sorranto, Italy, January 2009.

Gerdes, John, "Design of a Five-Axis Load Cell for Submerged Wing Testing in an Oil Tank." *ARL Technical Report 5843*, December 2011.

Kuethe, A.M., *On the Mechanics of Flight Insects, Swimming and Flying in Nature*, edited by T.Y. Wu, C.J. Browkaw, and C. Brennen, Vol. 2, Plenum, New York, pp. 803–813, 1975.

Rozen, Ofer, Tali Nachmias, Shmuel Kaldor, Irit Peleg Gabriel L. Smith, Christopher M. Kroninger, Jeffrey S. Pulskamp, Ronald G. Polcawich, and Daniel Weihs. "A PZT Actuated Millimeter Scale Flapping-Wing." *51st Israel Annual Conference on Aerospace Sciences*, Haifa, IL, February 23–24, 2011.

Pulskamp, J., Smith, G., Polcawich, R., Kroninger, C., and Wetzel, E., "Two Degree of Freedom PZT MEMS Actuated Flapping Wings with Integrated Force Sensing." *Proceedings of Hilton Head Workshop 2010: A Solid-State Sensors, Actuators and Microsystems Workshop*, Hilton Head, SC, June 2010.

Zussman, E., Yarin, A., and Weihs, D., "A Micro-Aerodynamic Decelorator Based on Permeable Surfaces of Nanofiber Mats." *Experiments in Fluids*, Vol. 33, No. 2, pp. 315–320, 2002.

First-Principles Characterization of Armor Ceramics

DeCarlos E. Taylor and Iskander Batyrev
US Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
{decarlos.e.taylor.civ, iskander.g.batyrev.civ}@mail.mil

Abstract

The effects of stoichiometry and atomic structure on the mechanical properties of the boron carbide armor ceramic have been studied using density functional theory. Computational cells spanning compositions ranging from 6.7% to 26.7% carbon were optimized quantum mechanically, and the full elastic constant tensor for each structure was determined. It was found that changes in stoichiometry, as well as variable atomic arrangements within a fixed stoichiometry, can have significant impact on the elastic response of the ceramic.

1. Introduction

Boron Carbide (BC), due to its extreme hardness and low-density, has been used as an armor ceramic for many years.[1] The elastic properties suggest that BC should sustain high-impact pressures without failure; however, there is an anomalous loss of impact resistance within the material, apparently due to increased fragmentation, by a factor of two to three. Recent experimental work within the Army Research Laboratory has identified the formation of nanoscale-sized intra-granular amorphous bands as the primary damage mechanism leading to the marked reduction in performance of BC.[2] The Army has a need to understand the mechanism for the formation of these amorphous bands, ***at the atomic level***, which will allow for the design of chemically-modified BC materials that will show improved impact resistance and better performance as a protection material for the Warfighter. Under the auspices of this High Performance Computing and Modernization Program (HPCMP) Challenge Project (C5L), first-principles quantum mechanics has been used to determine the elastic properties of BC, as a function of stoichiometry, which may yield insight on the origins of the deformation and damage mechanisms leading to the formation of the amorphous bands which weaken the BC structure after impact.

2. Boron Carbide

Boron carbide, with nominal stoichiometry B_4C, consists of 12-atom icosahedra cross linked by 3-atom chains (Figure 1-A) and belongs to the crystallographic space group $\bar{R}3m$.[3] The exact nature of the 3-atom chain is unknown experimentally due to the close spectroscopic signatures of boron and carbon and the crystal structure consists of a mixture of B_{12} icosahedra linked by C-C-C chains and $B_{11}C$ icosahedra linked with C-B-C chains (among others). Within each icosahedron, there exist two crystallographically-unique sites termed "polar" and "equatorial" as shown in Figure 1-B, and hereafter, any atom that specifically occupies a polar or equatorial site within an icosahedron will be labeled with a subscript "p" or "e" respectively. Although BC is generally regarded to have $\bar{R}3m$ symmetry, which requires an equal length for each lattice vector and angle, this is only true for the B_{12}(CCC) structure and placement of a carbon atom in either the polar or equatorial icosahedral position results in a monoclinic distortion of the unit cell, which lowers the symmetry of the lattice. Experimentally[3], only a single vector length/angle is reported; therefore, the experimental values must consist of an average over multiple structures since a single-lattice vector length and cell angle cannot possibly encompass all possible configurations.

BC can be produced by several methods such as reaction of boric oxide and carbon in an electric arc furnace,[4] or carbothermal reduction of a boric acid-citric acid gel.[5] The powders consist of a range of boron to carbon ratios and this variation in composition has an effect on the observed mechanical properties, as shown in Table 1, where the experimentally measured bulk and shear moduli for samples ranging from B_4C (20% carbon) to $B_{7.7}C$ (11.5% carbon) are presented.[6] The general trend evidenced by the experimental data is a reduction in stiffness as the boron concentration increases.

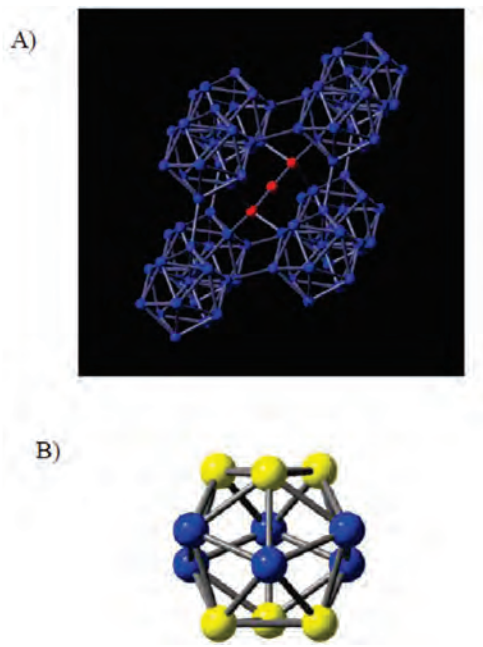


Figure 1. (A) Icosahedral structure of boron carbide. Boron=blue, Carbon=red. For clarity, only a single chain is shown in the image. (B) Icosahedron with polar (yellow) and equatorial (blue) positions.

Table 1. Experimental bulk and shear moduli (GPa) of boron carbide as a function of composition

Stoichiometry	% Carbon	Bulk Modulus	Shear Modulus
B_4C	20.0	235	197
$B_{4.5}C$	18.2	237	197
$B_{5.6}C$	15.2	236	197
$B_{6.5}C$	13.3	231	189
$B_{7.7}C$	11.5	178	150

Although the structure and formation enthalpy of B_4C as a function of stoichiometry have been well-documented, knowledge of the origins of the deformation and damage mechanisms, and their relation to the macro-mechanical properties, is critical. A first step in this analysis is characterization of the elastic properties of BC for the range of carbon compositions found in the material. This information can play a role in identification of "soft" configurations that may initiate failure in the BC structure when impacted with high-velocity projectiles. In this paper, we present a survey, using quantum mechanics and HPCMP computational resources, of the mechanical properties of 15 BC structures (listed in Table 2) with stoichiometries ranging from $B_{2.75}C$ to $B_{14}C$.

Table 2. Boron carbide stoichiometries used in this study. The subscript e and p labels denote equatorial and polar carbons, respectively.

Structure	Stoichiometry	%C
$B_{11}C_e$(CCC)	$B_{2.75}C$	26.66
$B_{11}C_p$(CCC)	$B_{2.75}C$	26.66
B_{12}(CCC)	B_4C	20.00
$B_{11}C_e$(CCB)	B_4C	20.00
$B_{11}C_p$(CCB)	B_4C	20.00
$B_{11}C_e$(CBC)	B_4C	20.00
$B_{11}C_p$(CBC)	B_4C	20.00
B_{12}(CCB)	$B_{6.5}C$	13.33
B_{12}(CBC)	$B_{6.5}C$	13.33
$B_{11}C_e$(BCB)	$B_{6.5}C$	13.33
$B_{11}C_p$(BCB)	$B_{6.5}C$	13.33
$B_{11}C_e$(BBC)	$B_{6.5}C$	13.33
$B_{11}C_p$(BBC)	$B_{6.5}C$	13.33
B_{12}(BBC)	$B_{14}C$	6.66
B_{12}(BCB)	$B_{14}C$	6.66

3. Numerical Methodology

3.1 Crystal Structure Optimization Algorithm

A crystal structure optimization program was written, based on the L-BFGS[7] algorithm, where any stress-state (hydrostatic, uniaxial, shear) can be imposed. Several software packages (such as CP2K[8]) already offer such implementations; however, many of those programs rotate the input coordinates to a different computational orientation making interpretation of the stress along a specific crystallographic direction, a key component of this work, more difficult to monitor and interpret. To circumvent this difficulty, a program that does not perform a rotation of the input orientation was written and used in this study. At each optimization step, the energy, forces, and stress tensor were evaluated using the Perdew-Burke-Ernzerhof[9] (PBE) functional in a double-zeta valence plus polarization basis set with a plane-wave cutoff of 800 Rydberg provided by the CP2K program. It should be noted that no symmetry restrictions were imposed, i.e., no constraints were applied to enforce linearity of the 3-atom chain. For each system in Table 2, a 2×2×2 computational cell containing 120 atoms was constructed, and at each optimization step the stress tensor returned by CP2K was converted to cell vector derivatives using the transformation from stress to cell vector gradients given by Doll[10]. Optimization was considered converged when the gradient norm of the cell vector derivatives was below 0.0001 atomic units.

3.2 Elastic Constants

Elastic constants are related to the second derivative of the total energy with respect to strain, ε_i, via

$$\mathrm{C}_{ij} = \frac{1}{\mathrm{V}} \frac{\partial^2 E}{\partial \varepsilon_i \partial \varepsilon_j}\bigg|_0 \tag{1}$$

where V is the unit cell volume and i,j=1…6 using the compact Voigt notation (1=xx, 2=yy, 3=zz, 4=yz, 5=xz, 6=xy). For this work, a program was written that evaluates the second derivatives in equation 1 via a finite-difference of analytic first-derivatives of the energy with respect to strain (stress tensor) provided by the CP2K code. We have monitored the change in the elastic constants as a function of hydrostatic and uniaxial load and the required stress corrections[11] for elastic constants, C_{ijkl}, under non-zero load were included using:

$$B_{ijkl} = C_{ijkl} + \frac{1}{2}\left(\delta_{ik}\sigma_{jl} + \delta_{jk}\sigma_{il} + \delta_{il}\sigma_{jk} + \delta_{jl}\sigma_{ik} - 2\delta_{kl}\sigma_{ij}\right) \tag{2}$$

with σ_{ij} being an element of the stress tensor and B_{ijkl} representing the stress corrected effective elastic constant.

4. Results

4.1 Structures

The lattice parameters for each structure resulting from optimization at 0K are presented in Table 3. The distortion of the lattice from purely rhombohedral symmetry is clearly evident in many of the systems, and within each carbon concentration the *a* lattice vector shows a larger contraction when the carbon atom resides in the polar site. It is noteworthy that many of the low-carbon content stoichiometries produced structures that were either unstable elastically (negative eigenvalue in the elastic constant tensor), or converged to a minimum energy structure with a non-linear 3-atom chain. It is known experimentally[12] that at an approximately 8% carbon content, boron begins to precipitate from the lattice-yielding mixtures of boron carbide and pure boron. The instability of the linear 3-atom chain structure for the many of the boron-rich compositions found in this work supports this finding.

4.1 Elastic Constants

The zero-stress elastic constants (with respect to rhombohedral axes), bulk modulus, shear modulus, and Young's modulus for each structure are presented in Table 4. It should be noted that only the B_{12}(CCC) and B_{12}(CBC) structures strictly adhere to $\bar{\mathrm{R}}3\mathrm{m}$ symmetry, and the small distortions present in the other structures introduce additional non-zero, albeit small, elements in the elastic constant tensor. As a result, for ease of comparison across the range of structures, only

the six non-zero Cij's that are present for $\bar{R}3m$ symmetry are presented. The elastic constants are similar in magnitude for all structures however B_{12}(CCB) shows a considerable reduction in stiffness compared to the other systems. The experimental values in Table 4 were measured by McClellan, et al.[13] using a sample with stoichiometry $B_{5.6}C$ (15.2%C). However, in their work, the value of the C_{14} modulus was indeterminate due to the hexagonal orientation assumed in determination of the elastic moduli. The theoretical C_{14} moduli are comparatively small and that for the B_{12}(CBC) structure is negative compared to the positive values obtained for the rest of the systems. The negative C_{14} value for B_{12}(CBC) is consistent with the result reported by Shirai.[14]

Table 3. Computed unit cell parameters. Entries with grey cells indicate that no stable configuration containing a *linear* 3-atom chain was found. (Lengths in Angstroms, angles in degrees, volume in cubic Angstroms).

Structure	Formula	%C	*a*	*b*	*c*	*α*	*β*	*γ*	Volume
Experiment[13]	$B_{5.6}C$	15.2	5.19	5.19	5.19	65.18	65.18	65.18	110.02
$B_{11}C_e$(CCC)	$B_{2.75}C$	26.66	**5.14**	**5.21**	**5.21**	**64.31**	**65.19**	**65.19**	**109.28**
$B_{11}C_p$(CCC)	$B_{2.75}C$	26.66	**5.05**	**5.21**	**5.21**	**64.86**	**66.05**	**66.05**	**108.93**
B_{12}(CCC)	B_4C	20.00	**5.19**	**5.19**	**5.19**	**66.01**	**66.01**	**66.01**	**112.09**
$B_{11}C_e$(CCB)	B_4C	20.00	**5.16**	**5.22**	**5.22**	**66.18**	**66.16**	**66.16**	**112.72**
$B_{11}C_p$(CCB)	B_4C	20.00	**5.05**	**5.23**	**5.23**	**66.17**	**67.33**	**67.33**	**112.27**
$B_{11}C_e$(CBC)	B_4C	20.00	**5.18**	**5.22**	**5.22**	**64.86**	**64.97**	**64.97**	**110.16**
$B_{11}C_p$(CBC)	B_4C	20.00	**5.07**	**5.22**	**5.22**	**65.24**	**66.07**	**66.07**	**109.75**
B_{12}(CCB)	$B_{6.5}C$	13.33	**5.16**	**5.22**	**5.22**	**66.79**	**67.89**	**67.89**	**115.29**
B_{12}(CBC)	$B_{6.5}C$	13.33	**5.20**	**5.20**	**5.20**	**65.83**	**65.83**	**65.83**	**112.10**
$B_{11}C_e$(BCB)	$B_{6.5}C$	13.33							
$B_{11}C_p$(BCB)	$B_{6.5}C$	13.33							
$B_{11}C_e$(BBC)	$B_{6.5}C$	13.33							
$B_{11}C_p$(BBC)	$B_{6.5}C$	13.33							
B_{12}(BBC)	$B_{14}C$	6.66							
B_{12}(BCB)	$B_{14}C$	6.66							

Table 4. Computed zero pressure elastic moduli. All values in GPa. The three values for Young's modulus correspond to the values along the *x*, *y*, and *z* stress axes where the z-axis is coincident with the rhombohedrally-oriented [111] direction.

Structure	Formula	%C	C_{11}	C_{12}	C_{13}	C_{14}	C_{33}	C_{44}	Bulk[a]	Shear	Young's
Experiment[13]	**B5.6C**	**15.2**	**542.8**	**130.6**	**63.5**	**—**	**534.5**	**164.8**	**236.8**	**195.6**	**460.1**
$B_{11}C_e$(CCC)	B2.75C	26.66	501.7	120.0	65.8	26.7	547.7	190.7	235.4	195.4	467/513/528
$B_{11}C_p$(CCC)	B2.75C	26.66	517.9	133.1	58.7	49.1	544.4	173.4	237.4	189.9	464/498/531
B_{12}(CCC)	B4C	20.00	486.9	188.8	64.9	14.7	518.1	133.6	221.1	173.2	451/451/504
$B_{11}C_e$(CCB)	B4C	20.00	457.8	119.8	61.5	36.3	536.1	111.6	221.8	162.4	409/438/520
$B_{11}C_p$(CCB)	B4C	20.00	470.3	124.6	53.6	44.6	505.7	132.5	217.8	170.9	417/441/492
$B_{11}C_e$(CBC)	B4C	20.00	518.3	116.8	65.9	30.6	522.5	159.6	234.1	196.5	481/522/507
$B_{11}C_p$(CBC)	B4C	20.00	534.2	120.2	58.1	38.0	525.7	168.4	233.6	199.7	494/519/514
B_{12}(CCB)	B6.5C	13.33	395.7	139.4	82.8	61.7	393.7	96.6	202.3	135.6	291/381/373
B_{12}(CBC)	B6.5C	13.33	531.3	105.3	54.2	–7.95	528.5	167.1	224.4	201.4	506/506/519

[a] Computed using Voigt-Reuss-Hill average.

5. Conclusion and Future Work

The calculations reported here have outlined the elastic properties of BC as a function of structure and stoichiometry. A change in carbon content can have a dramatic effect on the elastic response as evidenced by the significantly-reduced stiffness found in the B_{12}(CCB) structure containing 13.33% carbon. Based on these values, B_{12}(CCB) may be a point of initial failure in the BC crystal structure when impacted with a high-velocity projectile; however, further computational exploration is necessary before a definitive conclusion can be made. The overall goal of this Challenge Project is the characterization of the mechanism leading to structural failure of BC under impact and this can be readily studied using molecular dynamics (MD) simulation. However, MD is a much more time-consuming methodology requiring repeated evaluation of the energy and forces during the trajectory. We have initiated several of these simulations for a range of

pressures using density functional theory to provide the required atomic forces. Results of the MD simulations will be presented in a future report.

In addition to the work on boron carbide, we are also modeling the transparent armor ceramic aluminum oxynitride (AlON). AlON has unique hardness and strength characteristics that make it a potentially useful material for windows in military vehicles; however, its failure under high stress and strain rates is not fully understood. ARL has made significant investments in the identification of the failure mechanisms of this material with a goal to implement mitigating strategies that will enable its fielding into Army platforms. AlON has an unusual stoichiometry, $Al_{23}N_5O_{27}$, manifested in a cubic spinel structure where the oxygen/nitrogen atoms of the lattice form a face-centered cubic structure with aluminum atoms occupying tetragonal and octahedral interstitial sites, with one interstitial site remaining vacant. Unfortunately, the location of the nitrogen atoms and the interstitial vacancy in the unit-cell is not unique thus complicating a definitive characterization of the AlON structure. In order to identify possible structures, we are using a genetic algorithm to identify low-energy configurations with a varying distribution of atoms and vacancies. A low-energy structure identified using the genetic algorithm is shown in Figure 2.

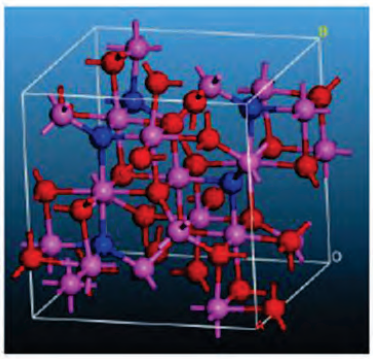


Figure 2. A low-energy structure found using genetic algorithm (Oxygen=red, Nitrogen=blue, Aluminum=pink)

Using this cell, elastic constants were calculated using density functional theory. The values of the non-zero elastic constants (C11=312.35, C12=155.05, and C44=169.01 GPa) are in good agreement with experimental[15] results. Future work will entail identification of additional AlON structures for comparison to experiment.

Acknowledgments

This work has been supported by the HPCMP through provision of a 12.5 million CPU-hour Challenge Project with time allocated at the US Army Research Laboratory, the US Air Force Research Laboratory, and the US Army Engineer Research and Development Center supercomputing centers.

References

1. Telle, R., *Structure and Properties of Ceramics*, ed. by M.V. Swain, VCH Publishers, Weinheim, 1994.

2. Chen, M., J.W. McCauley, and K.J. Hemke, "Shock-Induced Localized Amorphization in Boron Carbide." *Science*, 299, 1563, 2003.

3. Clark, H.K. and J.L. Hoard, *J. Am. Cer. Soc.*, 65, 2115, 1943.

4. Wilson, W.S. and P.J. Guichelaar, *Carbide, Nitride, and Boride Materials Synthesis and Processing*, ed. by A.W. Weimer, Chapman and Hall, London, 1997.

5. Khanra, A.K., *Bull. Mater. Sci.*, 30, 93, 2007.

6. Domnich, V., S. Reynaud, R.A. Haber, and M. Chhowalla, *J. Am. Ceram. Soc.*, 94, 3605, 2011 and references therein.

7. Liu, D.C. and J. Nocedal, *Math. Program.*, 45, 503, 1989.

8. CP2K is freely available from http://cp2k.berlios.de/.

9. Perdew, J.P., K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.*, 77, 3865, 1996.

10. Doll, K., *Mol. Phys.*, 108, 223, 2009.

11. Wang and S. Yip, *Phys. Rev. Lett.*, 71, 4182, 1993.

12. Bouchacourt, M. and F. Thevenot, *J. Less-Common Met.*, 82, 219, 1981.

13. McClellan, K.J., F. Chu, J. M. Roper, and I. Shindo, *J. Mat. Sci.*, 36, 3403, 2001.

14. Shirai, K., *Phys. Rev. B*, 55, 12235, 1997.

15. Gazonas, G.A., J.W. McCauley, I.G. Batyrev, D. Casem, J.D. Clayton, D.P. Dekar, R. Kraft, B.M. Love, B.M. Rice, B.E. Schuster, and N.S. Weingarten, *ARL-RP-337*, 2011.

Large-Scale Molecular Dynamics Simulations of Early- and Intermediate-Stage Sintering of Nanocrystalline SiC

Bryce D. Devine, Jeffrey B. Allen, Charles F. Cornwell, and Charles R. Welch
US Army Engineer Research and Development Center (ERDC), Vicksburg, MS
{bryce.d.devine, jeffrey.b.allen, charles.f.cornwell, charles.r.welch}@usace.army.mil

Abstract

Polycrystalline silicon carbide (SiC) has tremendous potential as a lightweight structural material if its fracture toughness and tensile strength could be significantly (factor of 4) improved, which is the long-term goal of this and related research. Such a "super" ceramic would allow for two-thirds weight reduction, or more, over that of steel and aluminum for most structural applications. The potential impact on military logistics is enormous. Key to the realization of such a super ceramic is the development of appropriate SiC composite designs and the development of methods to fabricate SiC composites to meet these designs through sintering. Technologies to support SiC composite design development are addressed in a companion paper. This paper discusses research to develop sintering fabrication methods. Recently developed sintering techniques allow for the production of ceramic materials with nanocrystalline grain structures, and for the incorporation of organic reinforcements in ceramic composites. Both reduction in grain-size and the incorporation of tensile members have been shown to improve the fracture toughness of SiC. We are performing multi-million-atom classical molecular dynamics (MD) simulations of early- and intermediate-stage Spark Plasma Sintering (SPS) of nanocrystalline SiC to better understand, and then engineer, the sintering process. We have developed continuum models to predict the thermal, electric, and displacement fields inside the sintering chamber. These provide boundary and initial conditions for the MD simulations of sintering. Several mechanisms were observed during each stage of sintering consolidation, with the rate-limiting mechanism dependent upon temperature, pressure and grain-size. This research helps lay the technical foundation for development of a lightweight structural "super" ceramic matrix composite.

1. Introduction

High-performance silicon carbide (SiC) ceramics are remarkably strong and lightweight materials, and potentially offer dramatic performance gains over current materials in structural applications. A few properties of current SiC-based materials are compared with other structural materials in Table 1. SiC ceramics are twice as stiff as steels, have three-times the compressive strength of very high-strength steel yet are comparable in density to aluminum alloys (Howatson, Lund, and Todd, 1991). The main limitation with ceramics materials in general is their relatively-low fracture toughness and weak bending strength, which is related to the tensile strength. If the fracture, toughness and tensile strength can be improved, SiC could replace steels and aluminum alloys many applications at a fraction of the weight, and possibly, at a fraction of the cost over the lifetime of the part.

Single-crystal SiC is very strong, with an ultimate tensile strength of 8.4 GPa (Petrovic, et al., 1985), more than 10-times that of high-strength steels. However, ceramic materials in application are polycrystalline which exhibit considerably-lower tensile strengths because of pores and micro-cracks that form within the matrix during synthesis. When overloaded, these defects in the microstructure act as stress risers leading to fracture and brittle failure at lower applied loads.

High-strength ceramic materials are produced by a sintering process, in which a green form consisting of a compacted powder is heated to and held at a temperature near the melting point. The individual particles in the green form fuse together to form a continuous solid network. Without modification, the process typically results in a porous microstructure that is weak and brittle in tension. Application of pressure, called hot pressing, and the addition of sintering aids, such as lower melting point materials that form secondary liquid phases between grains, have greatly improved the density and consequently the performance of the sintered materials (Rahaman, 2008).

Table 1. Material properties of SiC compared with common structural materials

Material	Young's Modulus (million psi)	Compressive Strength (ksi)	Density (lb/ft^3)	Fracture Toughness ($MPa\ m^{0.5}$)	Bending Strength (ksi)	Max Service Temp (F)
Silicon Carbide (SiC)	59.5	566	194	4.6	80	3000
Boron Carbide (B_4C)	63.8	560	157	3.1	62	1112
Aluminum Oxide (Al_2O_3)	43.5	305	230	3.5	47	3092
High Strength Steel (4340)	29.7	160 - 200	490	50	160 to 200	2600
Aluminum 7075-T6	10.4	74 - 78	170	24	74-78	900

Traditional liquid phase sintering methods have been optimized for SiC to the point at which the strength of the final materials is approaching the theoretical maximum of the single crystals. In these high-performance materials, fracture occurs along the shear planes within the individual SiC crystals, rather than through the grain boundaries between crystals. Routes to further improvement are focused on optimizing the sintering process to produce strong defect-free crystals and inclusion of tougher materials as tensile reinforcements.

One possible means to improve the strength of SiC is to maintain a nanocrystalline grain-size during sintering. Hardness and strength have been shown to increase with decreasing grain-size in SiC (Szlufarska, Nakano, and Vashishta, 2005; Mo and Szlufarska, 2007). During sintering, grain growth occurs concurrently with consolidation. Under conventional conditions, a nanocrystalline grain structure cannot be produced in the final product because of grain growth, even when a finally-graded, nanocrystalline powder is used as a starting material. Recently the development of field-assisted sintering techniques (FAST), also known as spark plasma sintering (SPS), allows for the production of nanocrystalline ceramics (Orru, et al., 2009). During FAST processing, an external electric field is applied across the compacted powder while the powder is heated and pressed, which increases the densification rate and allows for sintering at lower temperatures, resulting in reduced grain growth. The lower processing temperature also allows for the inclusion of temperature-sensitive tensile phases such as carbon nanotubes.

As a fairly new technique, FAST sintering is not well understood from a mechanistic point. Some fundamental aspects of the process, such as what effect the field provides, and the atomic mechanism that leads to densification at lower temperature are yet to be confirmed (Langer, Hoffmann, and Guillon, 2009). Multi-scale modeling is being used to address these basic questions about the process. Modeling efforts range from the atomistic to the continuum level. At the continuum scale, finite element modeling of the sintering chamber is being used to explore the temperature, electrical current, and stress distribution within the sample over time. The temperature and stress distributions set the boundary and initial conditions for the concurrent molecular dynamics simulations that identify the densification mechanisms that are rate-limiting at various temperatures and grain-sizes. Aspects of both these efforts are presented in this article.

2. Methods

2.1. Finite Element Modeling

Recently, we developed a fully-integrated thermal/electrical/mechanical multi-physics model to explore the evolution of the temperature and stress fields within the sample during the consolidation process (Allen, et al., forthcoming). The model incorporates a Proportional Integral Derivative (PID) thermal controller (Munoz, et al., 2010) that allows for control over the heating rates and includes effects caused by contact forces, radiation, and Joule heating. Thermal control of the system requires a description of material properties that vary with temperature and density.

A schematic of the FAST apparatus modeled in this study is presented in Figure 1. The powder sample is held within a graphite die between two graphite punches. During sintering, an electrical current is applied at high-amperage and low-voltage through the conductive punches and the sample being sintered. Pressure is concurrently applied to the sample to assist the consolidation process.

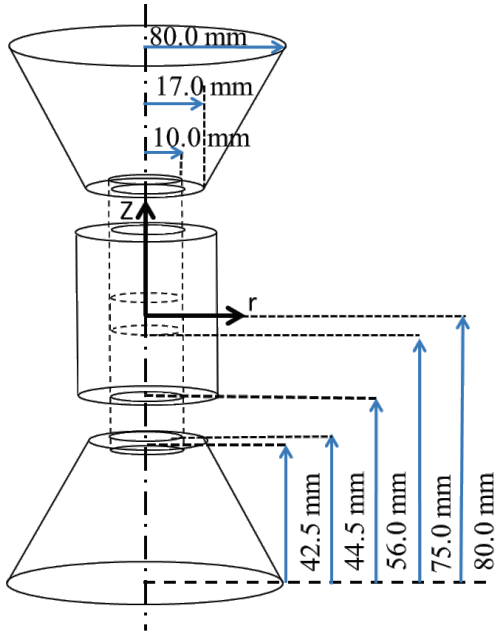


Figure 1. A wireframe schematic of the FAST apparatus (FCT HP 25/1, FCT, Germany) used in this study showing nominal dimensions and placement of the origin in cylindrical coordinates

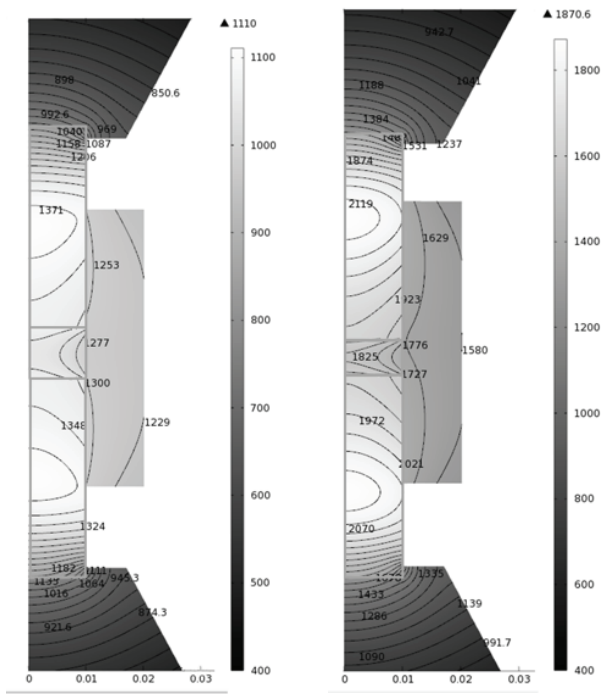
Comparisons of the model with experiments revealed excellent agreement in terms of quantifying the temperature evolution at a fixed-point outside the sample. Within the sample, the numerical results show significant, non-homogeneous temperature and hydrostatic stress distributions, both decreasing with radial distance from the centerline of the sample toward the die interface. The significant thermal gradients that arise during the process are illustrated in the contour plots in Figure 2. Previous studies have indicated that these temperature and stress gradients could compromise the integrity of the material.

Current simulations are exploring the effect of densification kinetics within the model, and reveal the importance of including such effects particularly during the early stages of sintering. The kinetic properties of the consolidation reactions within nanocrystalline SiC are unknown and are being explored through atomistic modeling.

2.2. Molecular Dynamics Simulations

The most computationally-expensive portion of this work consists of molecular dynamics (MD) simulations of the sintering process. This process determines the atomic scale mechanisms that govern the consolidation reactions as a function of grain-size and temperature. The complete sintering process occurs on the time-scale of minutes, which is beyond the scope of atomic-scale simulations.

The sintering process is typically described in terms of three stages (Rahaman, 2008). In the early or initial stage, necks form between separate particles and densification is governed by neck growth. Particles can slide and rotate to form lower energy interfaces that promote inter-granular growth. As necks grow, the pore sizes are reduced. Migration of material to-and-from free-surfaces is restricted. Densification becomes limited more by grain boundary and lattice diffusion. This diffusion-limited phase is characteristic of the intermediate stage of sintering. Pores remain interconnected during the intermediate stage. Eventually, in the final stage of sintering the intergranular channels close off and form isolated pores. The kinetic phenomena in the final stage are typically modeled as those of the bulk material. Until the availability of current high performance computing (HPC) systems, atomistic modeling of sintering was limited to the early-stage phenomena. However, intermediate-stage sintering processes can be observed on a time-scale of a few nanoseconds, which is now quite tractable.



t=523 sec. t=838 sec.

Figure 2. Temperature contour plots corresponding to two time intervals: t=523 sec. and t=838 sec

2.3. Molecular Dynamics Simulations Computational Details

All MD simulations were performed using the parallel COMB suites of codes developed at the University of Florida (Devine, et al., 2011). A three-body Tersoff potential parameterized by Albe and Erhart for Si, C, and SiC was used to describe bonded interactions (Tersoff, 1988; Erhart and Albe, 2005). Simulations were performed in the constant temperature and pressure (NPT) ensemble. Temperature during sampling was controlled with a Nose-Hoover chain thermostat, which has been shown to properly sample the statistical ensemble (Martyna, Klein, and Tuckerman, 1992). Pressure was maintained through a Parrinello-Rahman dynamical bariostat with periodic boundary conditions applied in three-dimensions (Parrinello and Rahman, 1981).

The model system consisted of 32 spherical nano-crystals of 3C-SiC stacked in a cubic close-packed arrangement. Crystals are of equal size to minimize effects due to grain growth, with sizes ranging from 3 nm to 20 nm in diameter. Crystals are randomly oriented relative to one another. Each crystal has 12 neighbors yielding 384 different interfaces. The largest systems in this study contain 16 million atoms.

During simulations, the systems are initially optimized via steepest descent algorithm and annealed at low-temperature to eliminate erroneous high-energy configurations. The system is then heated to 300K and compressed at 100 MPa to form a compacted green body. The compacted system was heated to the sampling temperature and pressure at a rate of 1 K/ps and held for 5 ns.

3. Results and Discussion

Throughout the simulations, temperatures were reported as a fraction of melting point (T_m). Sintering data was collected at temperatures ranging from 0.5 to 0.8 T_m. A comparison of density over time at each temperature, which is plotted in Figure 3 for a system of 5 nm crystals, demonstrates that early and intermediate stage phenomena are exhibited on the time scale of MD. At lower-temperatures, consolidation proceeds via neck growth. Initially, particles rotate to form lower-energy interfaces. The combination of grain rotations and neck formation leads to an increased rate of densification in the first few hundred ps of the simulation. Once necks form, densification proceeds at comparable rates at temperature below 0.8 T_m.

The increased rate seen at 0.8 T_m in Figure 3 is attributed to melting at the particle surfaces, indicating 5-nm-diameter crystals are below the critical size at that temperature. A plot of density versus time at 0.8 T_m for larger crystals, shown in Figure 4, indicates that larger crystals are more stable at this temperature. Likewise, the surfaces of 3-nm crystals begin melting at temperatures below 0.6 T_m.

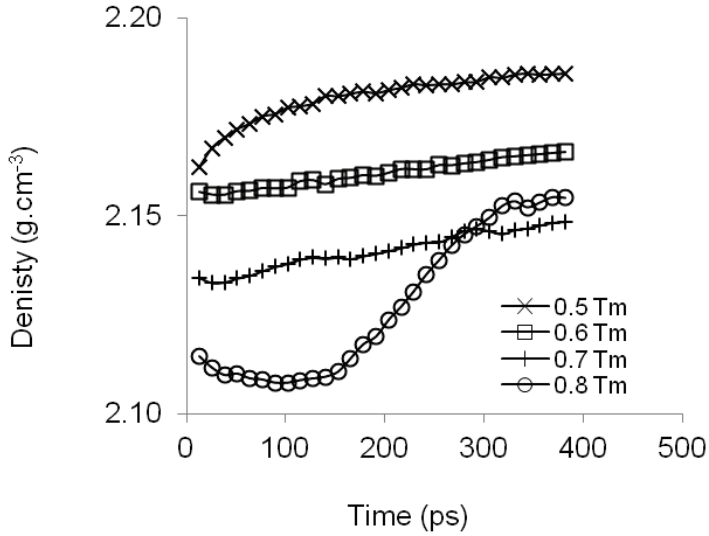


Figure 3. Density as a function of time for 5-nm sized crystals of 3C-SiC. The drop in density between temperatures is due to thermal expansion.

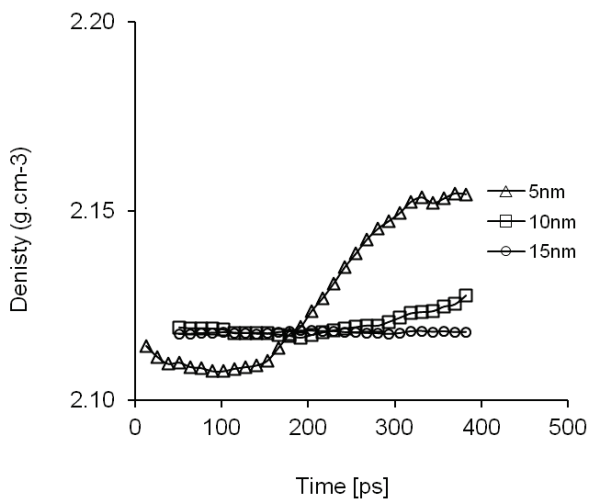


Figure 4. Density as a function of time at 0.8 T_m for various-sized crystals

3.1. Early Stage Sintering Mechanisms

Early-stage neck formation was observed during the simulations. Figure 5 shows an image of a 10-nm system early in the simulation at 0.5 T_m. Color in the image indicates defects; orange atoms occupy sites of point defects or dislocations, grey atoms occupy normal lattice sites, and black atoms are under-coordinated. Neck formation occurs preferentially between favorably-aligned interfaces, and does not occur between misaligned interfaces. This is consequential since most early-stage models and current meso-scale models assume neck formation across all interfaces (Rahaman, 2008). The material in the neck regions is perfectly crystalline, suggesting that material is moving into the neck region as vacancies move out, which is consistent with traditional models (Coble, 1970). No defects are observed within the crystals, suggesting that densification at this temperature and stage occurs predominately via surface diffusion.

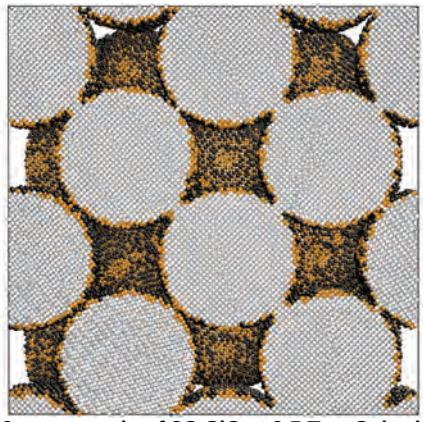


Figure 5. Neck formation and growth in 10-nm crystals of 3C-SiC at 0.5 T_m. Color indicates defects; black atoms are under-coordinated, orange atoms are dislocations or point defects, and grey atoms occupy normal lattice sites.

3.2. Intermediate Stage Sintering Mechanisms

Sintering is limited by diffusion with a rate equation that depends on the rate limiting diffusion mechanism. FAST sintering has been shown to follow similar rate behavior as hot pressing high-pressure. In which case, the rate of densification is given by (Rahaman, 2008; Coble, 1970):

$$\frac{1}{\rho}\frac{d\rho}{dt} = \frac{HD\phi^n}{G^m kT} P_a^n \tag{1}$$

where ρ is the relative density, H is a constant, D is the diffusion coefficient, k is Boltzmann's constant, T is temperature, P_a is the applied pressure, G is the grain size, and ϕ is a stress intensification factor. The exponents, m and n, vary with the rate-limiting mechanism according to Table 2. A slope of a logarithmic plot of the $d\rho/dt$ as a function of grain-size at constant pressure should indicate the primary rate-controlling mechanism at each temperature.

Table 2. Associated exponents for hot pressing mechanisms

Mechanism	Grain Size Exponent, *m*	Stress Exponent, *n*
Lattice diffusion	2	1
Grain boundary diffusion	3	1
Plastic deformation	0	≥3
Viscous flow	0	1
Grain boundary sliding or particle rotation	1	1 or 2

Densification rate as a function of grain-size at constant temperature and pressure is plotted in Figure 6. The values for the grain size exponent, m, indicate that at lower temperature, grain motion such as sliding and rotation is the limiting mechanism, although surface diffusion may also contribute significantly. At 0.6 T_m, densification is limited by lattice diffusion and finally by grain boundary diffusion at the upper temperatures.

The corresponding activation energy, Q_{act}, can be determined with an Arrhenius plot of $d\rho/dt$ versus T^{-1}. Figure 7 shows the plots for a 5-nm crystal system at 100 MPa for the various temperatures. The analysis yields linear curves for each temperature. Activation energies are derived from the slopes of the curves. At 0.5 T_m, Q_{ac} is determined to be 63.9 $kJ \cdot mol^{-1}$ which correlates with published values for carbon interstitial and surface diffusion of 58 $kJ \cdot mol^{-1}$. Similarly, the Q_{act} at 0.6 T_m is determined to be 338 $kJ \cdot mol^{-1}$, which corresponds to the reported values for C vacancy diffusion of 338 $kJ \cdot mol^{-1}$. At 0.7 T_m, the calculated Q_{act} of 342 kJ·mol-1 also corresponds to values for lattice diffusion of Si vacancy (Bockstedte, Mattaausch, and Pankratov, 2004).

Lattice diffusion is also observed during the MD simulations. Snapshots from the simulations show abundant lattice defects at temperatures of 0.6 T_m and above. Figure 8 shows the same 10-nm crystal system as shown in Figure 5 at 0.7 T_m. The numerous orange atoms within the crystals indicate point defects that are rarely observed at 0.5 T_m.

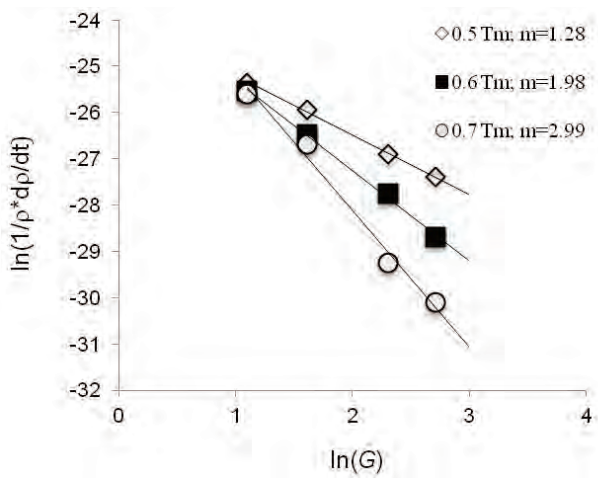


Figure 6. Densification rates as a function of grain size at constant temperature and pressure

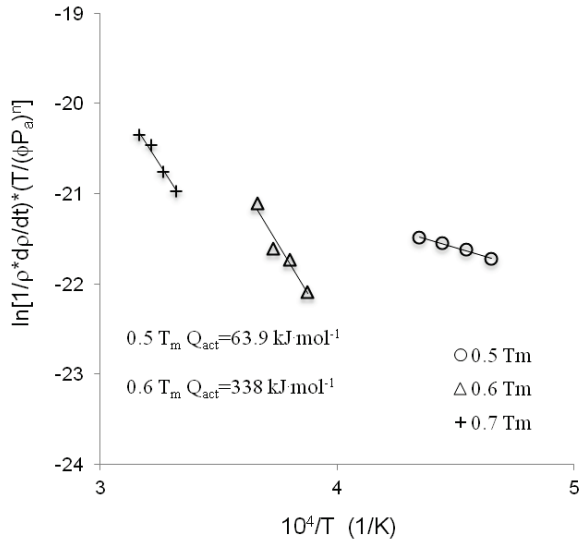


Figure 7. Arrhenius plots of densification rates as a function of T^{-1} at constant temperature and pressure

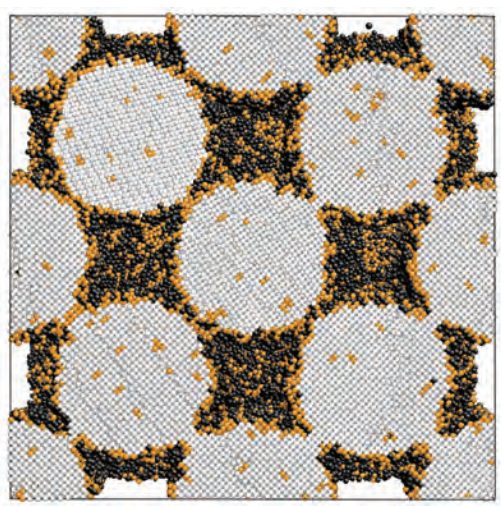


Figure 8. Point defects in 10-nm crystals at 0.7 T_m. Color indicates defects; black atoms are under-coordinated, orange atoms are dislocations or point defects.

4. Conclusion

Recent advancements in sintering, such as FAST techniques, enable the production of nanoscale ceramic materials and possibly nano-composites. However, optimizing these accelerated processes requires a much greater understanding of the governing mechanism than what we currently have. Our approach is to use multi-scale modeling to analyze the conditions that occur in the sintering chamber during processing and to identify the atomic scale mechanism that influences densification under those conditions. The atomistic level simulations to date have revealed several important phenomena, such as that neck formation occurs preferentially between favorably aligned interfaces and does not occur between misaligned interfaces. These results can be used to improve early-stage models and current meso-scale models that assume neck formation across all interfaces. Other findings from the MD simulations, such as activation energies, can be used to parameterize these longer length-scale models, allowing the meso-scale model to describe grain growth under various sintering conditions.

The intended outcome of this work is to have a continuous multi-scale capability that can be used to enhance our control of the sintering process to produce nano-crystalline ceramic materials with only an acceptable level and type of defect. Such ceramics should rival the strength of steel with greater stiffness, and with effective densities comparable to aluminum. Such a material would have revolutionary impact on military logistics.

We point out that the MD simulations of sintering are computationally-demanding to the point where realistic simulations cannot be performed without the capabilities provided by our HPC resources. Any development of larger length-scale models aided by MD simulation results would be greatly diminished, and nearly impossible without our ability to run these large atomic scale simulations.

Acknowledgements

The authors gratefully acknowledge funding support from the US Army Engineer Research and Development Center research programs “Nanoscale Studies of Polycrystalline Materials with Emphasis on Ceramic Synthesis” and “Multi-Scale Understanding and Engineering of Grain Boundaries in Ceramics,” and the allocation of computer time from the High Performance Computing Challenge Project “Molecular Dynamic Simulations to Underpin the Design and Development of High Performance Carbon Nanotube based Filaments, Membranes, and Coatings.” We thank the Chief of Engineers for permission to publish this article.

References

Allen, J.B., C.F. Cornwell, B.D. Devine, J.F. Peters, and C.R. Welch, “Finite Element Simulations of the Field Assisted Sintering Technique.” *Computational Materials Science*, forthcoming.

Bockstedte, M., A. Mattaausch, and O. Pankratov, “*Ab Initio* Study of Annealing of Vacancies and Interstitials in Cubic SiC: Vacancy-Interstitial Recombination and Aggregation of Carbon Interstitials.” *Phys. Rev. B.*, 69(23), 2004.

Coble, R.L., “Diffusion Models for Hot Pressing with Surface Energy and Pressure Effects as Driving Forces.” *J. Appl. Phys.*, 41(12), 1970.

Devine, B.D., T. Shan, A.J. McGaughey, S.R. Phillpot, and S.B. Sinnott, “Atomistic Simulations of Copper Oxidation and Cu/Cu_2O Interfaces using COMB Potentials.” *Phys. Rev. B.*, 84(12), 2011.

Erhart, P. and K. Albe, “Analytical Potential for Atomistic Simulations of Silicon, Carbon, and Silicon Carbide.” *Phys. Rev. B.*, 71(14), 2005.

Howatson, A.W., P.D. Lund, and J.D. Todd, *Engineering Tables and Data*, Chapman and Hall, New York, Sharpe, W.N.G.M, 1991.

Langer, J., M.J. Hoffmann, and O. Guillon, “Direct Comparison Between Hot Pressing and Electric Field-Assisted Sintering of Submicron Alumina.” *Acta Materialia*, 57(18), 2009.

Martyna, G., M.L. Klein, and M. Tuckerman, “Nose-Hoover Chains: The Canonical Ensemble via Continuous Dynamics.” *J. Chem. Phys.*, 97(4), 1992.

Mo, Y. and I. Szlufarska, “Simultaneous Enhancement of Toughness, Ductility, and Strength of Nanocrystalline Ceramics at High Strain-Rates.” *Appl. Phys. Letts.*, 90(18), 2007.

Munoz, S. and U. Anselmi-Tamburini, “Temperature and Stress Fields Evolution During Spark Plasma Sintering Processes.” *J. Mat. Sci.*, 45, 2010

Orru, R., R. Licheri, A. Locci, A. Cincotti, and G. Cao, “Consolidation/Synthesis of Materials by Electric Current Activated/Assisted Sintering.” *Mater. Sci. Eng.*, R 63, 2009.

Parrinello, M., and A. Rahman, “Polymorphic Transitions in Single Crystals: A New Molecular Dynamics Method.” *J. Appl. Phys.*, 52(12), 1981.

Petrovic, J.J., D.L Milewski, and F. Rohr, “Tensile Properties of SiC Whiskers.” *J. Mat. Sci*., 20, 1985

Rahaman, M.N., *Sintering of Ceramics*, Boca Raton, FL, Taylor and Francis, 2008.

Szlufarska, I., A. Nakano, and P. Vashishta, “A Crossover in the Mechanical Response of Nanocrystalline Ceramics.” *Science*, 39, 2005.

Tersoff, J., “Empirical Interatomic Potential for Silicon with Improved Elastic Properties.” *Phys. Rev. B.*, 38(14), 1988.

Modeling the Combustion Chamber Dynamics of Two Selectable-Thrust Rocket Motor Concepts

Michael J. Nusca, Chiung-Chu Chen, and Michael J. McQuaid
US Army Research Laboratory, Weapons and Materials Research Directorate (RDRL-WML-D), Aberdeen Proving Ground, MD
{michael.j.nusca, chiung-chu.chen, michael.j.mcquaid}.civ@mail.mil

Abstract

Challenge Project C4C is dedicated to developing and applying high performance computing capabilities to accelerate the development of hypergolic and hybrid rocket motor concepts. Computational fluid dynamics is employed to model chemically-reacting flows within motor combustion chambers, and computational chemistry is employed to characterize propellant physical and reactive properties. Recent accomplishments are presented and discussed.

1. Introduction

The US Army is developing hypergolic bipropellant and hybrid rocket motor concepts for tactical missile propulsion systems[1–3]. Hypergolic propulsion systems employ a pumpable (i.e., liquid or gel) fuel and pumpable-oxidizer that ignite spontaneously when mixed at ambient temperatures and pressures. Hybrid propulsion systems employ a pumpable-oxidizer with a solid fuel-grain. Both types have many potential advantages over the solid propellant-based rocket motors currently employed to propel tactical missiles. Among them is active thrust control, which, increasing targeting options and range will increase a tactical missile's lethality, while reducing the vulnerability of its launch platform. Because the fuel and oxidizer are stored separately in such systems, they are also inherently insensitive to a variety of stimuli that can produce catastrophic events in solid-propellant-fueled rocket motors. There are; however, a number of technical issues that must be addressed before these technologies can be fielded.

The hypergolic (bipropellant) propulsion system concept that the Army is developing for tactical missile applications is referred to as the Impinging Stream Vortex Engine (ISVE)[1,2]. In the ISVE, the fuel and oxidizer injection ports are located in the radial-wall of the combustion chamber, and they are oriented so that initially (at least) the fluids flow tangential to it. This configuration forces the propellants to mix in a highly-turbulent region between the injector orifices and the wall, promoting both an increase in the mixing path length and lower wall temperatures. These attributes, which have been observed in experimental tests of the concept conducted by the US Army Aviation and Missile Research, Development, and Engineering Center (AMRDEC), facilitate the design of propulsion systems that are smaller and lighter than traditional hypergolic propulsion system concepts. However, given the complexity of the interplay between design variables and engine performance, optimization of an engine design for a specific application based solely on experimental testing would be extremely inefficient and costly.

Propellant physical properties and reactivity being among the variables with a significant influence on engine design and performance, the selection and development of a fuel-oxidizer combination has been an important and integral aspect of the ISVE development effort. Until eight years ago, the combination employed in AMRDEC's test program was monomethylhydrazine-red fuming nitric acid (MMH-RFNA)[4]. (RFNA is composed primarily of nitric acid [HNO_3] and nitrogen dioxide [NO_2].) However, the superior performance attributes of MMH became outweighed by its carcinogenic potential, and the Army has since investigated a number of alternatives. The two classes of compounds considered to have the best prospects for meeting desired performance objectives, while being acceptable from the standpoint of the risks they pose to human health and the environment are saturated alkylamines and ethanamine azides[5]. Candidates that are representative of these two classes are, respectively, N,N,N',N'-tetramethylethylene-1,2-diamine (TMEDA) and 2-azido-N,N-dimethylethanamine (DMAZ), and a TMEDA-DMAZ blend referred to as TEDMAZ is the current candidate of choice.

In addition to the ISVE, AMRDEC is developing a novel hybrid rocket motor concept for tactical missile applications. The concept employs RFNA as an oxidizer, and has a fuel grain formulated with hydroxyl-terminated polybutadiene (HTPB, $C_{7.1102}H_{10.813}N_{0.1071}O_{0.1375}$)[3]. Among the combination's attributes is hypergolicity sufficient for an engine to be shutdown and reliably relit in flight following a pyrotechnic-ignited boost phase.

To gain insight into the relationships between design parameters and the performance of AMRDEC's hypergolic and hybrid rocket motor concepts, and thereby accelerate their development, the US Army Research Laboratory (ARL) has developed and employed models for simulating the combustion chamber flow physics and gas dynamics (via computational fluid dynamics, CFD), and propellant physical properties and reactivity (via computational chemistry and material science, CCM). These efforts are being performed with resources provided through the Challenge Project C4C. Presented herein are overviews of the CFD and CCM technical approaches and a brief summary of results achieved over the past year.

2. CFD Modeling

Modeling of the ISVE's combustion chamber fluid dynamics is being conducted using a CFD methodology that was developed to simulate unsteady, multi-component, chemically-reacting flows for a variety of gas-dynamic systems[6–9]. The latest upgrades to the model for the ISVE application include: 1) the addition of a sub-model for the simulation of injection schedules enabled by a pneumatically-actuated bipropellant valve (PABV) system[2,6], and 2) a reduced TMEDA-RFNA chemical kinetics mechanism.[10,12]

The PABV system is being employed by AMRDEC for the development of flight-weight ISVE designs[2]. The advantage of this system over prior valving systems is that it effectively decouples the feed lines from the injectors, thereby reducing pressure-drops and enabling more responsive injection control. Figure 1 shows schematics of an ISVE equipped with a PABV system. Each of three injector pairs has a dedicated PABV unit, and the units operate independently. Thrust levels are varied by varying the number of injectors that are open at a given time. The configuration shown is designed to deliver 800 lb_f of thrust when all three injector pairs are open.

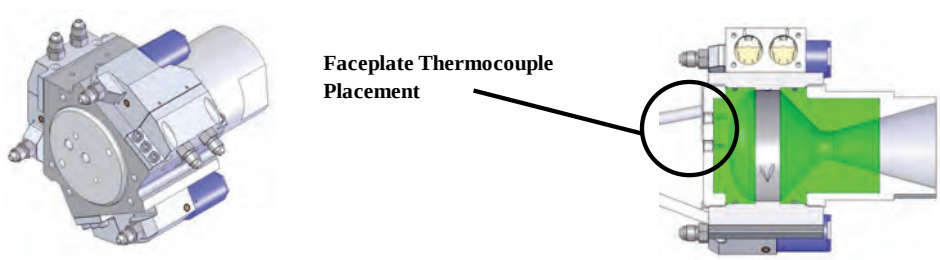


Figure 1. Isometric and cross-section views of a heavy-walled ISVE with a PABV system[2]

Another ISVE design consideration that remains under investigation is the fuel. In particular, concerns about the aging characteristics of DMAZ have led to consideration of the possibility of eliminating it from TEDMAZ. That would not only address the concern about DMAZ's aging, it would simplify the fuel blending process and preclude having fuel components with the potential to separate during storage. To make a case for testing this possibility experimentally, CFD simulations were performed to predict the performance of a TMEDA-RFNA-fueled ISVE firing vs. a TEDMAZ-RFNA-fueled firing[11]. To perform the study, full (381 species and 1401 reactions)[12] and reduced (73 species and 88 reactions) chemical kinetic mechanisms for TMEDA-RFNA[10,13] were developed using the same techniques employed to develop their TEDMAZ-RFNA counterparts.

Figure 2 compares computed pressures and thrust for simulated TEDMAZ-RFNA-fueled and TMEDA-RFNA-fueled ISVE firings. (The TEDMAZ-RFNA simulation is described in detail in Reference 11.) The injected fuel and oxidizer mass flow-rates were specified to mimic the first-pulse of a throttle schedule employed for a test-stand firing designated PABV04. This throttle schedule, which included four 0.3 sec thrusting-pulses separated by about 4.5 seconds of engine dwell-time, is described in detail in Reference 14. The times given in the figures in this paper correspond to those reported for the actual test. The comparison predicts that, compared to the TEDMAZ-RFNA-fueled firing, a TMEDA-RFNA-fueled firing employing the same throttle schedule will produce slightly higher pressure between injectors (green curve), slightly higher thrust (black curve), and slightly lower pressure near and in the engine head (red and blue curves).

Figures 3 through 8 show selected CFD results for the flowfield within an ISVE fueled with TMEDA-RFNA. (Note that these figures are not drawn to scale.) They can be compared to the corresponding results for an ISVE fueled with TEDMAZ-RFNA that is discussed in Reference 11. H_2O mass fraction contours produced by the TMEDA-RFNA

simulation are shown in Figures 3 and 4, and they can be compared to Figures 9 and 10 in Reference 11 for the TEDMAZ-RFNA simulation. Very similar patterns are observed in both cases, but the simulations predict that much less water will be produced when the ISVE is fueled with TMEDA-RFNA.

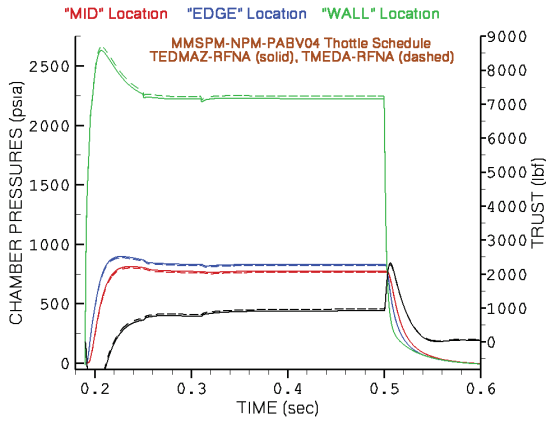


Figure 2. Computed ISVE chamber pressure and thrust profiles for the first thrust-pulse of a simulated firing with the PABV-04 throttle schedule: EDMAZ-RFNA vs. TMEDA-RFNA

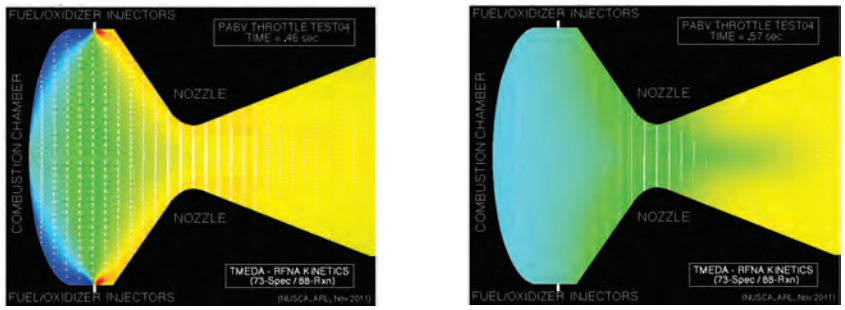


Figure 3. Computed H_2O mass fraction contours (blue to red: 0 to 0.01) and selected velocity vectors for a TMEDA-RFNA-fueled ISVE firing with the PABV-04 throttle schedule: During the first thrust-pulse (left) and dwell-time (right)

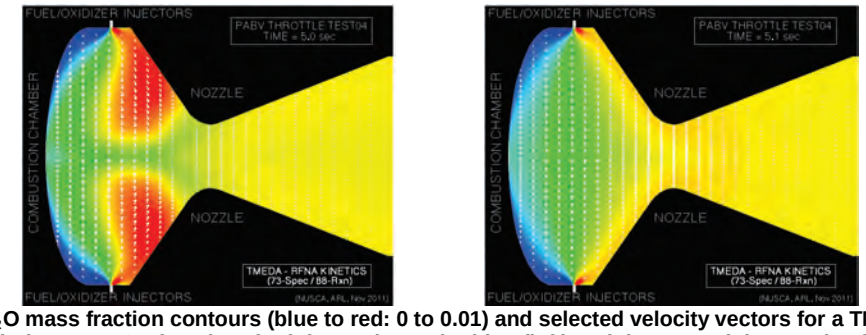


Figure 4. Computed H_2O mass fraction contours (blue to red: 0 to 0.01) and selected velocity vectors for a TMEDA-RFNA-fueled firing with the PABV-04 throttle schedule: During re-ignition (left) and the second thrust -ulse (right)

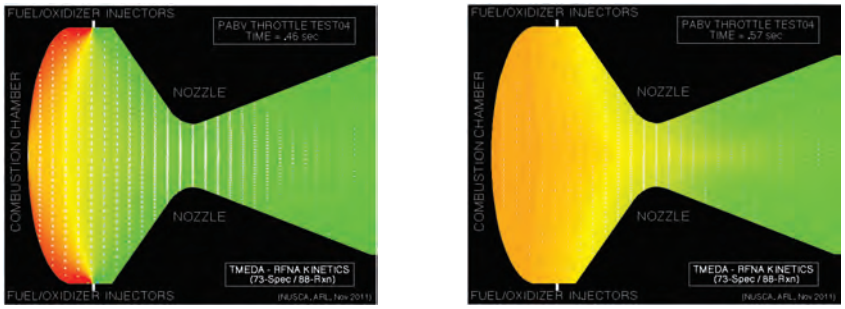


Figure 5. Computed OH mass fraction contours (blue to red: 0 to 0.75) and selected velocity vectors for a TMEDA-RFNA-fueled firing with the PABV-04 throttle schedule: During the first thrust-pulse (left) and dwell-time (right).

OH mass fraction contours produced by the TMEDA-RFNA simulation are shown in Figures 5 and 6, and can be compared to those shown in Figures 11 and 12 of Reference 11 for the TEDMAZ-RFNA simulation. Both the patterns and the levels of OH production indicated in the TMEDA-RFNA simulation are significantly different from those found in the TEDMAZ-RFNA simulation. In the TEDMAZ-RFNA simulation, OH is predominantly produced in the nozzle, while in the TMEDA-RFNA simulation, OH is produced primarily near the engine head.

Significant differences in the H-atom mass fraction contours were also observed for the two cases. Indeed they are so different that different scales were employed to render Figures 7 and 8 (for TMEDA-RFNA) and Figures 13 and 14 in Reference 11 (for TEDMAZ-RFNA). In the case of TMEDA-RFNA, H-atoms are predominantly produced near the engine head while in the case of TEDMAZ-RFNA, H-atoms are only found in significant amounts in the nozzle.

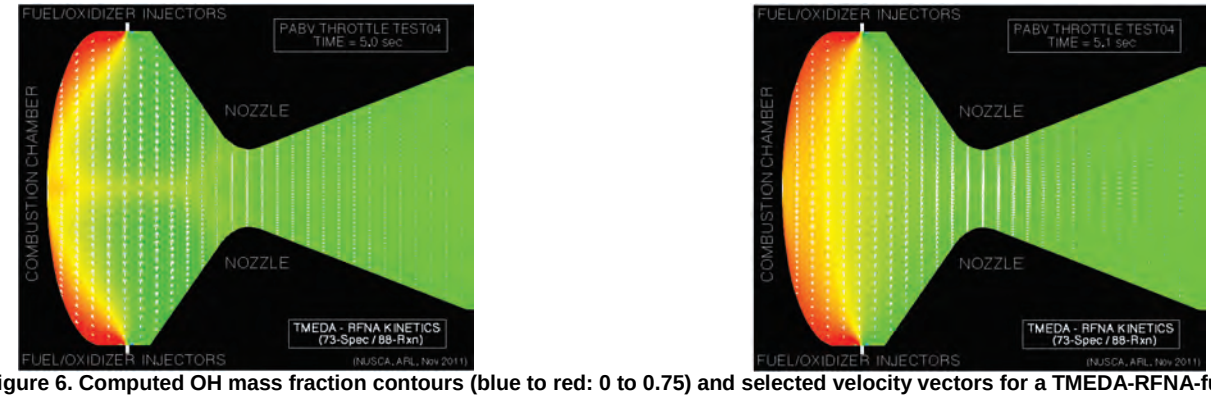


Figure 6. Computed OH mass fraction contours (blue to red: 0 to 0.75) and selected velocity vectors for a TMEDA-RFNA-fueled firing with the PABV-04 throttle schedule: During re-ignition (left) the second thrusting-pulse(right).

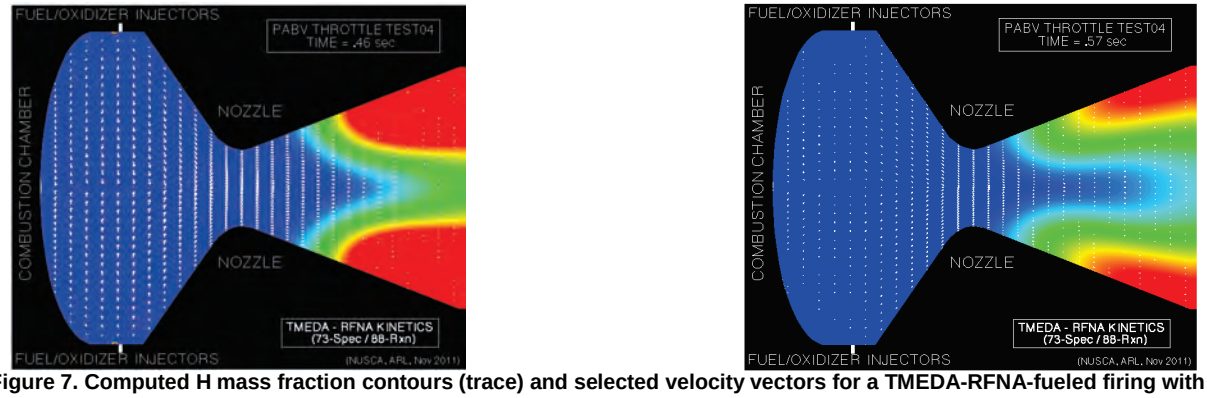


Figure 7. Computed H mass fraction contours (trace) and selected velocity vectors for a TMEDA-RFNA-fueled firing with the PABV-04 throttle schedule: During the first thrusting-pulse (left) and dwell-time (right).

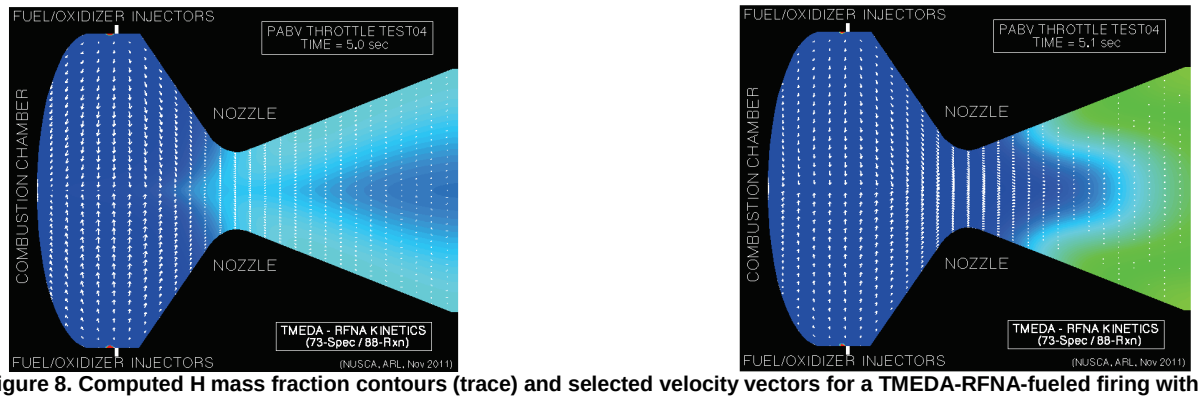


Figure 8. Computed H mass fraction contours (trace) and selected velocity vectors for a TMEDA-RFNA-fueled firing with the PABV-04 throttle schedule: During re-ignition (left) and the second thrusting-pulse (right).

In addition to performance, we were interested in the impact that a change from TEDMAZ to TMEDA would have on the chamber-wall temperature. Therefore, gas temperatures computed in the simulations were sampled near the head-end on the centerline (see Figure 1), and a heat conduction analysis was performed to compute the temperature of the wall just below the surface. (The analysis is discussed in Reference 11.) The results are shown in Figure 9. It is observed that gases adjacent to the chamber-wall are about 40 F higher in the TMEDA-RFNA simulation, and that produces a wall temperature that is about 30 F higher. Coupled with the results for performance, the results indicate that TMEDA could substitute for TEDMAZ if the increased thermal load on the chamber-wall can be managed.

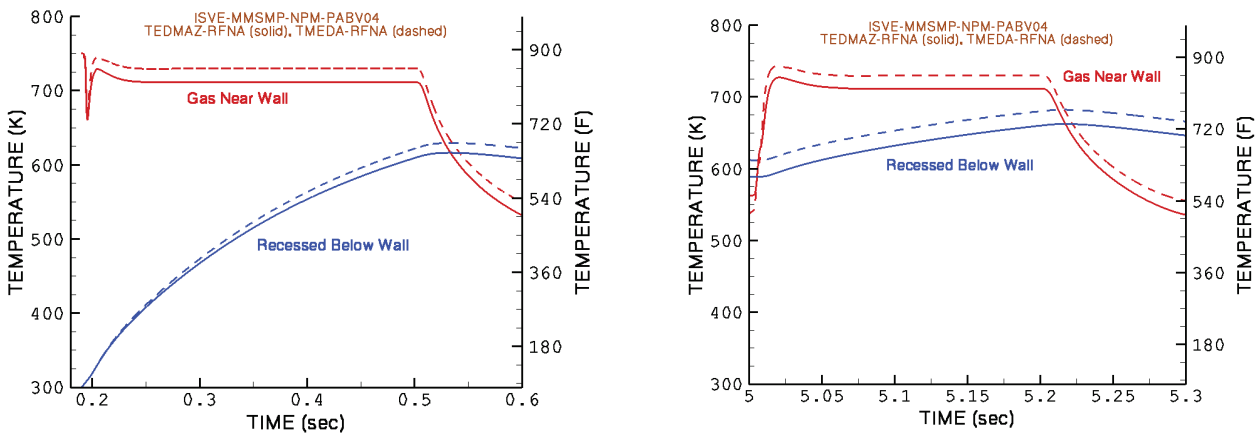


Figure 9. Computed gas and engine wall-temperatures the first throttle-pulse (left) and the second throttle-pulse (right) from an ISVE simulation with PABV04's throttle schedule: TEDMAZ-RFNA vs. TMEDA-RFNA.

A second objective of the Challenge Project was to advance a computational investigation of the AMRDEC hypergolic hybrid rocket engine-motor concept discussed by Wingard, Thompson, and Simmons[3]. That involved incorporating uupgrades to the CFD model of the concept[15], including: 1) assuming the propellant-grain (surface) was $C_{20}H_{32}$ (molecular weight 272) rather than the C_4H_6 (molecular weight 54), and 2) employing a finite-rate chemical kinetics mechanism to model reactions between gases evolving from the grain-surface and RFNA. The kinetics mechanism (93 species and 98 reactions)[10] was derived from a full HTPB-RFNA mechanism (469 species and 1609 reactions)[16] using a reduction routine described by McQuaid, Kotlar and Chen[10].

Figure 10(a) shows the computed results of HTPB grain regression and average engine-chamber pressure for three simulations: 1) HTPB modeled as C_4H_6 and no gas-phase chemical reactions (with RFNA), 2) HTPB modeled as $C_{20}H_{32}$ and no gas-phase chemical reactions, and 3) HTPB modeled as $C_{20}H_{32}$ and gas-phase chemical reactions with RFNA. The grain regression rate (k) was computed with a pyrolysis law written in Arrhenius form: $k=A'\exp(E_a/RT_s)$. For pure HTPB, the constants selected were E_a=4.91 kcal/mole and A'=11.04 mm/s for T_s > 722 K, and E_a=13.35 kcal/mole and A'=3965 mm/s for T_s<722 K (see Reference 7).

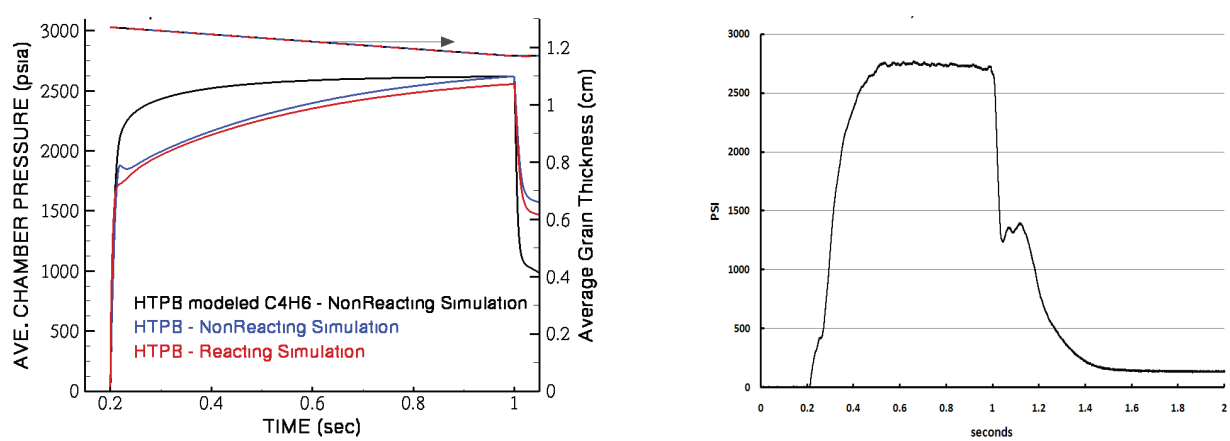


Figure 10. (a) Computed chamber pressure (average) and HTPB grain thicknesses for models with grain regression rates dictated by the RFNA injection rate: Simulations with and without a chemically-reacting gas phase. (b) Typical pressure vs. time measurement from a test firing of AMRDEC's Hypergolic Hybrid Rocket Engine [3].

For the two cases with no mechanism for gas-phase chemical reactions, there is a significant difference in the initial pressurization rates for $C_{20}H_{32}$ (HTPB) and C_4H_6. The difference is due mainly to the difference in molecular weight (i.e., pressure is inversely proportional to molecular weight), but it is also due to chemical reactions that are needed to break the long-chain hydrocarbons of $C_{20}H_{32}$ into smaller molecules. Comparing Figure 10(a) to the experimental pressure data given in Figure 10(b), it appears that a better match to the initial chamber pressurization transient is achieved in the $C_{20}H_{32}$ simulations.

To begin to understand the influence of gas-phase HTPB-RFNA reaction chemistry on the hybrid motor's performance, snapshots of the computed-mass fraction contours of various species were rendered for a very early time in the simulation. Figures 11 through 17 show representative examples. In each case a cross-section of the engine is displayed. Figure 11 shows computed $C_{20}H_{31}$ mass fractions. The analyis indicates this species is formed as soon as HTPB gas is evolved from the grain-surface, and it reacts promptly with the NO_2 (component) of RFNA. $C_{20}H_{31}$ reacts further to form C_4H_7 and $C_{16}H_{24}$, the mass fraction contours for which are given in Figures 12 and 13. The contours of HNO_2 are given in Figure 14, where it will be noted that this product is formed in very small amounts, and is confined to the part of the grain that is most rapidly regressing at the given point in time. As shown in Figure 15, $C_{20}O_3H_{32}$ reacts with NO_2, decomposing to form other daughters and NO (see Figure 16). As shown in Figure 17, HONO is also formed in small amounts. The region of the HTPB grain that is most rapidly regressing at the given time correlates with the location of the greatest concentrations of $C_{20}O_3H_{32}$, NO, and HONO. A comprehensive examination of the reaction kinetics associated with HTPB-RFNA and their influence upon engine performance is planned.

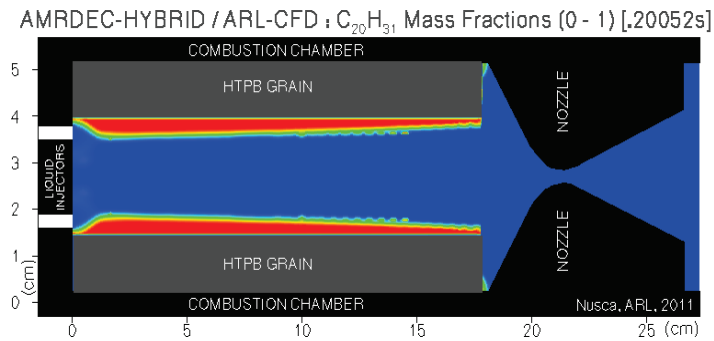


Figure 11. Computed $C_{20}H_{31}$ mass fraction contours (blue to red: 0 to 1) from a hybrid engine simulation

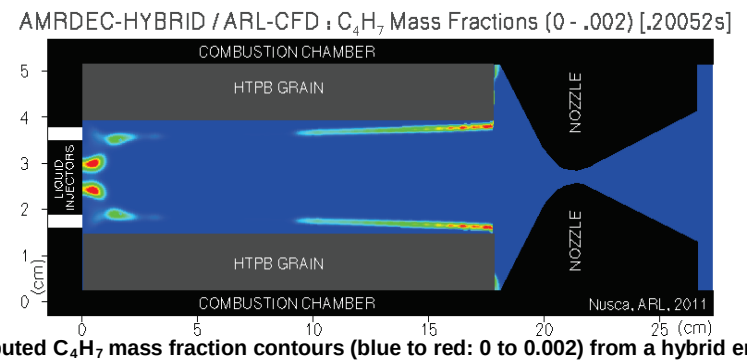


Figure 12. Computed C_4H_7 mass fraction contours (blue to red: 0 to 0.002) from a hybrid engine simulation

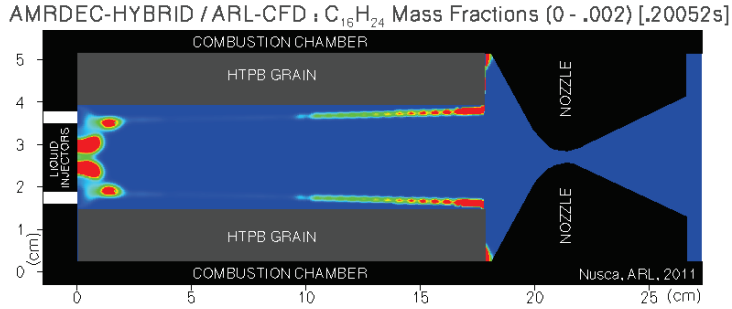


Figure 13. Computed $C_{16}H_{24}$ mass fraction contours (blue to red: 0.000 to 0.002) from a hybrid engine simulation

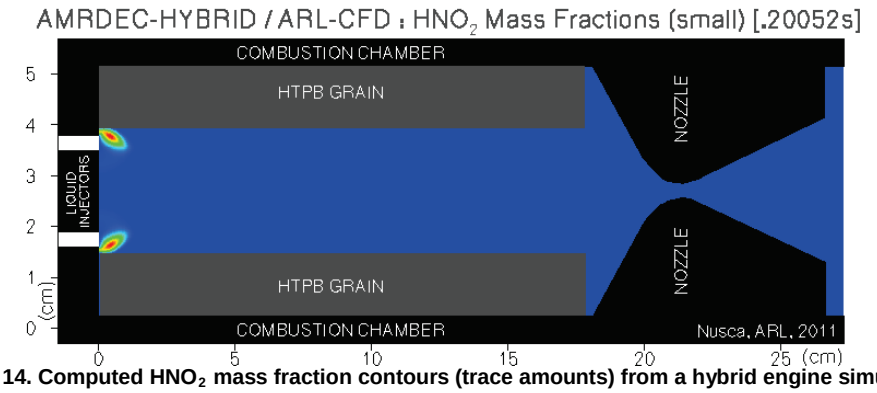


Figure 14. Computed HNO_2 mass fraction contours (trace amounts) from a hybrid engine simulation

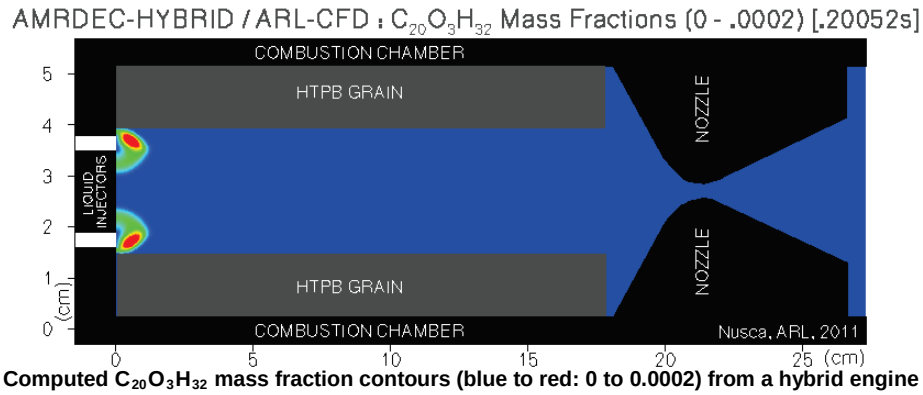


Figure 15. Computed $C_{20}O_3H_{32}$ mass fraction contours (blue to red: 0 to 0.0002) from a hybrid engine simulation

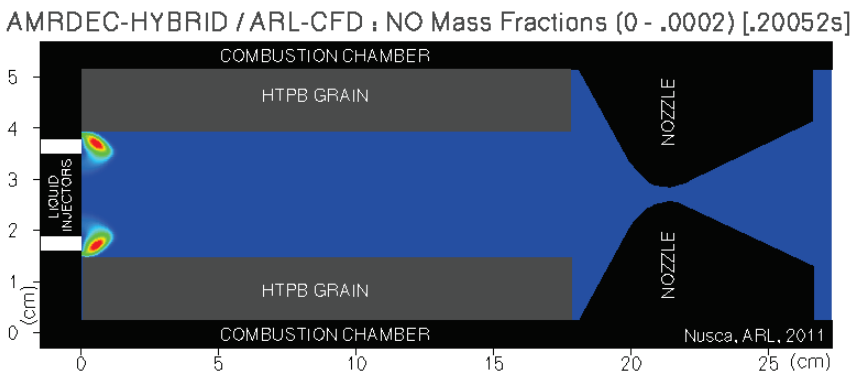


Figure 16. Computed NO mass fraction contours (blue to red: 0 to 0.0002) from a hybrid engine simulation

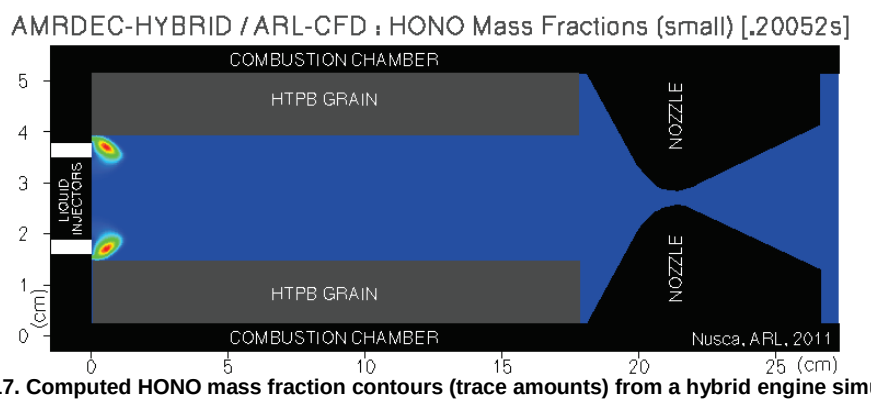
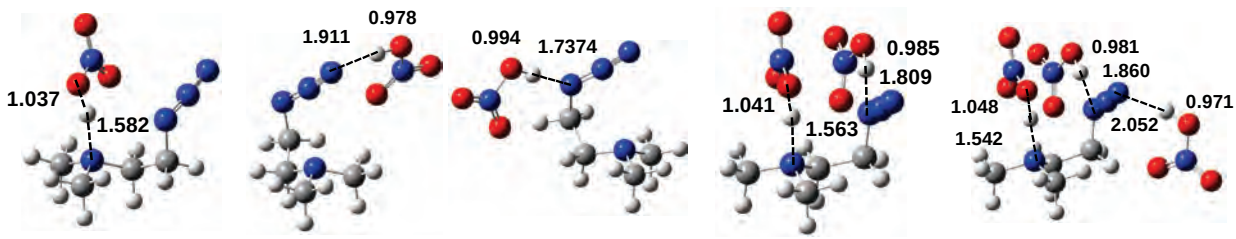


Figure 17. Computed HONO mass fraction contours (trace amounts) from a hybrid engine simulation

3. CCM Modeling

Computationally-based methods were employed to derive rate expressions for elementary reactions postulated for: 1) the thermal decomposition of DMAZ, 2) the (exothermic) complexation of DMAZ and HNO_3, 3) the abstraction of H-atoms from DMAZ by NO_2, 4) bimolecular reactions between (radical) intermediates and NO_2, and 5) unimolecular dissociations of R-NO_2 adducts. Thermochemical properties and reaction-path parameters were determined from the results of *ab initio* and density functional theory-based quantum chemistry calculations. Molecular structures and vibrational frequencies for postulated species were determined with a MPWB1K/6-31+G(d,p) model[17], and energies for the structures were determined via composite methods, including CBS-QB3[18,19], G3[20], G4[21], and G3MP2[22]. These data were then utilized to calculate rate expressions for the reactions based on: 1) the principles of thermochemical kinetics, 2) transition-state theory (TST)[23]/canonical variational transition-state theory (CVTST)[24], and 3) quantum Rice-Ramsberger-Kassel (QRRK) theory with master equation analysis[25–27].

Three types of DMAZ-HNO_3 complexes were found, with the H-atom of HNO_3 bonding to either the amine N-atom, the azido N-atom adjacent to the -CH_2 group, or the terminal azido N-atom. Molecular structures determined for the three types are shown in Figure 18. Based on results from a G3MP//MPWB1K model, the lowest-energy DMAZ-HNO_3 complex is 13.6 kcal/mol lower in energy than the DMAZ + HNO_3 reactant asymptote—a value that is similar to the analogous MMH + HNO_3 complexation reaction. The exothermicities of the complexation of DMAZ with two HNO_3 and three HNO_3 were also calculated. The most stable complexes of DMAZ-$(HNO_3)_2$ and DMAZ-$(HNO_3)_3$, which are also shown in Figure 8, are 24.4 and 28.1 kcal/mol, respectively, lower in energy than the (isolated) reactants.



DMAZ-HNO_3 DMAZ-HNO_3 DMAZ-HNO_3 DMAZ-$(HNO_3)_2$ DMAZ-$(HNO_3)_3$

Figure 18. DMAZ-$(HNO_3)_x$ complexes (Bond distances in Å)

Also expected to be important in the ignition of DMAZ-RFNA systems is the abstraction of H-atoms from DMAZ by NO_2. Since DMAZ has three different nominal H-atom types; i.e., those associated with $–CH_3$ groups, those associated with the $–CH_2–$ group adjacent to an amine N-atom, and those associated with the $–CH_2–$ group adjacent to an azido group (-N_3), three different products can be postulated for the first abstraction reaction: $H_3CN(C•H_2)CH_2CH_2N_3$, $(CH_3)_2NC•HCH_2N_3$, and $(CH_3)_2NCH_2C•HN_3$. With *cis*-HONO as the other product, these reactions are:

$$\text{DMAZ} + NO_2 \rightarrow H_3CN(C•H_2)CH_2CH_2N_3 + \textit{cis}\text{-HONO} \quad \text{(R1)}$$

$$\text{DMAZ} + NO_2 \rightarrow (CH_3)_2NC•HCH_2N_3 + \textit{cis}\text{-HONO} \quad \text{(R2)}$$

$$\text{DMAZ} + NO_2 \rightarrow (CH_3)_2NCH_2C•HN_3 + \textit{cis}\text{-HONO} \quad \text{(R3)}$$

Potential energy diagrams for R1, R2, and R3 are shown in Figure 19. Based on G3MP2//MPWB1K results, $\Delta H_r(298)$ estimates for these reactions are +15.3, +12.6, and +12.3 kcal/mol, respectively. In addition, transition states for R1, R2, and R3 are, respectively, 13.6, 12.5, and 20.9 kcal/mol above the reactant asymptote. The transition states for R1 and R2 are below the product asymptote by 1.7 and 0.1 kcal/mol, respectively. Internal reaction coordinate (IRC) walks identified the presence of (equilibrium structures) PR-I, PR-II, and PR-III between the DMAZ+NO_2 reactant asymptote and transition states TS1, TS2, and TS3, respectively. G3MP2//B1K-based calculations indicate the three structures are ~1 kcal/mol lower in energy than the reactants. IRC walks also located COMP-I, COMP-II, and COMP-III between the transition states and their respective product asymptotes. G3MP2//MPWB1K-based results indicate they will be 5.2, 5.9, and 11 kcal/mol, respectively, below their respective transition state.

Rate expressions for the H-atom abstraction reactions corresponding to these paths are provided in Table 1, and Figure 20 compares the rate as a function of temperature. As shown in the figure, at low-temperatures the rates for R1 and R2 are similar, but as the temperature increases R1 becomes faster. R3 is less important because it has a relatively high-barrier (E_a).

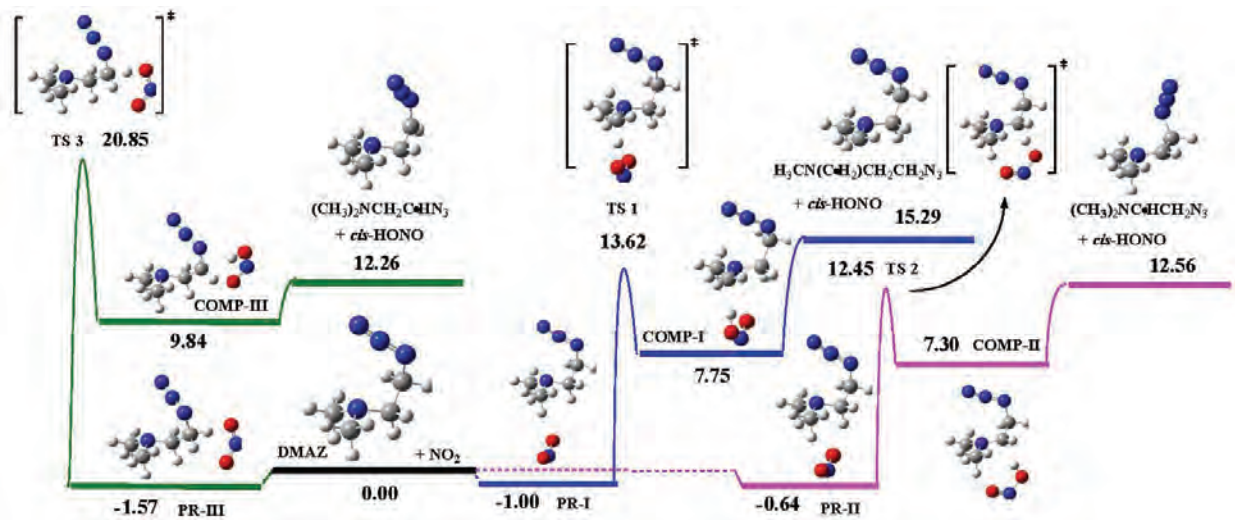


Figure 19. Potential energy diagram for H-atom abstraction from DMAZ by NO_2: G3MP2//B1K results

Table 1. Kinetic parameters[a]

	Reaction	*A* (s^{-1})	*n*	E_a (kcal/mol)	Ref.
R1	DMAZ + NO_2=$CH_3CN(C{\bullet}H_2)CH_2CH_2N_3$ + *cis*-HONO (via TS1)	3.65 x10^0	3.30964	9.3	b
R2	DMAZ + NO_2=$(CH_3)_2NC{\bullet}HCH_2N_3$ + *cis*-HONO (via TS2)	2.05 x10^2	2.99847	11.2	b
R3	DMAZ + NO_2=$(CH_3)_2NCH_2C{\bullet}HN_3$ + *cis*-HONO (via TS3)	1.13 x10^{-1}	3.99253	17.8	b

a High-pressure-limit rate parameters. Units: cm3/mol-sec and kcal/mol. k=A'Tnexp(-E_a/RT). Calculated for T=300 to 2000K;
b. <TST> Fit with three parameter modified Arrhenius equation; H tunneling effect has been included.

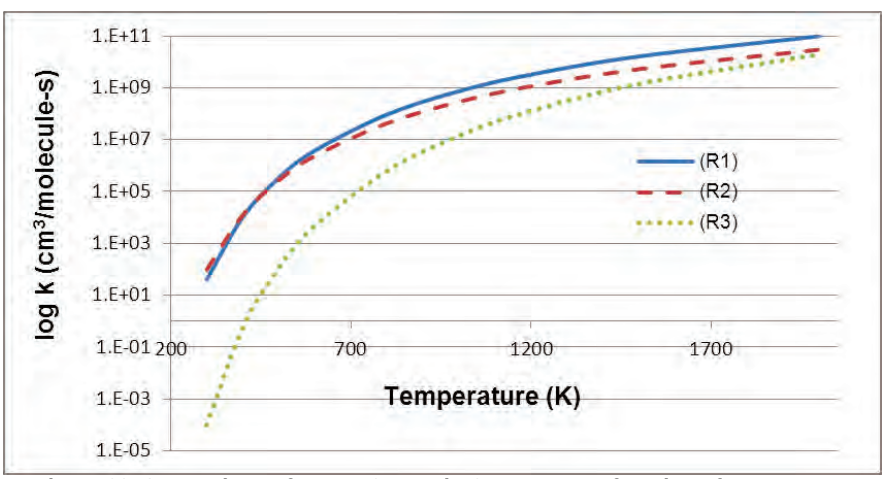


Figure 20. Comparison of R1's, R2's, and R3's rates as a function of temperature

A chemical kinetics mechanism for DMAZ-RFNA was constructed from such results. It consists of 1,391 reactions and involves 292 species. Toward establishing its validity and refining it if necessary, it was employed with a homogeneous reactor model to estimate time-to-ignition values for various initial conditions. Figure 21 shows results obtained for a mixture with an oxidizer-to-fuel ratio of 2.8 that was initially at 300K and 2,000 psi. The simulation indicates a time-to-ignition of ~6.8 msec, which is in agreement with ignition delays reported for the combination (6 ms).[28] Sensitivity analyses performed in conjunction with such simulations confirmed the time-to-ignition's primary dependence on: 1) the $\Delta H_r(T)$ of the HNO_3+DMAZ complexation reaction, and 2) the rate of H-atom abstraction from DMAZ by NO_2 (R1 and R2).

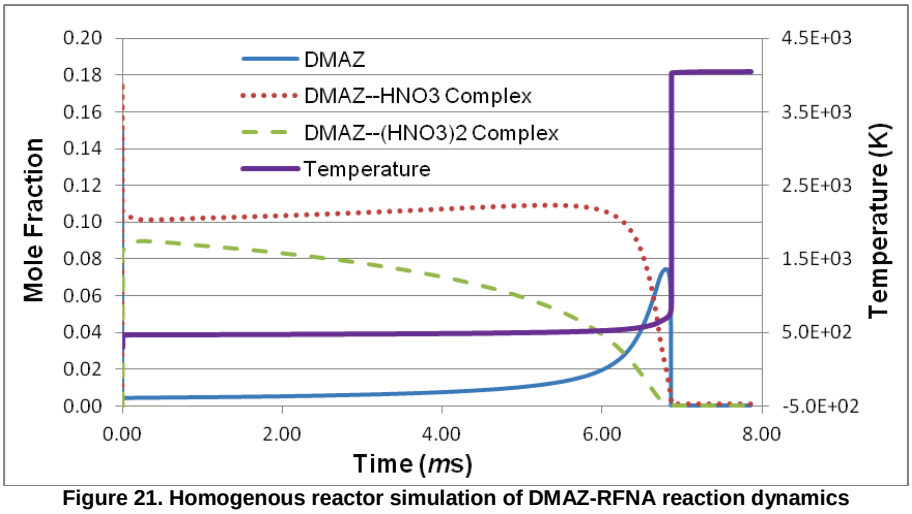


Figure 21. Homogenous reactor simulation of DMAZ-RFNA reaction dynamics

4. Summary of Accomplishments

1) The application of computational fluid dynamics for the visualization of chemically-reacting flow within the combustion chamber of the hypergolic propulsion system concept referred to as the impinging stream vortex engine (ISVE) has yielded insights into the nature of the performance of a design equipped with a pneumatically-actuated bipropellant valve (PABV) system and fueled with the hydrazine-alternative hypergol TMEDA. The simulations reveal that the build-up of combustion gases in the engine and their attendant radiative properties contribute to a gradual but continuous heating of the engine walls, despite long dwell-times between thrust-producing pulses. The simulations also indicate that that substituting TMEDA for TEDMAZ would not degrade the ISVE's performance, but higher wall-heating might be realized. Upgrades to a CFD model of an AMRDEC hybrid motor concept, including the incorporation of a finite-rate chemical kinetics mechanism for modeling HTPB-RFNA combustion, were also demonstrated and documented. An acceptable degree of concurrence between the model and test-stand results was obtained, and it was observed that hot solid-propellant particles can be a source of heat transfer to the nozzle-wall during extended engine operation.
2) 2-Azido-N,N-dimethylethanamine (DMAZ) is being employed in a fuel blend that is the current candidate of choice for the ISVE development program. As a step toward modeling the system's ignition and combustion chemistry, a detailed, finite-rate chemical kinetics mechanism for DMAZ-RFNA was developed. Computationally-based methods were employed to derive rate expressions for elementary reactions related to: (1) the thermal decomposition of DMAZ, (2) the (exothermic) complexation of DMAZ and HNO_3, (3) the abstraction of H-atoms from DMAZ by NO_2, (4) bimolecular reactions between (radical) intermediates and NO_2, and (5) unimolecular dissociations of R-NO_2 adducts. The detailed mechanism consists of 1391 reactions and involves 292 species. Comparisons of time-dependent, homogeneous reactor simulations based on the mechanism and results of ignition-delay experiments indicate the mechanism's validity for the application.

Acknowledgements

The DoD HPC Modernization Office supported this project by supplying supercomputer time under the Computing Challenge Project C4C. The computer time was made available at the DoD Supercomputing Resource Centers at AFRL and NAVO. Mr. R.S. Michaels and Mr. N. Mathis (AMRDEC) provided data and technical information concerning the ISVE. Mr. D. Thompson (AMRDEC) and Mr. Z. Wingard (ARL) provided data and technical details for the hybrid rocket motor. Dr. Anthony Kotlar (ARL) assisting in reduction of the chemical mechanisms.

References

1. Michaels, R.S. and Wilson, B.F., "The Low L/D Vortex Engine for Gel Propulsion." *Proceedings of the 1995 JANNAF Gel Propulsion Technology Symposium*, Chemical Propulsion Information Center, CPIA Publication 627, pp. 9–16, 1995.

2. Mathis, N.P., Salehi, D.M., Michaels, R.S., Turner, T.W., "Evaluation of the Army's Throttleable PABV Vortex Engine." *Proceedings of the 57th JANNAF Propulsion Meeting*, CPIAC Publication JPM-CD-09, 2010.

3. Wingard, Z.K., Thompson, D.M., and Simmons, E.L., "Tactical Army Hybrid Propulsion Development." *Proceedings of the 5th JANNAF Liquid Propulsion Subcommittee Meeting*, Chemical Propulsion Information Analysis Center, CPIAC Publication JSC CD-62, 2010.

4. Anderson, W.R., McQuaid, M.J., Nusca, M.J., and Kotlar, A.J., "A Detailed, Finite-Rate Chemical Kinetics Mechanism for Monomethylhydrazine-Red Fuming Nitric Acid, MMH-RFNA Systems." *ARL-TR-5088*, US Army Research Laboratory, Aberdeen Proving Ground, MD, 2010.

5. McQuaid, M.J., "Notional Hydrazine-Alternative Hypergols: Design Considerations, Computationally-Based Property Determinations, and Acquisition Possibilities.," *ARL-TR-3694*, US Army Research Laboratory, Aberdeen Proving Ground, MD, 2006.

6. Nusca, M.J., Mathis, N.P., and Michaels, R.S., "Computational Study of Throttling and Heat Transfer for the Army's Impinging Stream Vortex Engine." *Proceedings of the 44th JANNAF Combustion Subcommittee Meeting*, Chemical Propulsion Information Analysis Center, CPIAC Publication JSC-CD-66, April 2011.

7. Nusca, M.J., "Computational Fluid Dynamics Modeling of the Army's Hybrid Hypergolic Rocket Engine." *Proceedings of the 44th JANNAF Combustion Subcommittee Meeting*, Chemical Propulsion Information Analysis Center, CPIAC Publication JSC-CD-66, April 2011.

8. Chen, C.-C., Nusca, M.J., Kotlar, A.J., and McQuaid, M.J., "Combustion Chamber Fluid Dynamics and Hypergolic Gel Propellant Chemistry Simulations for Selectable Thrust Rocket Engines." *Proceedings of the DoD HPCMP Users Group Conference*, San Diego, CA, IEEE Computer Society, 2009.

9. Nusca, M.J., Chen C.-C., and McQuaid, M.J., "Modeling the Combustion Chamber Dynamics of Selectable-Thrust Rocket Motors." *Proceedings of the DoD HPCMP Users Group Conference*, Nashville, TN, IEEE Computer Society, 2010.

10. McQuaid, M.J., Kotlar, A.J., and Chen, C.-C., "TMEDA-RFNA and HTPB-RFNA Chemical Kinetics Mechanisms for Computational Fluid Dynamics Modeling." *Proceedings of the 6th JANNAF Liquid Propulsion Subcommittee Meeting*, Chemical Propulsion Information Analysis Center, CPIAC Publication JSC-CD-67, December 2011.

11. Nusca, M.J., "Computational Study of the Army's Gel Bipropellant Vortex Engine: Throttling, Heat Transfer and Fuel Alternatives." *Proceedings of the 8th JANNAF 8th Modeling and Simulation Subcommittee Meeting*, Chemical Propulsion Information Analysis Center, CPIAC Publication JSC-CD-67, December 2011.

12. Chen, C.-C. and McQuaid, M.J., „Tetramethylethylenediamine-Red Fuming Nitric Acid, TMEDA-RFNA Reaction Kinetics." *Proceedings of the 34th JANNAF Propellants and Explosives Development and Characterization Subcommittee Meeting*, Chemical Propulsion Information Analysis Center, August 2007.

13. Kotlar, A.J., Personal communication, US Army Research Laboratory, APG, MD, 2011.

14. Mathis, N.P., Salehi, D.M., Michaels, R.S., and Turner, T.W., "Evaluation of the Army's Throttleable PABV Vortex Engine." *Proceedings of the 57th JANNAF Propulsion Meeting*, CPIAC Publication JPM-CD-09, May 2010.

15. Nusca, M.J., "Computational Modeling of Finite-Rate HTPB-RFNA Chemistry for the Army's Hybrid Hypergolic Rocket Engine." *Proceedings of the 6th JANNAF Liquid Propulsion Subcommittee Meeting*, Chemical Propulsion Information Analysis Center, CPIAC Publication JSC-CD-67, December 2011.

16. Chen, C.-C., and McQuaid, M.J., "Thermochemistry and Kinetics Modeling of Hydroxyl-Terminated Polybutadiene-Red Fuming Nitric Acid, HTPB-RFNA Systems." *Proceedings of the 5th JANNAF Liquid Propellant Subcommittee Meeting*, Chemical Propulsion Information Analysis Center, JSC-CD-62, May 2010.

17. Zhao, Y. and Truhlar, D.G., "Hybrid Meta Density Functional Theory Methods for Thermochemistry, Thermochemical Kinetics, and Noncovalent Interactions: The MPW1B95 and MPWB1K Models and Comparative Assessments for Hydrogen Bonding and van der Waals Interactions." *Journal of Physical Chemistry A*, Vol. 108, pp. 6908–6918, 2004.

18. Montgomery, J.A., Frisch, M.J., Ochterski, J.W., and Petersson, G.A., "A Complete Basis Set Model Chemistry. VI. Use of Density Functional Geometries and Frequencies." *Journal of Chemical Physics*, Vol. 110, pp. 2822–2827, 1999.

19. Montgomery, J.A., Frisch, M.J., Ochterski, J.W., and Petersson, G.A., "A Complete Basis Set Model Chemistry. VII. Use of The Minimum Population Localization Method." *Journal of Chemical Physics*, Vol. 112, pp. 6532–6542, 2000.

20. Curtiss, L.A., Raghavachari, K., Redfern, P.C., Rassolov, V., and Pople J.A., "Gaussian-3, G3 Theory for Molecules Containing First and Second-Row Atoms." *Journal of Chemical Physics*, Vol. 109, pp. 7764–7776, 1998.

21. Curtiss, L.A., Redfern, P.C., and Raghavachari, K., "Gaussian-4 Theory." *Journal of Chemical Physics*, Vol. 126, pp. 84108–84119, 2007.

22. Curtiss, L.A., Redfern, P.C., Raghavachari, K., Rassolov, V., and Pople, J.A., "Gaussian-3 Theory Using Reduced Møller-Plesset Order." *Journal of Chemical Physics*, Vol. 110, pp. 4703–09, 1999.

23. Steinfeld, J.I., Franciso, J.S., and Hase, W.L., *Chemical Kinetics and Dynamics*, Prentice Hall, Engelwood Cliffs, NJ, 1989.

24. Da Silva, G. and Bozzelli, J.W., "Variational Analysis of the Phenyl + O_2 and Phenoxy + O Reactions." *Journal of Physical Chemistry A*, Vol. 112, pp. 3566–3575, 2008.

25. Dean, A.M., "Predictions of Pressure and Temperature Effects Upon Radical Addition and Recombination Reactions." *Journal of Physical Chemistry*, Vol. 89, pp. 4600–4608, 1985.

26. Dean, A.M., Bozzelli, J.W., Ritter, E.R., "CHEMACT: A Computer Code to Estimate Rate Constants for Chemically-Activated Reactions." *Combustion Science and Technology*, Vol. 80, pp. 63–85, 1991.

27. Chang, A.Y., Bozzelli, J.W., and Dean, A.M., "Kinetic Analysis of Complex Chemical Activation and Unimolecular Dissociation Reactions Using qRRK Theory and the Modified Strong Collision Approximation." *Zeitschrift fur Physikalische Chemie*, Vol. 214, pp. 1533–1568, 2000.

28. Thompson, D.M., *Tertiary Amine Azides in Hypergolic Liquid or Gel Fuels Propellant Systems*, United States Patent 6013143, 2000.

6.1 Basic Research

Navy

Computational Studies of Oxidation States, Polar Ferroelectric Films, and Defects

Lai Jiang, Miguel Angel Méndez Polanco, Ilya Grinberg, Sergey V. Levchenko, Alexie M. Kolpak, and Andrew M. Rappe
The Makineni Theoretical Laboratories, Department of Chemistry, University of Pennsylvania, Philadelphia, PA
{jianglai, mpolanco, ilya2, rappe}@sas.upenn.edu

Abstract

Quantum mechanical simulation is an invaluable tool for studying material properties and behaviors due to its versatility and ability to probe properties not easily accessible in experiments. Employing the computational power of high performance HPCMP supercomputers, we investigated the oxidation states of ions in insulating solids, the stabilization of highly-polar $PbTiO_3$ films, and defects in SONAR related materials. The ability to identify oxidation states guides our understanding of the behavior of the materials under investigation, providing great potential to design functional materials from an atomic level.

1. Introduction

Ferroelectric ABO_3 oxides are used in a variety of technological applications. For example, perovskite piezoelectric materials can interconvert mechanical and electrical energy, forming the core of modern US Navy SONAR devices. Ferroelectric materials are also used for hardened, non-volatile memory that is invulnerable to electromagnetic pulse radiation. There is a critical demand for improved performance such as greater sensitivity and energy conversion efficiency, higher temperature limits and lower cost. The materials with the best properties for Navy applications are perovskite oxides with parent formula ABO_3, combined into binary or ternary solid solutions with complex local structure. Recently, first-principles density functional theory has been shown to be a powerful method for studying complex oxide solid solutions, due to its accuracy and ultra-high resolution. Here, we review our recent High Performance Computing Modernization Program (HPCMP)-supported density functional theory (DFT) studies of fundamental physical properties in solids, of polarization stability in ferroelectric thin-films of the type used in non-volatile memory, and of defects commonly present in materials at the core of the next-generation Naval SONAR devices.[1,2,3]

2. Results and Discussion

2.1 Rigorous Calculation of Oxidation State in Solids

Oxidation state (OS) is a concept used in all disciplines of physical sciences to understand and predict the properties of materials. A clear knowledge of the oxidation states of the atoms in a material is often a key to connect the microscopic quantum mechanical behavior to the observed macroscopic phenomena. The original definition of OS treats all inter-atomic bonds as 100% ionic and is not feasible when covalency is involved.[4] Several theoretical treatments of OS have been proposed over the past century; however, all of these methods require parameters other than the structure of a system to determine the oxidation state.[5,6] For example, to integrate the charge density around an atom, one needs to choose what boundary condition to impose. Similarly, in order to calculate the population of the all-electron wavefunction projected onto local orbitals, the choice of a particular atomic basis must be made. These drawbacks, along with the fact that most traditional methods give non-integer oxidation states, undercut their potential to serve as a rigorous definition of integer number OS, which would agree with the original charge transfer definition.

To gain a full understanding of OS in insulating solids, we proposed and tested a wave-function topology-based framework in which the atomic oxidation state is unique for a given structure and is guaranteed to be an integer. This rigorously reconciles the quantum mechanical nature of electrons with the simple ionic model. Our method can be applied to various functional materials (both oxides and other ceramics) that are of interest for application in military operations.

Our approach to defining a rigorous integer OS is based on the fact that a periodically evolving Hamiltonian results in quantized charge pumping.[7] According to the modern theory of polarization, such a charge quantization can be further conceptualized as the transfer of an integer number of Wannier centers (Figure 1).[8] To use this fundamental quantization, we choose to obtain a periodic Hamiltonian by moving an atom and all of its images by a lattice vector **R**, under periodic boundary conditions. We proved that such a sub-lattice displacement results in $\Delta\vec{P} = \frac{eN\vec{R}}{V}$, a change in polarization that is along the displacement vector **R** and has the magnitude of the change in dipole moment per volume when an integer-charged (Q=eN) particle is moved by the same distance **R**. The OS N therefore can be easily deduced by $N = \frac{V}{e}\frac{\Delta\vec{P}\cdot\vec{R}}{\vec{R}^2}$.

We further proved that the OS N for any displacement **R** is the same. Our definition of integer OS is therefore a function of only the species that is being moved, and the initial chemical environment of the atom.

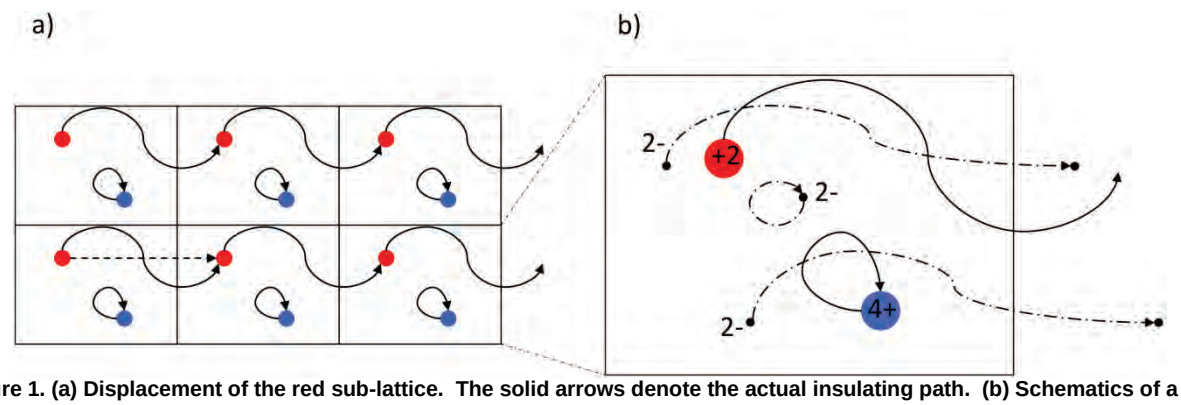


Figure 1. (a) Displacement of the red sub-lattice. The solid arrows denote the actual insulating path. (b) Schematics of a few Wannier centers moving with the nuclear motion.

To demonstrate the applicability of our definition, we calculate the OS of atoms in materials ranging from the simple H_2O and LiH to the complicated $BaBiO_3$ and Sr_2FeWO_6. The first two systems are calculated using the *Abinit* DFT package[9,10] with the generalized gradient approximation to the exchange-correlation functional, and the last two using the *Quantum Espresso* DFT package[11] with the local density approximation to the exchange-correlation functional. Both packages use the optimized norm-conserving nonlocal planewave pseudopotential method, with pseudopotentials generated from the OPIUM code[36].

The results for H_2O and LiH are shown in Figure 2. Our method successfully predicts that in H_2O, H is +1 and O is −2, and in LiH, Li is +1 and H is −1. Also shown is the component of **ΔP** perpendicular to the displacement vector **R**, which is zero in all cases. This agrees with our statement that **ΔP** is parallel to **R**. The comparison between H^+ in H_2O and H^- in LiH shows that the OS in our definition is a function of the chemical environment, because the same H pseudo-potential is used in both cases.

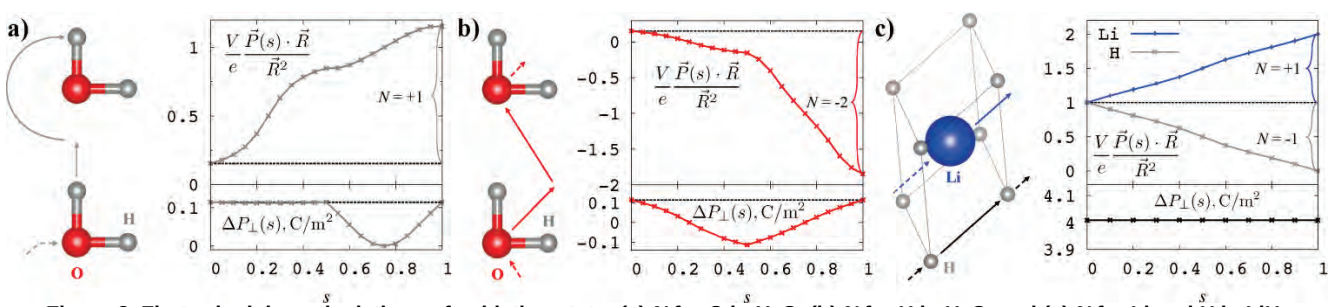


Figure 2. First-principles calculations of oxidation state: (a) *N* for O in H_2O, (b) *N* for H in H_2O, and (c) *N* for Li and H in LiH

To illustrate the relationship between OS and Wannier center motions, we plotted the maximally-localized Wannier center trajectories for the H_2O example above using the *Wannier90* package.[13] The results indicate that when O is moving, all four plotted Wannier centers move closely with the O atom to the cell above the original one, but when H is moving, Wannier centers merely shift around within the original cell and do not accumulate net displacement. This can be explained by the fact that the plotted Wannier centers are O $2p_x$ and sp^2 orbitals, which should belong to the O and therefore move with it.

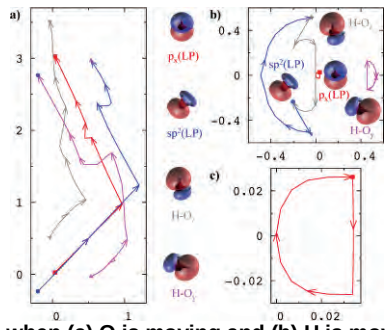


Figure 3. Wannier center trajectories for H_2O when (a) O is moving and (b) H is moving; (c) is the close-up of p_x trajectory in (b). The units are in Å.

To test the validity of our definition in more complex and challenging systems, we calculated B-site cation oxidation states in two perovskite systems, $BaBiO_3$ and Sr_2FeWO_6. $BaBiO_3$ undergoes a charge disproportionation via oxygen breathing mode,[14] where neighboring Bi ions are enclosed by O_6 cages of different sizes. As shown in Figure 4(a), we can identify the Bi in the larger cage as Bi^{3+} and the one in the smaller cage as Bi^{5+}. This result agrees well with electrostatics, since Bi^{5+} should attract anions more strongly. This also provides an example in which we use our OS definition to distinguish the same element in the same material but in different local environments.

In Figure 4(b), we calculated the OS for Fe in Sr_2FeWO_6. In this antiferromagnetic double perovskite,[15] *B*-site transition metals can be either $Fe^{2+}W^{6+}$ or $Fe^{3+}W^{5+}$. Unlike the $BaBiO_3$ case, there is no distinctive structural feature that can help identify the actual charge state of Fe. By using our method, however, we unambiguously obtained +2 as the OS of Fe, supporting early Mossbauer spectroscopy observations.[16]

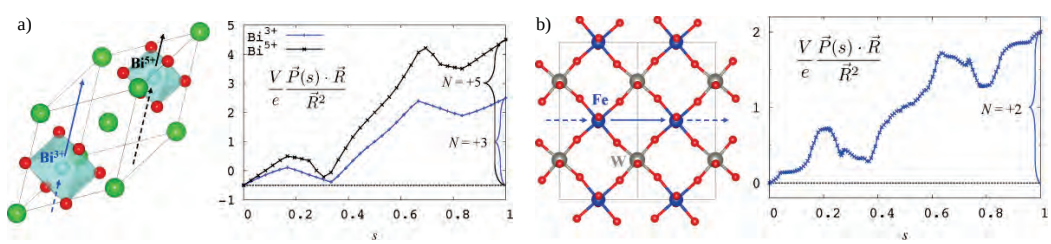


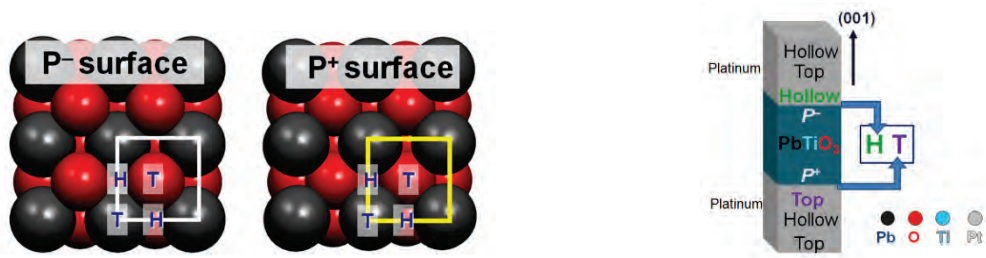
Figure 4. Oxidation state (a) *N* for two inequivalent Bi's in $BaBiO_3$ and (b) N for Fe in Sr_2FeWO_6

2.2 Polarization Stability in Ultra-thin Ferroelectric Films

Among the various technological applications of perovskite-based devices, high-density storage is of great interest due to its key role for the advancing information technology. Ferroelectric-based memory devices, known as Ferroelectric Random Access Memories (FeRAM), possess the advantages of fast read/write of data and electric-efficiency compared to the commonly used flash memories.[17] Further understanding of the properties of FeRAMs as both integrated devices and their individual components, (i.e., the electronic properties of the capacitor and the effects of particular capacitor settings on the ferroelectric properties of the oxide layer) would be essential for increasing reliability and efficiency of such devices. Progressive miniaturization of FE devices is challenging due to the inherent presence of a depolarizing field (ε_{dep}) in ultra-thin films, which can eventually suppress its polarization completely. Thus, finding ways to stabilize polarization in the ultra-thin film regime has been the subject of multiple experimental and theoretical research efforts.[18–27,17,28–30] However, none of the previous investigations have considered the consequences of the lateral changes at the metal-oxide interface on ferroelectricity. Using DFT calculations, we studied the role of the interfacial atomic arrangement in the stability of polarization in nanoscale Pt/$PbTiO_3$/Pt capacitors. First, we studied geometries constrained to have bulk-like in-plane lattice parameters, and later we allowed for relaxation of all degrees-of-freedom, e.g., in-plane symmetry broken

symmetry structures. We compared the structure and energy of polar films with respect to the paraelectric (PE) configurations for four different interface arrangements: Top/P^-:::P^+/Top; Hollow/P^-::: P^+/Hollow, Top/P^-::: P^+/Hollow; Hollow/P^-::: P^+/Top, that hereafter we name TT, HH, TH, and HT, respectively.

All our calculations were carried out using the *Quantum Espresso*[31] DFT package with LDA and PBEsol exchange-correlation functionals[32,33]; both give the same qualitative results indicating that our findings are functional-independent. We sampled the Brillouin zone with a Monkhorst-Pack grid[34] of 4×4×1 k-points in our slab geometries that included at least 18 Å of vacuum between images and dipole correction in the center of the vacuum region.[35] We represented the atoms using norm-conserving optimized pseudo-potentials[12] generated from the OPIUM code[36] with a plane-wave cut-off of 25 Ha. The geometries were optimized at the $PbTiO_3$ (PTO) theoretical lattice constant by the BFGS minimization scheme until a maximum force was converged to less than 0.01eV/Å on each atom. The capacitors in our study are PTO films between Pt (001) electrodes, nPt/PbO-(TiO_2-PbO)$_m$/nPt, with varying n and m. The structures are named using the following convention: Pt layers are absorbed on both sides of the PTO slab in Top or Hollow sites: T and H, respectively [Figure 5(a)]. We first refer to the Pt layer registry on the P^- interface, and later the P^+ interface. Further Pt layers allow an fcc (001) stacking, [Figure 5(b)].

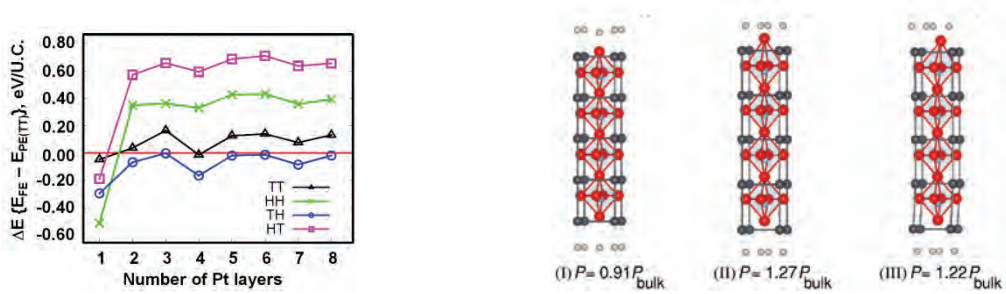


(a) (b)

Figure 5. (a) Schematics of the adsorption sites for Positive and Negative $PbTiO_3$ (PTO) surfaces. (b) Nomenclature convention for the systems studied here, i.e., (Hollow Top) HT describes the system with the following interfaces and Pt stacking: HTH/P^-–(TiO_2-PbO)–P^+/THT. Color scheme: Pb, dark grey; Ti, cyan; O, red.

A polarized oxide creates distinct chemical environments at both (P^+ and P^-) oxide/electrode interfaces, giving rise to a difference in the work functions at the two surfaces of the FE slab, crucial for the screening of the P surface charges.[20,29]

In-plane symmetry constrained films. The geometry at the interface has a strong impact on the energetics and P stability of the capacitors. TT is the lowest in energy for non-polar (paraelectric, PE) structures. However, for the polarized film, TH is lower in energy than the PE-TT system. The stabilization depends on the thickness (n) of the electrode but, on average, TH is 120 meV/PTO–unit-cell more stable than the PE structure [Figure 6(a)]. We observed an oscillation period in ΔE that is similar to Friedel oscillations in Pt (100) surfaces,[37] but much stronger due to the presence of the charged surface of the polarized oxide. We found HH and HT to be only metastable configurations that show significantly higher energies than TT and TH.



(a) (b)

Figure 6. (a) Stabilization Energy (ΔE) as a function of the number of Pt layers, for capacitor configurations with in-plane symmetry preserved and LDA functional (b) Fully-relaxed four-unit cell Pt/PTO/Pt films and its corresponding P, estimated from Ti-displacements compared to bulk values, using the PBEsol functional. (I) TT_{sym}, (II) HH_{sym} and (III) HH_{brk}. For clarity, only the interface Pt layer is shown.

The interface structure helps to explain the stability of capacitors. For the T geometry, half of the Pt atoms bond to the surface O atoms and half of Pt atoms bond to the surface Pb atoms. In contrast, in the H geometry, every Pt sits between two O atoms and two Pb atoms. For the Top/P^+ interface, the large Pt–O distances (≈3Å) are evidence of the breaking of the Pt-O bond; hence, only half of the Pt atoms are bound to the PTO surface. However, the H geometry enables full oxide-electrode interaction through Pt-Pb bonds and stabilizes the TH system relative to the TT system.

Fully-relaxed films. To investigate further the role of the interface local structure, we allowed full-relaxation of all degrees of freedom in the capacitors. This breaks in-plane symmetry at the interface and lowers the energy of polar oxide structures. The stabilization is particularly large for HH, making it the lowest energy configuration, $\Delta E^{HH}(_{sym-brk})\approx-$250meV/PTO-u.c and $\Delta E(_{THsym-HHbrk})\approx-20$ meV/PTO-u.c. This is in contrast with the TH configuration for which symmetry breaking only stabilizes the capacitor in ≈10meV/PTO-u.c.

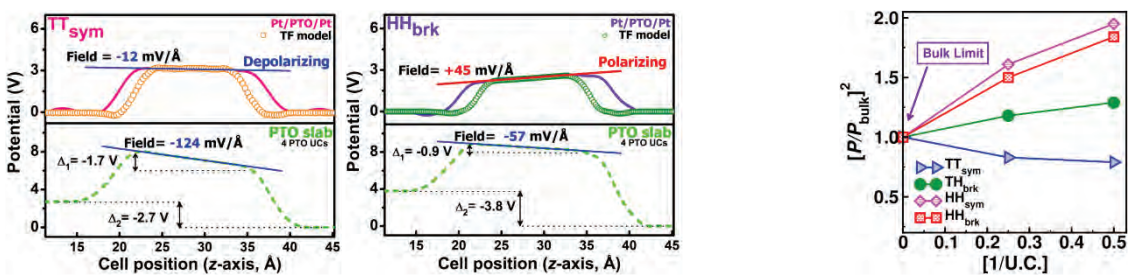
Stabilization is due to deformations occurring at the oxide-metal interfaces. [Figure 6(b)] For the P^- surface, the interfacial oxygen tilts ≈8°. In the P^+ surface, Pb slides sideways in the *xy*-plane in order to minimize the Pb-O (decreases ≈0.3Å), while the oxygen-tilt is only ≈2.5°. Energy gains from the corresponding symmetric to asymmetric structures indicate that the oxygen-tilting occurring at the P^- surface makes up for the largest contribution to the stabilization of broken-symmetry registries.

A net polarizing-field in the oxide enhances *P* in the capacitors. If the work function difference between the P^- and P^+ surfaces of the free-standing slab (Δ_2) is greater than the voltage drop inside the oxide due to *P* (Δ_1), the screening of Δ_2 by the electrodes implies an over-screening of Δ_1, and this can then result in a net-polarizing (Equation 1). We observe this in HH_{brk}, by the flip in the slope of the macroscopic averaged electrostatic potential (EP). [Figure 7(a)] For TT_{sym}, however, the depolarizing field is reduced ≈90% but not completely overcome, agreeing with previous investigations.[20]

We propose a standard depolarizing field equation that highlights the importance of including the work function difference between the surfaces of a polarized FE in the screening model. The electric field inside the FE slab can be calculated as:

$$\varepsilon_{FE} = \frac{P}{\varepsilon_0}\left(\frac{2\lambda}{d}\right) + \frac{(\Delta_2 - \Delta_1)}{d}\left(1 - \frac{2\lambda}{d}\right) \tag{1}$$

where (Δ_2–Δ_1) is the difference in the surface electrostatic potential drops at the P^- and P+ surfaces, *d* is the length of the ferroelectric slab, λ is the screening length due to electronic screening only and *P* is the average polarization of the film. We use EPs from our DFT calculations of frozen bare-oxide slabs to find λ values using a Thomas-Fermi (TF)-like model and our generalized equation. We find that HH systems have rather larger λ values. This should lead to larger ε_{dep}, according to the classic model and opposite to the strong polarizing field found by DFT calculations; therefore highlighting the importance of the distinct work functions at the interfaces.



(a) (b)

Figure 7. (a) Macroscopic averaged electrostatic potential in the PTO film of the capacitors (from DFT and Thomas-Fermi modeling) and the corresponding freestanding slab for TT_{sym} and HH_{brk}. *P* direction is ($P^+ \leftarrow P^-$) in the plots. (b) Relationship between P^2 and the inverse of PTO film thickness (1/*d*). Except for TT, *P* increases with decreasing film thickness, indicating the absence of a critical size for *P* enhancement.

The EPs from our TF model agree with those found by DFT [open symbols Figure 7(a)] implying that the main effect of the interface changes on the depolarizing field is due to changes in (Δ_2–Δ_1). This difference is created by layer rumplings [$\delta_{Pb}(z)$–$\delta_O(z)$ in this case]; consequently, the largest effect will be present in interfaces with larger surface rumplings in the FE.

There is no critical size limit for the *P* enhancement in TH and HH related capacitors. Figure 7(b) shows that *P* varies inversely to the film thickness, increasing for smaller films. This means that for carefully constructed metal-electrode interface, the depolarizing field is completely eliminated and will not prevent further miniaturization of ferroelectric devices.

2.3 A Study of Oxygen-vacancy Defects in Heterovalent Ferroelectrics

Extrinsic effects are crucial to fatigue in SONAR and NVRAM devices—the degradation of piezoelectric performance and polarization switchability over time. Fatigue is currently thought to be caused by oxygen-vacancy pinning of 180° and 90° domain-walls. Oxygen vacancies are the most common type of defect in perovskite ferroelectrics, and clustering of vacancies at the domain wall and the consequent pinning of the domain walls decrease the extrinsic contribution to the piezoelectric response.

Since fatigue is related to the gradual accumulation of oxygen vacancies over time, the time until performance degradation should be closely related to the rate of O-vacancy migration through the perovskite. The oxygen-vacancy diffusion rate also plays an important role in the solid oxide fuel cell (SOFC) devices, where a faster O-vacancy transport is favorable, and allows the SOFC to be operated at lower temperatures. This is in contrast to the case in SONAR devices, where a slow diffusion rate is favorable.

Currently, vacancy energetics, migration pathways and interactions of vacancies with ferroelectric domain walls are poorly understood, due to the difficulties of studying these extrinsic phenomena. While oxygen-vacancy diffusion kinetics has been studied for the SOFC materials,[38] there have been very few *ab initio* studies of how the local chemistry in complex solutions controls O-vacancy diffusion rate and the accumulation and clustering of vacancies.

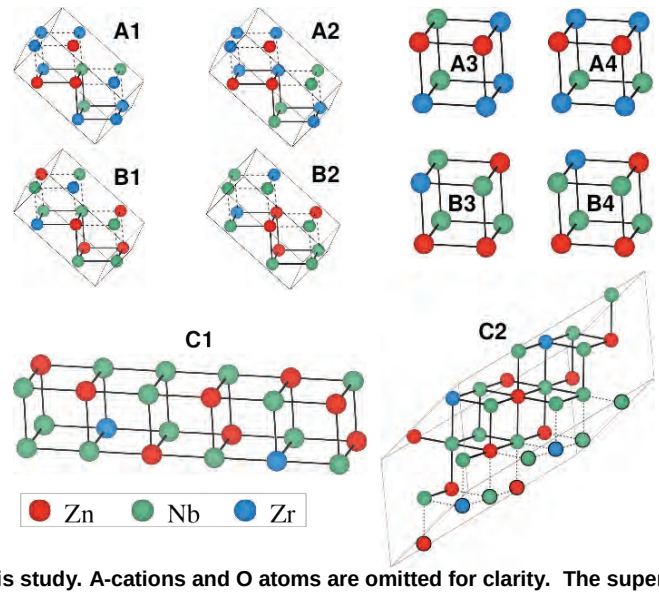


Figure 8. Supercells used in this study. A-cations and O atoms are omitted for clarity. The supercell dimensions for A1, A2, B1, and B2 are $\sqrt{2}\times2\times2\sqrt{2}$, and 2×2×2 for A3, A4, B3, and B4. C1 and C2 supercells are 6×2×2 and $\sqrt{6}\times2\sqrt{2}\times4$, respectively.

Ab initio calculations are ideally suited for the studies of vacancies due to their combination of high-accuracy and precise knowledge of the calculated system. For vacancies in the heterovalent perovskites, such as the ferroelectric solid solutions based on $Pb(Mg_{1/3}Nb_{2/3})O_3$ (PMN) and $Pb(Zn_{1/3}Nb_{2/3})O_3$ (PZN) that are used in the current state-of-the-art SONAR, even the basic energetics of vacancy locations are not trivial. Unlike the case of the simple perovskites such as $PbTiO_3$ or even the more complicated but still isovalent $Pb(Zr,Ti)O_3$ solid solution, the variation in the valence on the *B*-site in PMN- and PZN-based materials can lead to strong preference for a particular pair of *B*-cations neighbors of the vacancy.

We have examined the energetics of vacancies in $(1\text{-}x)Pb(Mg_{1/3}Nb_{2/3})O_3$–$xPbZrO_3$ (PMN-PZ) and $(1\text{-}x)Pb(Zn_{1/3}Nb_{2/3})O_3$–$xPbZrO_3$ (PZN-PZ) solid-solutions to elucidate how the chemistry of the solid solution can control vacancy diffusion behavior and its impact on properties. We choose PMN-PZ and PZN-PZ because experimentally in PMN-PZ high-content of vacancies can be achieved, enabling the comparison of our calculations to the experimental data.[39] We expect that the results obtained for PMN-PZ and PZN-PZ will be general to all heterovalent Pb-based perovskites.

To understand the trends in the DFT obtained energies, we analyze the results using crystal chemical concepts that have previously been shown to be important for local structure in ferroelectric perovskites. Our previous research on the relationship between composition, local structure and properties in ferroelectric ABO_3 has shown that ionic size (R_0), ionic valence (V_0) and displacement preference (D_0) are the key parameters.[40,41] While R_0 and V_0 are intuitive, the

displacement preference D_0, defined as the average magnitude of cation off-center displacement from the center of its oxygen cage in tetragonal ferroelectrics, is specific to ferroelectrics and is related to the ability of the cation to form covalent bonds with oxygen atoms. Simple metal cations, such as Mg, have small D_0 (0.08 Å) as they make ionic bonds to the O anions, which favor the symmetric structure. Electronegative 3*d* transition metal cations such as Ti and Zn can make short, covalent cation-oxygen bonds due to second-order Jahn-Teller distortion mechanism and have high D_0 values (0.25 Å). Other ions, like the 4*d* transition metal Nb (0.17 Å) and Zr (0.13 Å) are somewhere in the middle, due to the presence of the *d*-states and the resulting possibility of the second-order Jahn-Teller distortion on the one hand, but a smaller electronegativity relative to the 3*d* metals on the other hand. Covalency, as characterized by D_0, plays an important role in controlling relaxor behavior and compositional phase boundaries crucial for the excellent piezoelectric properties exhibited by the PMN-based SONAR materials. We therefore examine our results to see if we can find trends in the vacancy energetics with respect to these parameters.

For the supercells examined here, we find that V_0 and D_0 play the crucial role. Figure 9 shows that energy differences between the PZN-PZ supercells follow a trend opposite to that of the D_0 magnitude of the two *B*-cation neighbors of the vacancy. Higher D_0 magnitudes lead to lower energies due to the increased ability of the *B*-cation to compensate for a loss of an O neighbor by making stronger bonds with the rest of the O atoms in its cage.

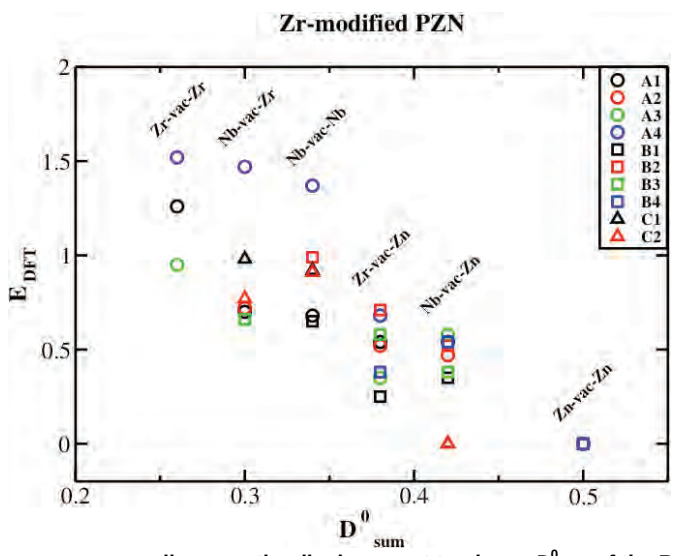


Figure 9. Vacancy location energy per supercell versus the displacement tendency D^0_{sum} of the B-cation neighbors in Zr-modified PZN. Please note that the plotted energy represents the difference in energy of each structure relative to the lowest energy structure within the same cation ordering. A clear inverse trend is apparent with a larger D^0_{sum} corresponding to a lower DFT-calculated energy difference.

Examination of PMN-PZ data shows that valence is also important. Despite its small D_0 value, the Mg-Vac-Mg configuration is the lowest energy configuration for all supercells. In contrast, the highest energy configurations always have the high-valent Zr or Nb atoms surrounding the vacancy. We assign this to the effect of valence on vacancy energetics. A vacancy placed between two *B*-cations of low-valence disrupts fewer bonds than a vacancy placed between two high-valence *B*-cations, and therefore is lower in energy. Thus, the energy of vacancy location is governed by a combination of the amount of bonds lost due to the presence of the vacancy (governed by V_0), and by the ability to compensate for this by making stronger bonds to the rest of the O neighbors (governed by D_0).

The large energy differences between the different vacancy positions indicate that vacancy diffusion in FE perovskites should be sensitive to *B*-cation ordering. Large energy differences between two vacancy positions suggest that vacancies can be trapped at sites with low-vacancy energies, such as Mg-Vac-Mg and Mg-Vac-Nb in PMN-based materials or Zn-Vac-Zn and Zn-Vac-Nb in PZN-based materials. This suggests that vacancy diffusion will be smallest and fatigue properties most favorable for ferroelectric perovskites with disordered *B*-cation arrangements, as *B*-cation disorder promotes large energy differences between vacancy positions.

3. Conclusion

We have developed a rigorous integer OS definition, based on the change in polarization upon a displacement of the atom in interest. A suite of test systems is employed to demonstrate that our method not only reproduces obvious and common results, but also is able to resolve ambiguities where traditional methods fall short. This method is solely dependent on the wave-function topology of the system. This OS calculation approach opens doors for further investigation of electronic properties, as well as providing accurate results that can facilitate the understanding of the behavior of functional materials employed in military applications.

In our study of ultra-thin ferroelectric films, we have shown the critical importance of the interface arrangement for the polarization stability of ultra-thin perovskite capacitors, and described the two necessary criteria to attain it. First, the electrode must screen the surface charges well enough to remove the depolarizing field, obtained through large (Δ_2–Δ_1) values. Second, the bonding energy of the metal-oxide interface for the ferroelectric state must be equal to or greater than the metal-oxide bonding energy of the paraelectric state, achieved by in-plane symmetry breaking and ensuring the structure is a global minimum in the potential energy surface.

We have examined the energetics of vacancies in PMN-PZ and PMN-PZ solid-solutions to elucidate how the chemistry of the solid-solution can control vacancy diffusion behavior and its impact on properties. Oxygen-vacancies were found to prefer locations between two B-cations, with a lower combined valence and a larger covalency. Cations that meet these criteria show the largest capacity for accommodating a vacancy. This suggests that increased disorder in the B-cation arrangement will trap the oxygen vacancies in their preferred location, hampering their migration and preventing fatigue.

4. Significance to DoD

Perovskite oxides are used extensively in modern Naval SONAR devices, and the relaxor ferroelectric perovskites such as $PbMg_{1/3}Nb_{2/3}O_3$-$PbTiO_3$ will be used in SONAR devices of the next decade. Perovskite oxides are also used in high-density, robust non-volatile memory. Our development of new analysis methodology applicable to solids will support the design of more sensitive materials with enhanced performance. Our studies of ultra-thin films show that a careful construction of an epitaxial metal-electrode interface enables robust ferroelectric non-volatile memory devices of extremely small dimensions. Our studies of extrinsic effects in ferroelectrics such as oxygen vacancy energetics will facilitate the design of SONAR materials with lower energy losses and longer life-times.

Systems Used

The work presented here was performed using the Navy's Cray XT5 and IBM Power 6, the US Army Engineer Research and Development Center's Cray SGI ICE, the US Air Force Research Laboratory's Cray XE6 and the DoD HPCMP Open Research System's Cray XE6. Both DFT codes use an iterative diagonalization procedure to solve the Kohn-Sham that relies heavily on BLAS, LAPACK and FFTW routines. Matrix diagonalization and FFT are parallelized, but the scalability is usually reduced for more than a hundred processes, requiring fast interconnections between cores and high-throughput I/O operations. K-point parallelization in the first Brillouin zone is highly-scalable (almost linear) and vital in reducing wall-time with minimal cross-process communications, thanks to the MPI implementation.

Computational Technology Area

Computational Material Science

Acknowledgements

This work was supported by the US AFOSR under Grant No. FA9550-10-1-0248, by the US DoE BES under Grants No. DE-FG02-07ER15920, and No. DE-FG02-07ER46431; and by the US ONR under Grants No. N00014-09-1-0157, No. N00014-11-1-0578, and No. N00014-11-1-0664. Computational support was provided by US DoD HPCMP and a DURIP grant.

References

1. Jiang, L., Levchenko, S.V., and Rappe, A.M., *Phys. Rev. Lett.*, 108, 166403, April 2012.

2. Méndez Polanco, M.A., Grinberg, I., Kolpak, A.M., Levchenko, S.V., Pynn, C., and Rappe, A.M., *Phys. Rev. B*, 85, 214107, June 2012.

3. Qi, T., Curnan, M.T., Kim, S., Bennet, J.W., Grinberg, I., and Rappe, A.M., *Phys. Rev. B.*, 84, 245206–1–6, 2011.

4. Jørgensen, C.K., *Oxidation Numbers and Oxidation States*, Springer-Verlag New York Inc., 1969.

5. Löwdin, P.O., *J. Chem. Phys.*, 18, 365, 1950.

6. Mulliken, R.S., *J. Chem. Phys.*, 23, 1833, 1955.

7. Thouless, D.J., *Phys. Rev. B*, 27, pp. 6083–6087, 1983.

8. Rabe, K.M., Ahn, C.H., and Triscone, J.-M., *Physics of Ferroelectrics: A Modern Persp*ective, Springer-Verlag, 2007.

9. Gonze, X., Amadon, B., Anglade, P.-M., Beuken, J.-M., Bottin, F., Boulanger, P., Bruneval, F., Caliste, D., Caracas, R., Cote, M., Deutsch, T., Genovese, L., Ghosez, P., Giantomassi, M., Goedecker, S., Hamann, D., Hermet, P., Jollet, F., Jomard, G., Leroux, S., Mancini, M., Mazevet, S., Oliveira, M., Onida, G., Pouillon, Y., Rangel, T., Rignanese, G.-M., Sangalli, D., Shaltaf, R., Torrent, M., Verstraete, M., Zerah, G., Gonze, J. Z. X., Amadon, B., Anglade, P.-M., Beuken, J.-M., Bottin, F., Boulanger, P., and' D. Caliste, F. B., Caracas, R., Cote, M., Deutsch, T., Genovese, L., Ghosez, P., Giantomassi, M., Goedecker, S., Hamann, D., Hermet, P., Jollet, F., Jomard, G., Leroux, S., Mancini, M., Mazevet, S., Oliveira, M., Onida, G., Pouillon, Y., Rangel, T., Rignanese, G.-M., Sangalli, D., Shaltaf, R., Torrent, M., Verstraete, M., Zerah, G., Gonze, J. Z. X., Amadon, B., Anglade, P.-M., Beuken, J.-M., Bottin, F., Boulanger, P., and' D. Caliste, F. B., Caracas, R., Cote, M., Deutsch, T., Genovese, L., Ghosez, P., Giantomassi, M., Goedecker, S., Hamann, D., Hermet, P., Jollet, F., Jomard, G., Leroux, S., Mancini, M., Mazevet, S., Oliveira, M., Onida, G., Pouillon, Y., Rangel, T., Rignanese, G.-M., Sangalli, D., Shaltaf, R., Torrent, M., Verstraete, M., Zerah, G., and Zwanziger, *J. Computer Phys. Commun.*, 180, 2582, 2009.

10. Gonze, X., Rignanese, G.-M., Verstraete, M., Beuken, J.-M., Pouillon, Y., Caracas, R., Jollet, F., Torrent, M., Zerah, G., Mikami, M., Ghosez, P., Veithen, M., Raty, J.-Y., Olevano, V., Bruneval, F., Reining, L., Godby, R., Onida, G., Hamann, D., and Allan, D., *Zeit. Kristallogr.*, 220, 558, 2005.

11. Giannozzi, P., Baroni, S., Bonini, N., Calandra, M., Car, R., Cavazzoni, C., Ceresoli, D., Chiarotti, G.L., Cococcioni, M., Dabo, I., Corso, A.D., de Gironcoli, S., Fabris, S., Fratesi, G., Gebauer, R., Gerstmann, U., Gougoussis, C., Kokalj, A., Lazzeri, M., Martin-Samos, L., Marzari, N., Mauri, F., Mazzarello, R., Paolini, S., Pasquarello, A., Paulatto, L., Sbraccia, C., Scandolo, S., Sclauzero, G., Seitsonen, A.P., Smogunov, A., Umari, P., and Wentzcovitch, R.M., *J. Phys.: Condens. Matter*, 21, pp. 395502–20, 2009.

12. Rappe, A.M., Rabe, K.M., Kaxiras, E., and Joannopoulos, J.D., *Phys. Rev. B Rapid Comm*., 41, pp. 1227–30, 1990.

13. Mostofi, A.A., Yates, J.R., Lee, Y.-S., Souza, I., Vanderbilt, D., and Marzari, N., *Comput. Phys. Commun.*, 178, 685, 2008.

14. Thonhauser, T. and Rabe, K.M., *Phys. Rev. B*, 73, 212106 1–4, 2006.

15. Kawanaka, H., Hase, I., Toyama, S., and Nishihara, Y., *J. Phys. Soc. Jpn.*, 68, 2890, 1999.

16. Kawanaka, H., *Physica B: Condensed Matt.*, 281-282, pp. 518–520, 2000.

17. Umeno, Y., Albina, J.M., Meyer, B., and Elsässer, C., *Phys. Rev. B*, 80(20), 205122, Nov 2009.

18. Junquera, J. and Ghosez, P., *Nature*,422, pp. 506–9, 2003.

19. Ederer, C. and Spaldin, N.A., *Phys. Rev. Lett.*, 95, 257601–1–4, 2005.

20. Sai, N., Kolpak, A M., and Rappe, A.M., *Phys. Rev. B Rapid Comm.*, 72, 020101(R)–1–4, 2005.

21. Umeno, Y., Meyer, B., Elsässer, C., and Gumbsch, P., *Phys. Rev. B*, 74, 060101R–1–4, 2006.

22. Gerra, G., Tagantsev, A.K., Setter, N., and Parlinski, K., *Phys. Rev. Lett.*, 96, 107603–1–4, 2006.

23. Kolpak, A. M., Sai, N., and Rappe, A.M., *Phys. Rev. B*, 74, 054112–1–5, 2006.

24. Asthagiri, A. and Sholl, D.S., *Phys. Rev. B*, 73, 125432(1)–125432(1), 2006.

25. Spanier, J.E., Kolpak, A.M., Urban, J.J., Grinberg, I., Ouyang, L., Yun, W.S., Rappe, A.M., and Park, H., *Nano Lett.*, 6, pp. 735–9, 2006.

26. Gerra, G., Tagantsev, A.K., and Setter, N., *Phys. Rev. Lett.*, 98(20), 207601, May 2007.

27. Tenne, D.A., Turner, P., Schmidt, J.D., Biegalski, M., Li, Y.L., Chen, L. Q., Soukiassian, A., Trolier-McKinstry, S., Schlom, D.G., Xi, X.X., Fong, D.D., Fuoss, P.H., Eastman, J.A., Stephenson, G.B., Thompson, C., and Streiffer, S.K., *Phys. Rev. Lett.*, 103(17), 177601, October 2009.

28. Stengel, M., Vanderbilt, D., and Spaldin, N., *Nat. Mater*., 8, pp. 392–397, 2009.

29. Stengel, M., Vanderbilt, D., and Spaldin, N.A., *Phys. Rev. B*, 80(22), 224110, December 2009.

30. Wadati, H., Hawthorn, D.G., Geck, J., Higuchi, T., Hikita, Y., Hwang, H.Y., Kourkoutis, L.F., Muller, D.A., Huang, S.-W., Huang, D.J., Lin, H.-J., Schussler-Langeheine, C., Wu, H.-H., Schierle, E., Weschke, E., Ingle, N.J.C., and Sawatzky, G.A., *Journal of Applied Physics*, 106(8), 083705, 2009.

31. Giannozzi, P., Baroni, S., Bonini, N., Calandra, M., et al., *J. Phys.: Condens. Matter*, 21, pp. 395502–20, 2009.

32. Perdew, J.P. and Zunger, A., *Phys. Rev. B*, 23, pp. 5048–79, 1981.

33. Perdew, J., Ruzsinszky, A., Gábor, C., Vydrov, O., Scuseria, G., Constantin, L., Zhou, X., and Burke, K., *Phys. Rev. Lett.*, 100, 136406–1–4, 2008.

34. Monkhorst, H.J. and Pack, J.D., *Phys. Rev. B*, 13, pp. 5188–5192, 1976.

35. Bengtsson, L., *Phys. Rev. B*, 59, pp. 12301–4, 1999.

36. http://opium.sourceforge.net.

37. Fiorentini, V. and Methfessel, M., *Journal of Physics: Condensed Matter*, 8(36), 6525, 1996.

38. S. Stølen, E.B. and Mohn, C.E., *Physical Chemistry Chemical Physics*, 8, pp. 429–47, 2006.

39. Zhao, X., Qu, W., and Tan, X., *J. Am. Ceram. Soc.*, 91, pp. 3031–8, 2008.

40. Grinberg, I., Suchomel, M.R., Davies, P.K., and Rappe, A.M., *J. Appl. Phys.*, 98, 094111–1–10, 2005.

41. Grinberg, I., Juhás, P., Davies, P.K., and Rappe, A.M., *Phys. Rev. Lett.*, 99, 267603–1–4, 2007.

Direct Numerical Simulation of Stably Stratified Turbulence at High Reynolds Number and Low Froude Number

S. Almalkie and S.M. de Bruyn Kops
Department of Mechanical and Industrial Engineering, University of Massachusetts Amherst, Amherst, Massachusetts
debk@acad.umass.edu

Abstract

Direct numerical simulations with up to 4,096×4,096×2,048 grid-points are used to study the dynamics of kinetic energy in forced, homogeneous and axisymmetric, stably-stratified flow with unity Prandtl number. No mean shear or internal waves are introduced into the flows, forcing represents persistent large horizontal motions with a small amount of noise, and turbulence develops in response. Three different stratification levels result in buoyancy Reynolds numbers ranging from 11 to 230, and multiple cross-checks for consistency in various dimensionless ratios show that the flows are consistent with theory for strong stratification. The one-dimensional spectra exhibit a broad $\kappa_1^{-5/3}$ *scaling range, perhaps due to the multiple mechanisms resulting in the same power-law scaling. Both three dimensional turbulence at scales smaller than the Ozmidov length scale, and stratified turbulence at scales between the Ozmidov and buoyancy scales produce power-law scaling of* $\kappa_1^{-5/3}$*. A sharper power-law is observed at scales larger than the buoyancy scale. The second-order longitudinal structure functions while exhibiting power-law over the same scaling range do not corroborate Kolmogorov-Obukhov scaling.*

1. Introduction

Much of the ocean and atmosphere is density-stratified with heavier fluid below lighter fluid. The stratosphere takes its name from this phenomenon, but stratification is also an important feature in the lower atmosphere over land at night, and in the ocean below the mixed layer. The military operates in, and relies on sensors "seeing through," these flows. Unfortunately, predictive models for this flow regime lag far behind those for many other fluid flows. A major limitation is that turbulence in stratified flows is not very well understood. In this paper, we consider a data set of direct numerical simulation results being developed to study the fundamental physics of turbulence induced by residual shear subject to strong stable stratification.

When a high Reynolds number flow subject to a stabilizing density gradient becomes unstable due to vertical shearing, the resulting turbulence is thought to be fully three-dimensional, but with the vertical length scale constrained by buoyancy forces. Further, it is thought that the turbulence develops a cascade with net transfer of energy down-scale. These generalizations are based on laboratory experiments, numerical simulations and theoretical studies[e.g., 1–6]. The flow regime is termed stratified turbulence by Lilly[7], Riley and Lelong[8], who provide a review, and Riley and Lindborg[9] discuss its possible relevance to geophysical flows.

While there is some agreement in the literature on the basic features of stratified turbulence, a detailed picture of the energetics is not clear. One of the difficulties is that the flow does not abruptly transition globally from laminar to turbulent, but rather forms patches of turbulence. It has been hypothesized that the flow must have buoyancy Reynolds number, $\mathrm{Re_b}$, above a threshold value of about 30 in order for turbulence to be significant[10–13], and Hebert and de Bruyn Kops[14] show that the strain rate tensor is strongly dependent on $\mathrm{Re_b}$ for low values of that parameter. It seems likely that the energetics of stratified turbulence may also depend on $\mathrm{Re_b}$.

In this paper, we report analyses of all of the terms in the balance equations for the horizontal and vertical contributions of kinetic energy. The data are from direct numerical simulations of stratified flow with a range of $\mathrm{Re_b}$.

There is no mean shear and no internal waves are intentionally introduced, so that the turbulence in the flows results primarily from instabilities in horizontal layers that spontaneously form. A very simple flow configuration is considered for this research. Specifically, persistent numerical forcing at large scales in the horizontal are used to represent a two-dimensional outer-flow. This force induces homogeneous and axisymmetric turbulence at smaller scales. A range of Froude numbers is considered to understand how the turbulence responds to the outer-flow as a function of stratification strength. Importantly, the Froude numbers are order unity or smaller, and are chosen to span the threshold value of Re_b=30. The Prandtl number is unity, close to that in the atmosphere but not that in the ocean; de Stadler, et al.[15] report on a recent investigation on the effect of Prandtl number in simulations of stratified turbulence.

The flow configuration is motivated by the simulations of Lindborg[6] with several important differences. First, in the current simulations the forcing is highly-correlated with itself in time to represent a slowly evolving outer-flow. Second, the simulations are direct numerical simulations (DNS), not large-eddy simulations (LES), so that flows with Re_b typical of laboratory experiments, as well as those above 30, can be compared. The simulations are also motivated by the long history of research in un-stratified isotropic homogeneous turbulence, which forms the foundation for much of turbulence theory. Homogeneous and axisymmetric flow is the analogous "simplest-but-no-simpler" configuration for stratified turbulence.

The paper is organized as follows. In the next section, the literature on stratified turbulence is reviewed in conjunction with defining the governing equations and the principal statistical quantities used to describe the flows. In particular, relationships between various quantities theorized for the limit of very strong stratification are presented in order to provide a qualitative understanding, later in the paper, of where the simulated flows lie in the Reynolds-Froude number parameter space. An overview of the simulations is given in §3 and §4, followed by analyses of the kinetic energy budget in §5. Some conclusions are drawn in §6.

2. Background

The effects of a stabilizing density gradient on turbulence depend on the relative importance of the buoyancy force compared with other forces in the flow. The scope of this paper is limited to flows in which the stratification is "strong," by which it is meant that the buoyancy force is at least as important as the inertial force. The strongly-stratified regime provides a foundation for understanding more general flows influenced by buoyancy. One reason for this is that decaying stratified flow will eventually become strongly-stratified regardless of the initial Froude number[3,16].

When a laboratory flow is strongly-stratified, the dynamics are dramatically different from those of flows subjected to weaker stratification. For instance, horizontal eddies form and decouple in the vertical[1,17–25], and the flows evolve much more slowly in time[18]. A number of experiments have been designed to isolate and study the structure of stratified flows[e.g., 26–31].

Numerical simulations have also been used to understand the stratified regime. Several of the early numerical studies involved hypothetical flows[32–34]. Even as computers have become more capable, simple flow configurations that are not readily realized in physical experiments have proven valuable for understanding certain aspects of stratified turbulence[e.g., 6,14,35–38]. Other simulations have addressed laboratory flows such as wakes, mixing layers, and shear layers[e.g., 39–43]. Recently, Diamessis, et al.[44] used large-eddy simulations to extend the wake experiments of Spedding, et al.[18].

In many instances, simulation results are consistent with laboratory studies, particularly when the Reynolds and Froude numbers in the simulations and laboratory are very close. In other instances, differences in numerical and laboratory results require consideration. For instance, vertical shear between the horizontal layers that form in stratified flows is an important feature of many of the laboratory experiments cited in the preceding paragraphs. Using simulation data, however, Hebert and de Bruyn Kops[45] show that the importance of vertical shear, compared with the other components of the strain rate tensor, decreases rapidly as the buoyancy Reynolds number rises toward the threshold value of 30 cited in §1. Several theoretical analyses lead to the conclusion that the behavior of stratified flow depends on the product of the Froude number squared and the Reynolds number[35,46], which is linearly related to the buoyancy Reynolds number, at least for some ranges of the parameters[14]. Zhi, et al.[47] report on the parameters required for large-eddy simulations to exhibit characteristics of turbulence in geophysical flows.

Apparent differences between various simulations and laboratory experiments suggest that careful consideration must be paid to the location in Reynolds-Froude number parameter space of an experiment or simulation. Simply reporting Reynolds and Froude numbers may not be sufficient, particularly if the dynamically relevant length and velocity scales are among the quantities being investigates. There is an analogous need to accurately identify the Reynolds number regime in studies of un-stratified turbulence. In that case, well-developed theory for the dynamics of turbulence at high Reynolds number exists against which to verify experimental or simulation data. For stratified turbulence, the theory is not as

advanced, but a number of studies provide expected characteristics for flows in the limit of strong stratification. Theoretical analyses of self-similar scaling[48] and the "zigzag" instability[2,36,49] provide frameworks for characterizing a flow. Lindborg[6] reviews a number of theoretical relationships expected to hold for strongly stratified flow at high Reynolds number.

2.1 Governing Equations

The simulated flows considered in this research are solutions to the Navier-Stokes equations in a non-rotating reference frame subject to the Boussinesq approximation. Our interest is in turbulence that results from forcing at large horizontal length scales, and so we parametrize the flow in terms of a velocity scale $\hat{\mathrm{U}}$ and a length scale $\hat{\mathrm{L}}$ that are characteristic of the forcing. The relationship between these characteristic scales and specific lengths and velocities that develop in the simulations is deferred until the simulations are introduced in §3. The remaining quantities used in the parametrization are the acceleration due to gravity, $\hat{g}$, the ambient density, $\overline{\hat{\rho}}(\hat{z})$, the molecular viscosity, $\hat{\mu}$, and the thermal diffusivity, $\hat{\alpha}$. The notation ˆ denotes dimensional quantities which combine to form the nominal Froude, Prandtl, and Reynolds numbers

$$\mathrm{F} = \frac{2\pi\hat{\mathrm{U}}}{\hat{N}\hat{\mathrm{L}}}, \quad \mathrm{Pr} = \frac{\hat{\mu}/\hat{\rho}_0}{\hat{\alpha}} \quad \text{and} \quad \mathrm{Re} = \frac{\hat{\mathrm{U}}\hat{\mathrm{L}}}{\hat{\mu}/\hat{\rho}},$$

with $\hat{N}^2 = -\left(\hat{g}/\overline{\hat{\rho}}\right)\left(d\overline{\hat{\rho}}/d\hat{z}\right)$ the square of the buoyancy frequency and $\hat{\rho}_0$ the reference density. Note that the factor of 2π in the Froude number is often omitted; we find it is advantageous to retain it in light of the limited range of length- and time-scales that exist in our simulations and discuss the point further in conjunction with Table 1.

With the scaling just defined, the dimensionless governing equations are

$$\nabla \cdot \vec{u} = 0 \tag{1a}$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\left(\frac{2\pi}{\mathrm{F}}\right)^2 \rho \vec{e}_z - \nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \vec{u} + \vec{b} \tag{1b}$$

$$\frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho - w = \frac{1}{\mathrm{Re}\,\mathrm{Pr}} \nabla^2 \rho. \tag{1c}$$

Here, $\vec{u} = (u, v, w)$ is the velocity vector, and ρ and p are the deviations of density and pressure from their ambient values. The force, $\vec{b}$, is explained in §3.1. Also, $\vec{e}_z$ is the unit vector in the vertical direction. The pressure is scaled by the dynamic pressure, $\hat{\rho}_0 \hat{\mathrm{U}}^2$, and the density using the ambient density gradient, i.e., it is scaled by $\hat{\mathrm{L}}\left|d\overline{\hat{\rho}}/d\hat{z}\right|$.

2.2 Theoretical Relationships

Many length- and time-scales are used to describe stratified turbulence, and the theoretical relationships between them have been developed in the limit of low Froude number[3,6,48,50]. In simulations, due to the limited dynamic range, these relationships can be difficult to discern, particularly when there are multiple ways to compute a theoretical quantity. Waite[51] provides a recent analysis of the importance of careful attention to length-scales in numerical simulations. In §4, we present numerous cross-checks of the current data set to verify that the flows are consistent with strong stratification. Doing so entails the comparison of several length-scales and dimensionless quantities, which we define below. Some readers may wish to proceed directly to §3.

2.2.1 Length Scales

The outer horizontal scale of stratified turbulence can be estimated by dimensional reasoning as

$$L_{E_h} \equiv \langle E_h \rangle^{3/2} / \langle \epsilon_h \rangle,$$

where $\langle\ \rangle$ indicates an average over the entire spatial domain and multiple times. Calculation of this length-scale in stratified flows is complicated due to the existence of regions that are in approximately solid-body motion with high-energy

but low-dissipation rate. Hebert and de Bruyn Kops[14] found that in simulations, even with significant turbulence, that L_{E_h} can be larger than the size of the simulation domain. The integral, or correlation, scale is more readily interpreted as the size of the horizontal motions in the flow. The horizontal integral length-scale L_{hh} is defined as the integral of the autocorrelation of u_h in the horizontal direction.

The vertical length analogous to L_{hh} for un-stratified flow is L_{vv}, the integral of the longitudinal autocorrelation of u_3. Strongly-stratified flow dynamics; however, are thought to be dominated by the horizontal velocity, and of principal interest in stratified flows is the characteristic thickness of the horizontal layers that form. Therefore, the integral scale L_{hv} based on the vertical autocorrelation of the horizontal velocity is expected to be relevant in the case of strong stratification. Applying order-of-magnitude estimates to the continuity equation leads to $L_{hh}u_z/u_h$ as an estimate for the dynamically important vertical-scale. Here, u_h and u_z are the root-mean-square horizontal and vertical velocities, respectively.

The buoyancy time-scale is related to the buoyancy frequency by $T_b \equiv 2\pi/N$, from which the buoyancy length-scale is defined as

$$L_b \equiv 2\pi u_h/N,$$

As is done in the definition of Froude number, we retain the factor of 2π in the conversion between time and frequency, because omission of order-one constants can make it difficult to interpret DNS data given the limited range of length- and time-scales in the simulated flows. The buoyancy scale is an important length in the generation of turbulence by overturning of internal gravity waves[52,53], and the zigzag instability in columnar vortices[27]. Of significance in the current study is that it is expected to characterize the thickness of the shear layers in stratified turbulence[7,48,54], and so should be comparable to L_{hv}.

Several smaller length-scales are defined in terms of the total dissipation rate of kinetic energy, $\epsilon_k \equiv \epsilon_h + \epsilon_v$. The outer-scale of three-dimensional turbulence in a stratified flow is the Ozmidov scale[55,56],

$$L_O \equiv (\langle \epsilon_k \rangle / N^3)^{1/2},$$

which is based on dimensional reasoning, as is the smallest scale of turbulence, the Kolmogorov length-scale

$$L_K \equiv (\nu^3 / \langle \epsilon_k \rangle)^{1/4}.$$

If the factor of 2π is included in the conversion of frequency-to-time then the Ozmidov scale is

$$L_O^* \equiv \left[\langle \epsilon_k \rangle / \left(N / 2\pi \right)^3 \right]^{1/2}.$$

This is not a standard definition, but it is consistent with including the factor of 2π in the horizontal Froude number defined next.

2.2.2 Dimensionless Relationships

Several definitions of Reynolds and Froude numbers can be written in terms of the foregoing length-scales. Five definitions of horizontal Froude numbers are

$$\mathrm{F}_h = 2\pi u_h / N L_{hh},$$
$$\mathrm{F}_{h1} = L_{hv} / L_{hh},$$
$$\mathrm{F}_{h2} = \left(L_O / L_{hv} \right)^2.$$
$$\mathrm{F}_{h3} = \left(L_O^* / L_{hv} \right)^2,$$

and

$$\mathrm{F}_{h4} = \epsilon_h^{1/3} / N L_{E_h}^{2/3}.$$

The last of these is included for comparison of the current simulations with some in the literature. By the analysis of Billant and Chomaz[48], the vertical length-scale will adjust to the stratification so that the vertical Froude number

$$F_v = 2\pi u_h / N\, L_{hv} \sim 1.$$

Horizontal Reynolds numbers are

$$\mathrm{Re}_h = u_h L_{hh} / \nu$$

and

$$\mathrm{Re}_{h1} = (L_{hh} / L_K)^{4/3},$$

and the buoyancy Reynolds number is

$$\mathrm{Re}_b = (L_O / L_K)^{4/3},$$

Riley and de Bruyn Kops[3] postulate that $\mathrm{Re}_h \mathrm{F}_h^2$, rather than the either the Reynolds or Froude numbers by themselves, is the important parameter for predicting the occurrence of stratified turbulence. Hebert and de Bruyn Kops[14] report that $\mathrm{Re}_h \mathrm{F}_h^2$ and Re_b are linearly related in their simulations.

3. Numerical Simulations

The governing equations (2.1) are solved numerically using the pseudo-spectral technique described in Riley and de Bruyn Kops[3]. Spatial derivatives are computed in Fourier-space, the non-linear terms are computed in real- space, and the solution is advanced in time in Fourier-space with the variable-step, third-order, Adams-Bashforth algorithm with pressure projection. The non-linear term in the momentum equation is computed in rotational form, and the non-linear term in the internal energy equation is computed in conservation and advective forms on alternate time-steps. These techniques are standard to eliminate most aliasing errors, but the simulations reported in this paper are fully-dealiased in accordance with the 2/3 rule via a spectral cutoff filter.

3.1 Forcing

The force $\vec{b}$ in (2.1) is implemented by the deterministic forcing schema denoted Rf in Reference 57. The objective is to force all of the simulations to have the same spectra E_h (κ_h, κ_z) with $\kappa_h<\kappa_f$ and κ_z=0. The highest wave number forced is $\kappa_f=16\pi/\mathcal{L}_h$ with $\mathcal{L}_h$ the horizontal dimension of the numerical domain. Deterministic forcing requires choosing a target spectrum E_f (κ_h <κ_f , 0). Unlike for turbulence that is isotropic and homogeneous in three dimensions, there are no theoretical model spectra for E_f (c.f. [58]). Therefore, run 2 from Lindborg[6] was rerun using a stochastic forcing schema similar to that used by Lindborg and denoted schema Qg in [57]. The spectrum for E_h (κ_h <κ_f , 0) was computed from this simulation and used as the target for the simulations reported in the current paper.

In addition to forcing the large horizontal-scales, 1% of the forcing energy is applied stochastically to wave number modes with κ_h=0 and $\kappa_z=2\pi j/\mathcal{L}_v$, j=2, 3, 4. Here $\mathcal{L}_v$ is the vertical dimension of the numerical domain. This random forcing induces some vertical shear[6].

3.2 Numerical Grid

The extent of the domain in the horizontal and vertical directions is $\mathcal{L}_h$ and $\mathcal{L}_v$ with $\mathcal{L}_h/\mathcal{L}_v$ chosen to accommodate the vertical motions that develop in the flow. The simulation parameters are given at the top of Table 1. While the simulation domains are not cubes and the vertical extent of the domain varies with Froude number, the grid-spacing Δ is the same in all directions. It is assumed for the purpose of choosing the resolution of the numerical grid that the flows are approximately isotropic at the smallest length-scales in the simulation. Therefore, a three-dimensional grid with spacing Δ in all directions is used and any small-scale anisotropy in the flows can be attributed to flow physics rather than to numerical artifacts of an anisotropic grid (c.f. Waite[51]).

4. Overview and Verification of Simulated Flows

Three simulations are considered all having the same forcing spectrum, Prandtl number, and nominal Reynolds number, but different values of Froude number. They are denoted by F1, F2, and F3 in order of descending Froude number. The simulations were run until the energy was statistically stationary, and then run for an additional dimensionless time of about one to collect statistics. Overview data from the simulations is provided in three formats. First, domain-averaged data is tabulated in Table 1. Second, contour plots of velocity are given as Figures 1 and 2. Third, all the terms in the spectral energy balances are plotted in Appendix B. The three formats taken together enable a detailed understanding of the flows that is difficult to attain by analyzing each format sequentially.

Before considering the data, it is worthwhile to review some choices made in choosing the simulation parameters. On the one hand, we are interested in the strongly stratified regime with buoyancy Reynolds number spanning 30. So it is desirable to have one case with Froude number near the upper-limit expected to be consistent with strong stratification. On the other hand, one can hypothesize a lower-limit of the Froude number at which, for a given Reynolds number, the stratification is so strong that two-dimensional turbulence will be pronounced even if three-dimensional stratified turbulence is present. If this occurs, then upscale transfer of energy at large length- scales will preclude a numerically stable and statistically stationary solution without revamping the simulations. For the current paper, a simulation more strongly stratified than those reported was run, but abandoned when the energy at the largest length-scales rose so high as to cause numerical difficulties. This simulation is mentioned because it suggests that the simulations that are reported here fairly bracket the range of strong stratification that can be studied with the current methodology until computers become sufficiently powerful to simulate flows at higher Reynolds number.

Table 1. Simulation parameters and results. Where shown, the ranges are the RMS values of the temporal fluctuations. Parameters from the literature are provided for reference. Run3 is the large-eddy simulation denoted "Run 3" in Lindborg[6]. D9.6 is the DNS denoted D9.6 in Brethouwer, et al.[46].

Case	F1	F2	F3	Run3	D9.6
F	1.048	0.5239	0.2615		
Re	9216	9216	9216		
Pr	1	1	1		
$\mathcal{L}_h/\mathcal{L}_v$	2	4	8		
$\mathcal{L}_h/\Delta$	4096	4096	4096	256	1024
$\mathcal{L}_v/\Delta$	2048	1024	512	32	320
$F_h = 2\pi u_h/N\,L_{hh}$	1.47	0.584	0.258		
$F_{h1} = L_{hv}/L_{hh}$	0.739	0.325	0.194		
$F_{h2} = (L_O/L_{hv})^2$	0.0093	0.0028	0.0007		
$F_{h3} = (L_O^* / L_{hv})^2$	2.30	0.699	0.170		
$F_{h4} = \epsilon_h^{1/3} / NL_{E_h}^{2/3}$	0.0615	0.0231	0.0105		
$F_v = 2\pi u_h/N\,L_{hv}$	1.98	1.79	1.33		
$u_z\,L_{hh}/u_h L_{hv}$	0.685	1.21	1.40	0.0154	0.016
$u_z\,L_{hh}/u_h L_{vv}$	1.63	1.78	2.65		
$Re_h = u_h\,L_{hh}/\nu$	10300	12800	16300		
$Re_{h1} = (L_{hh}/L_K)^{4/3}$	7550	9120	11700		
$Re_h\ F_h^2$	22100	4330	1080		
Re_b	222.6	40.62	10.19	n/a	9.6
$L_{E_h} = \mathcal{L}_h$	0.540	0.711	0.831		
$L_{hh} = \mathcal{L}_h$	0.142	0.177	0.213		
$L_b = \mathcal{L}_v$	0.42	0.41	0.44		
$L_b = L_{hv}$	1.99	1.80	1.33		
$L_{hv} = \mathcal{L}_v$	0.21	0.23	0.33		
$L_{vv} = \mathcal{L}_v$	0.088	0.156	0.174		
$L_O = \Delta$	41.4	12.5	4.42		
$L_k = \Delta$	0.718	0.777	0.775	n/a	0.18
$\langle E_h \rangle$	1.56±0.008	1.54±0.004	1.74±0.013		
$\langle E_v \rangle$	0.20±0.003	0.12±0.003	0.06±0.005		
$\langle E_p \rangle$	0.27±0.006	0.29±0.007	0.22±0.010		
$\langle \epsilon_h \rangle$	0.58±0.091	0.43±0.103	0.44±0.036		
$\langle \epsilon_v \rangle$	0.29±0.047	0.21±0.052	0.20±0.019		
$\langle \epsilon_p \rangle$	0.50±0.051	0.48±0.072	0.36±0.041		

4.1 Contour Plots of Velocity

A horizontal slice of vertical velocity for case F3 is plotted in Figure 1 at four magnifications. The lowest magnification, panel A of the figure, reveals large horizontal structures with linear dimension about one-quarter of the simulation domain. This suggests $L_{hh}/\mathcal{L}_h \approx 1/5$, which is close to the value of 0.213 given in Table 1. Also evident are the patches of strong vertical velocity interspersed with regions of near-zero vertical velocity. These are consistent with Figure 4 in Hebert and de Bruyn Kops[14]. In the other panels of the figure, the region of high vertical motion in the lower-left corner of panel A is magnified to reveal some of its structure. From panel D, which shows a region of 64×64 grid-points, it appears that the flow is well-resolved spatially.

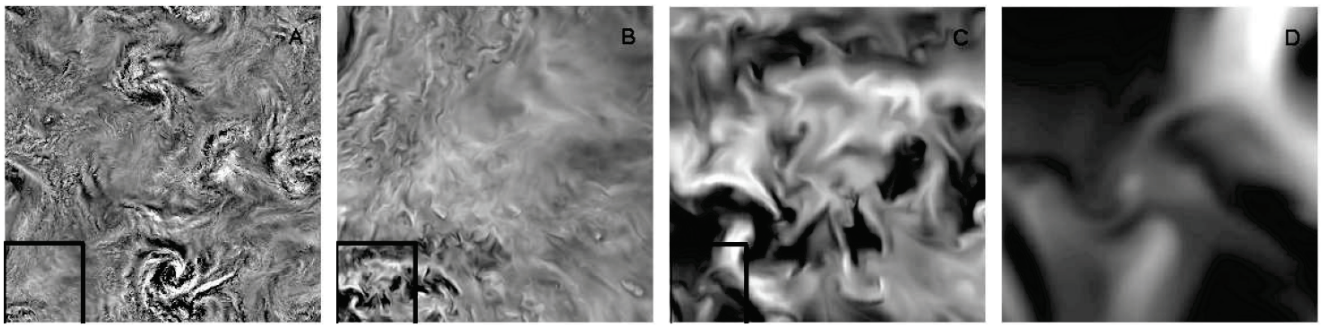


Figure 1. A horizontal slice of vertical velocity normalized by its RMS value for case F3. The color scale goes from -3 (black) to +3 (white). Each panel shows the region in the black square on the preceding panel. The number of grid-points in each direction are: A) 4,096, B) 1,024, C) 256, D) 64.

A vertical slice of the u-velocity for case F3 is shown in Figure 2. At the lowest magnification, it appears that there are four horizontal layers in much of the image, which leads us to expect that the vertical length-scale of the horizontal motions is about 0.25 $\mathcal{L}_v$. The visual estimate of the length-scale is a little smaller than the numerical values of $L_{hv}/\mathcal{L}_v$ and $L_b/\mathcal{L}_v$ given in the table. Note that the ratio $L_{hv}/\mathcal{L}_v$ is increasing with stratification strength while $L_b/\mathcal{L}_v$ stays almost constant for all three cases. This suggests that the buoyancy length-scale is decreasing linearly with Froude number and is the appropriate characteristic vertical length-scale of the horizontal motion. By magnifying the image to produce panels B and C of the figure, Kelvin-Helmholtz billows are evident, which is consistent with the observations of Riley and de Bruyn Kops[3].

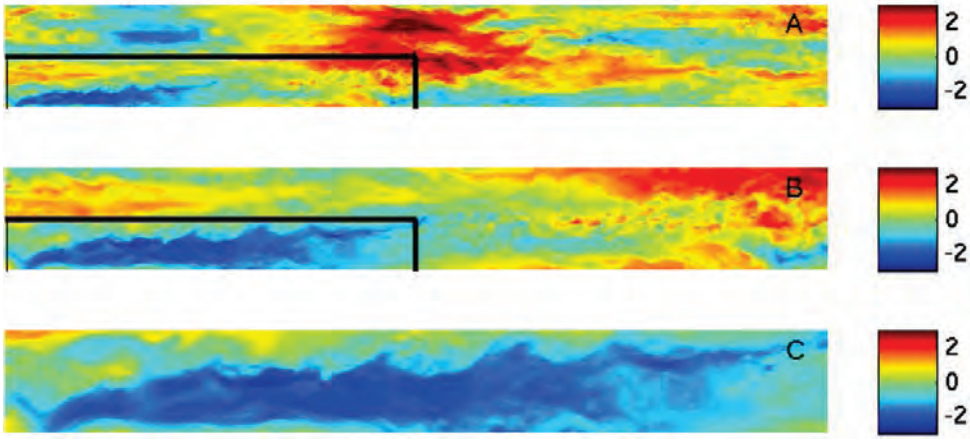


Figure 2. A vertical slice of horizontal velocity normalized by its RMS value for case F3. Each panel shows the region in the black square on the preceding panel. The number of grid-points is: A) 4,096×512, B) 2,048×256, C) 1,024×128.

From the agreement between the visual and calculated vertical length-scales it is concluded that the flow is forming layers with the expected thickness[7,48,54]. It is also concluded that the calculations of the thickness of these layers, in particular the inclusion of the factor of 2π in the definition of L_b, are correct. This is important groundwork for the discussion of Froude numbers which follows.

4.2 Stratification Strength

The Froude numbers are tabulated in the second section of Table 1 below the nominal values and grid parameters. The horizontal Froude numbers F_h and F_{h1} agree very well with the nominal values for all cases, while F_{h3} is in reasonable agreement. F_{h2}, while a valid quantity, is misleadingly low because of the treatment of the factor of 2π. F_{h4} is included for comparison with several simulations reported in the literature.

Next in the table are the vertical Froude numbers. The theory of Billant and Chomaz[48] suggests F_v should be order-one if the flows are strongly stratified, and the data are consistent with this. In all cases, though, F_v is distinctly greater than unity. Recall the relative values of F_h and F_{h1} for each case. A possible explanation for $F_h > F_{h1}$ and $F_v > 1$ is that characteristic vertical scale for the flows is larger than L_{hv}, which is the conclusion arrived at in the discussion of Figure 2. The continuity scaling shown just below the Froude numbers in the table is also consistent with this explanation, except for case F1. Also recall from the discussion in §2 that the suitability of L_{hv} as the characteristic vertical length-scale depends on the stratification being strong.

Taking all of the Froude numbers together, plus the continuity scaling, it is concluded that all of the simulated flows are strongly stratified. Horizontal Froude numbers in the range 0.25 to 1 well-describe the flows. Froude numbers of order 0.01, as suggested by F_{h2} and F_{h4}, are probably misleadingly low.

4.3 Reynolds Numbers

The horizontal Reynolds numbers are listed in the next section of the table. One might expect the Reynolds numbers to be about the same for all the cases, since the large horizontal scales are forced to the same spectrum. Significantly more energy resides at large scales in cases F3 than in cases F1 and F2, though. This can be discerned from two-dimensional spectra (not shown), but is more readily apparent from the tabulated values for energy at the bottom Table 1. The energy cascade will be discussed further in §5, but for now it is simply noted that differences in energies between the various cases results in Re_h spanning almost a factor of two.

The variation in Re_{h1} among the three cases is even more pronounced. This is not particularly surprising since this definition of Reynolds number is based on theory for isotropic homogeneous turbulence. It is interesting that this Reynolds number tracks Re_h as well as it does. If one were to assume that the simulation parameters, except for N, are the same for all the simulations, then one would expect Re_b to decrease by a factor of 4 from one simulation to the next. The data listed in Table 1 is broadly consistent with this expectation.

4.4 Length Scales

In the next section of Table 1, the length-scales defined in §2 are tabulated. As was found by Hebert and de Bruyn Kops[14], the turbulence length-scale, L_{E_h}, is larger than half of the simulation domain size, which is the largest length that can exist in the flow. As noted in conjunction with Figure 1, L_{hh} is consistent with a visual estimate for the horizontal length-scale for case F3. L_{hh} increases with decreasing Froude number, but it is important to note that since the domain size is comparable to L_{hh} there are not that many large structures in the simulation domain and so there is considerable uncertainty in L_{hh}.

The buoyancy length is proportional to Froude number as expected, while the vertical integral length trends with the Froude number. In all three cases, the buoyancy length-scale is larger than the Ozmidov scale. This scale separation allows stratified turbulence to develop. The Ozmidov scale is also significantly larger than the grid spacing, so that the simulations have the dynamic range to allow turbulence at scales smaller than L_O. Recall from the discussion of F_{h2} versus F_{h3}, however, that L_O^*, which includes the factor of 2π in the conversion from frequency-to-time, is likely a better representative of the outer-scale of the three-dimensional turbulence. Since $L_O^* / L_O \approx 15.7$, the dynamic range in all the simulations is more than sufficient to allow three-dimensional turbulence to develop. Similarly, the relationship between the Kolmogorov length-scale and the grid spacing indicates that the simulations are very well-resolved, at least on average. Internal intermittency reduces the spatial resolution locally, though.

5. Analysis

5.1 One-Dimensional Spectra and Structure Functions

The longitudinal, transverse, and vertical spectra of horizontal velocity are denoted $E_{11}(\kappa_1)$, $E_{22}(\kappa_1)$, and $E_{22}(\kappa_3)$, respectively. These are computed taking advantage of the axisymmetric configuration of the flow, that is, $E_{11}(\kappa_1)$ is the average of the longitudinal spectra of u_1 and u_2 and similarly for the transverse spectra. The vertical spectra are related to the vertical autocorrelations of u_1 and u_2, not the autocorrelation of u_3. As is usual with simulation data, all the spectra are averaged over the computational domain. The spectra are shown in Figure 3. The longitudinal spectra are plotted true, while the transverse and vertical spectra are offset by one and two decades, respectively. To make judging the slope of the curves easier, some of the expected power-laws are also plotted. To identify the "scaling range" in terms of theoretical length-scales, wave numbers corresponding to L_O^*, L_b, and L_K are indicated by tick marks near the bottom of the figure. Recall from the discussion of Table 1 that L_O^*, not L_O, seems to be consistent with the outer length-scale of three-dimensional turbulence postulated by Ozmidov.

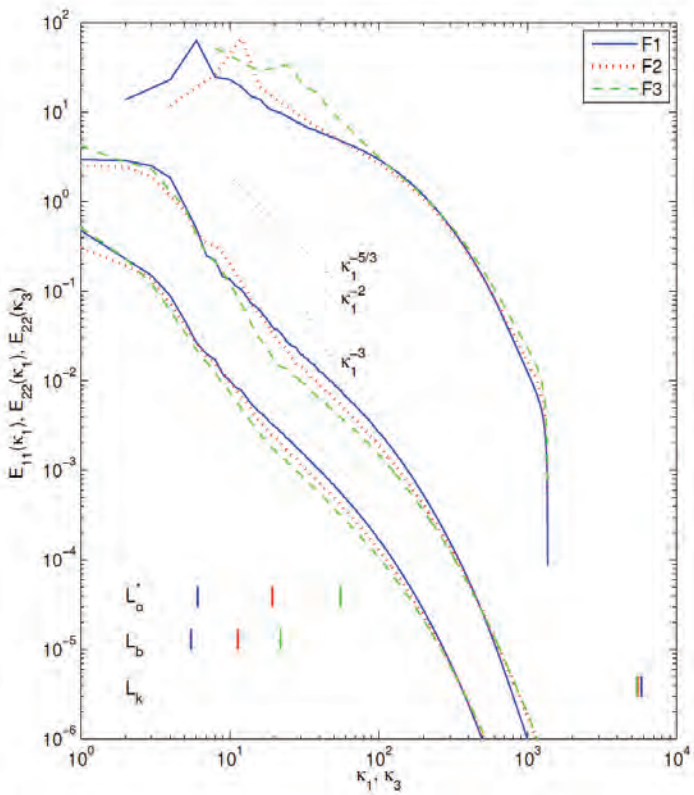


Figure 3. Longitudinal, transverse, and vertical spectra. The longitudinal spectra is plotted true, the transverse spectra are offset by one decade, and the vertical spectra are offset by two decades. The length-scales L_O^*, L_b, L_k are shown below the spectra.

For the longitudinal and transverse spectra, it is expected that three regions with power-law scaling might be evident: a region between the forcing length-scale and L_b, a second between L_b and L_O^*, and a third for scales smaller than L_O^*. For case F3 all three regions are observed. In the small range between the forcing length and L_b the spectra scale with κ_1^{-3}. Between L_b and L_O^*, $\kappa_1^{-5/3}$ scaling is observed, which is consistent with the literature[e.g., 5,6,9,59–61]. At scales smaller than L_O^*, $\kappa_1^{-5/3}$ scaling is also observed, as expected, based on the theory of isotropic homogeneous turbulence. For cases F1 and F2, there is little scale separation between L_b and L_O^* but scaling with, $\kappa_1^{-5/3}$ is apparent over an extended range of wave numbers.

Scaling of the vertical spectra with κ_3^{-3} satisfies self-similarity under the assumption of strong stratification[48] and is also discussed in, e.g., the references cited in the previous paragraph. For case F3, this scaling is apparent between L_b and L_O^*. It is not apparent in the other cases, but it is not clear if this result is significant or if it is just an artifact of the limited scale separation and the inherent aliasing in one-dimensional spectra.

It is tempting to conclude that both three-dimensional turbulence and stratified turbulence produce $\kappa_1^{-5/3}$ scaling in one-dimensional spectra. This would explain why the range of wave numbers over which this power-law scaling is observed is significantly greater than expected based on simulations of un-stratified turbulence at comparable Reynolds numbers[e.g., 62]. To provide more information about the power-law scaling range, we consider the second-order longitudinal structure functions of velocity $D_{11}(r_1$, averaged for u_1 and u_2, plotted in Figure 4.

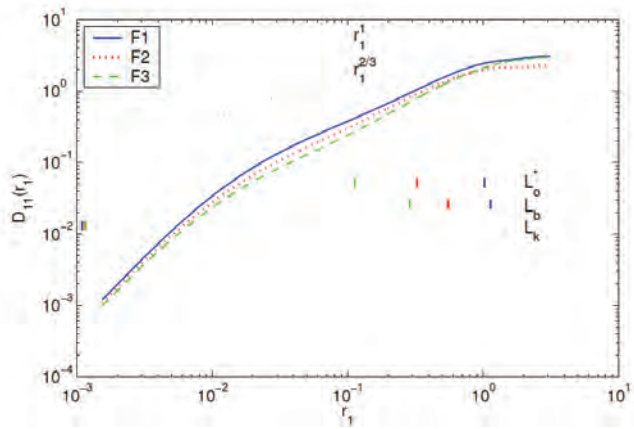


Figure 4. Second-order longitudinal structure functions

Structure functions and spectra contain the same information, and one can be computed from the other. In un-stratified turbulence with large scale separation, the power-law scaling of $D_{11}(r_1)$ can be deduced from that of $E_{11}(\kappa_1)$; Obukhov's $\kappa_1^{-5/3}$ scaling corresponds to Kolmogorov's $r_1^{2/3}$ scaling. This relationship is observed in simulations of un-stratified turbulence[e.g., 62]. It is not observed in the present simulations, though. In Figure 4, the structure functions scale with roughly r_1^1 at length-scales corresponding to the $\kappa_1^{-5/3}$ region of the spectra. This suggests that the flows do not exhibit Kolmogorov-Obukhov scaling, which is in agreement with the hypotheses of Riley and Lindborg[9]. Note that much finer spatial resolution is required for simulations to exhibit r_1^2 scaling for small r_1 (c.f. [62]), and so this relationship cannot be checked for the current simulations.

Next, consider the ratio of the transverse to longitudinal spectra plotted as Figure 5. The theoretical value for the ratio for un-stratified isotropic homogeneous turbulence at infinite Reynolds number is 4/3, and DNS show this value almost exactly for flows with Reynolds numbers comparable to those in the current simulations[e.g. 62]. While the ratios in Figure 5 are close to 4/3, there is a clear trend away from that theoretical value as the Froude number decreases.

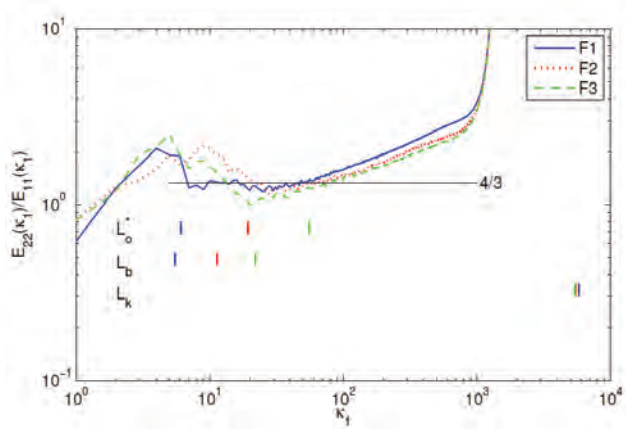


Figure 5. Ratio of the transverse to the longitudinal spectra. The black line is provided as an aide for reading the logarithmic scale and not to suggest that the theoretical relationship based on Kolmogorov[63] should apply.

6. Conclusion

Direct numerical simulations of forced, homogeneous and axisymmetric flow with a range of density gradients in the vertical direction were computed to understand aspects of turbulence in stratified flows. There is no mean shear, and no internal waves were intentionally introduced into the simulations, so that the turbulence in the flows results primarily from instabilities in horizontal layers that spontaneously form. Numerical forcing was used to represent persistent motions at large horizontal length-scales and numerical grids with $\mathcal{O}(10^{10})$ grid-points provide more than three decades of scale separation between the length-scale of the forcing and the smallest length-scales in the flows. The small scales are well-resolved, both in terms of the ratio of the grid spacing to the *average* Kolmogorov length-scale, the relevance of which in stratified turbulence is an open question, and in terms of visualizations of the velocity fields in patches of intense turbulence characteristic of stratified flows.

Cross-checks of multiple definitions of dimensionless quantities and relationships between length-scales summarized in Table 1, show that the flows are consistent with theoretical characteristics of strongly-stratified turbulence at high Reynolds number. In particular, the data set includes buoyancy Reynolds numbers well above the often-accepted threshold value of 30 necessary for the flows to be strongly turbulent[10–13]. At the same time, the horizontal Froude numbers are $\mathcal{O}(1)$ or smaller. It is concluded that the data set well-represents a canonical flow configuration for studying stratified turbulence.

The power-law scaling of the one-dimensional spectra of these simulations suggest a much broader scaling range compared with the DNS of un-stratified isotropic homogeneous turbulence with similar dynamic range, perhaps due to the multiple mechanisms resulting in the same power-law scaling. Scaling with $\kappa_1^{-5/3}$ extends over two wave number bands: the range of scales smaller than Ozmidov length-scale, which corresponds to the three- dimensional turbulence, and the range between the Ozmidov and buoyancy scales, which corresponds to the stratified turbulence. The second-order longitudinal structure functions, while exhibiting power-law scaling over the same range of length-scales, do not corroborate Kolmogorov-Obukhov scaling. It is concluded that power-law scaling of $\kappa_h^{-5/3}$ in one-dimensional spectra does not indicate Kolmogorov-Obukhov scaling, nor is it sufficient to identify the inertial range. More research is required, with flows having broader scaling ranges, to determine how to identify the inertial range in stratified flows based on one-dimensional spectra.

Acknowledgments

This work is funded by the US Office of Naval Research, Grant No. N00014-12-10583, managed by Dr. R.D. Joslin and benefited from many discussions with Prof. Jim Riley. Grants of high performance computer time were provided through the US Department of Defense High Performance Computing Modernization Program by the Arctic Region Supercomputing Center, the US Army Engineer Research and Development Center, and the US Army Research Laboratory.

References

1. Fincham, A.M., T. Maxworthy, and G.R. Spedding, Dyn. Atmos. Oceans, 23, pp.155–169, 1996.
2. Billant, P. and J.-M. Chomaz, *J. Fluid Mech.*, 418, pp. 167–188, 2000.
3. Riley, J.J. and S.M. de Bruyn Kops, *Phys. Fluids*, 15(7), pp. 2047–2059, 2003.
4. Praud, O., A.M. Fincham, and J. Sommeria, *J. Fluid Mech.*, 522, pp. 1–33, 2005.
5. Lindborg, E., *Geophys. Res. Let.*, 32, pp. L01809, 2005.
6. Lindborg, E., J. Fluid Mech., 550, pp. 207–242, 2006.
7. Lilly, D.K., *J. Atmos. Sci.*, 40, pp. 749–761, 1983.
8. Riley, J.J. and M.-P. Lelong, *Annu. Rev. Fluid Mech.*, 32, pp. 613–657, 2000.
9. Riley, J.J. and E. Lindborg, *J. Atmos. Sci.*, 65(7), pp. 2416–2424, 2008.
10. Gibson, C.H., In J. C.J. Nihous, editor, *Marine Turbulence*, Elsevier, pp. 221–257, 1980.

11. Imberger, Jörg and Boualem Boashash, *J. Phys. Oceanogr.*, 16, pp. 1997–2012, 1986.
12. Gregg, M.C., *J. Geophys. Res.–Oceans*, 92, pp. 5249–5286, 1987.
13. Smyth, W.D. and J.N. Moum, *Phys. Fluids*, 12, pp. 1327–1342, 2000.
14. Hebert, D.A. and S.M. de Bruyn Kops, Phys. Fluids, 18(6), pp. 1–10, 2006.
15. de Stadler, Matthew B., Sutanu Sarkar, and Kyle A. Brucker, *Phys. Fluids*, 22(9), 2010.
16. Chomaz, J.-M., P. Bonneton, A. Butet, and E. J. Hopfinger, *Phys. Fluids A*, 5, p. 2799, 1993.
17. Lin, J.-T. and Y.-H. Pao, *Annu. Rev. Fluid Mech.*, 11, p. 317, 1979.
18. Spedding, G.R., F.K. Browand, and A.M. Fincham, *J. Fluid Mech.*, 314, p. 53, 1996.
19. Voropayev, S.I., Y.D. Afanasyev, and I.A. Filippov, *J. Fluid Mech.*, 227, p. 543, 1991.
20. Voropayev, S.I., D.L. Boyer, Z. Zhang, H.J.S. Fernando, and C.W. Pok, *Phys. Fluid.*, 9, p. 115, 1997.
21. Flór, J.B. and G.J.F. van Heijst, *J. Fluid Mech.*, 279, p. 101, 1994.
22. Trieling, R.R. and G.J.F. van Heijst, *Fluid Dyn. Res.*, 23, p. 735, 1998.
23. Beckers, M., H.J.H. Clerx, G.J.F. van Heijst, and R. Verzicco, *Phys. Fluids*, 14, p. 704, 2002.
24. Bonnier, M. and O. Eiff, *Phys. Fluids*, 14, p. 791, 2002.
25. Meunier, Patrice, Peter J. Diamessis, and Geoffrey R. Spedding, *Phys. Fluids*, 18(10), 2006.
26. Bonnier, M., O. Eiff, and P. Bonneton, *Dynamics of Atmospheres and Oceans*, 31, pp. 117–137, 2000.
27. Billant, P. and J.-M. Chomaz, *J. Fluid Mech.*, 418, pp. 167–188, 2000.
28. Voropayev, S.I., S.A. Smirnov, and A. Brandt, *Phys. Fluids*, 13, p. 3820, 2001.
29. Spedding, G.R., *J. Fluid Mech.*, 454, p. 71, 2002.
30. Godoy-Diana, R. and J.-M. Chomaz, *Phys. Fluids*, 15, pp. 1058–1064, 2003.
31. Godoy-Diana, R., J.-M. Chomaz, and P. Billant, *J. Fluid Mech.*, 504, pp. 229–238, 2004.
32. Herring, J.R. and O. Métais, *J. Fluid Mech.*, 202, p. 97, 1989.
33. Métais, O. and J. R. Herring, *J. Fluid Mech.*, 202, p. 117, 1989.
34. Jacobitz, F.G., S. Sarkar, and C.W. VanAtta, *J. Fluid Mech.*, 342, pp. 231–261, 1997.
35. Riley, J.J. and S.M. de Bruyn Kops, *Phys. of Fluids*, 15(7), pp. 2047–2059, 2003.
36. Deloncle, A., P. Billant, and J.-M. Chomaz, *J. Fluid Mech.*, 599, pp. 229–239, 2008.
37. Brethouwer, G. and E. Lindborg, *J. Fluid Mech.*, 631, pp. 149–163, 2009.
38. Deloncle, A., P. Billant, and J.-M. Chomaz, *J. Fluid Mech.*, 678, pp. 482–510, 2011.
39. Basak, S. and S. Sarkar, *J. Fluid Mech.*, 568, pp. 19–54, 2006.
40. Brucker, K.A. and S. Sarkar, *Phys. Fluids*, 19(10), 2007.
41. Arobone, E. and S. Sarkar, *Phys. Fluids*, 22(11), 2010.
42. Brucker, K.A. and S. Sarkar, *J. Fluid Mech.*, 652, pp. 373–404, 2010.
43. Pham, H.T. and S. Sarkar, *J. Turbulence*, 11(24), pp. 1–23, 2010.
44. Diamessis, P.J., G.R. Spedding, and J.A. Domaradzki, *J. Fluid Mech.*, 671, pp. 52–95, 2011.
45. Hebert, D.A. and S.M. de Bruyn Kops, *Geophys, Res. Let.*, 33, p. L06602, 2006.
46. Brethouwer, G., P. Billant, E. Lindborg, and J.-M. Chomaz, *J. Fluid Mech.*, 585, pp. 343–368, 2007.
47. Zhi, Shen, Li YuPeng, Cui GuiXiang, and Zhang ZhaoShun, *Science China-Physics Mechanics & Astronomy*, 53(1), pp. 135–146, 2010.
48. Billant, P. and J.-M. Chomaz, *Phys. Fluids*, 13, pp. 1645–1651, 2001.
49. Billant, P., A. Deloncle, J.M. Chomaz, and P. Otheguy, *J. Fluid Mech*., 660, pp. 396–429, 2010.
50. Tseng, Yu-heng and J.H. Ferziger, *Phys. Fluids*, 13(5), pp. 1281–1293, 2001.
51. Waite, L., *M. Phys. Fluids*, 23(6), p. 066602, ISSN 1070-6631, JUN 2011.
52. Carnevale, G.F., M. Briscolini, and M. Orlandi, *J. Fluid Mech.*, 427, pp. 205–239, 2001.
53. Waite, M.L. and P. Bartello, *J. Fluid Mech.*, 546, pp. 313–339, 2006.
54. Waite, M. and P. Bartello, *J. Fluid Mech.*, 517, pp. 281–308, 2004.
55. Lumley, J.L., *J. Atmos. Sci*., 21(99-102), 1964.
56. Ozmidov, R.V., *Izv. Atmospheric and Oceanic Physics*, 1, pp. 493–497, 1965.
57. Rao, K.J. and S M. de Bruyn Kops, *Phys. Fluids*, 23, pp. 065110, 2011.

58. Overholt, M.R. and S.B. Pope, *Computers & Fluids*, 27, pp. 11–28, 1998.
59. Dewan, E.M., *Science*, 204(4395), pp. 832–835, 1979.
60. Dewan, E.M and R.E. Good, *J. Geophys. Res.-Atmos.*, 91, pp. 2742–2748, 1986.
61. Kitamura, Y. and Y. Matsuda, *Geophys. Res. Let.*, 33(5), p. L05809, 2006.
62. Almalkie, S. and S.M. de Bruyn Kops, *J. Fluid Mech.*, 697, pp. 204–236, 2012.
63. Kolmogorov, A.N., *Dokl. Akad. Nauk SSSR*, 30, pp. 299–303, 1941.

High Power Density Energy Storage and Bio-Inspired Devices

J. Bernholc, V. Ranjan, B. Tan, M. Hodak, and W. Lu
Center for High Performance Simulation and Department of Physics, North Carolina State University, Raleigh, NC
{bernholc, vranjan, btan, mhodak, wlu2}@ncsu.edu

M. Buongiorno Nardelli
Department of Physics and Department of Chemistry, University of North Texas, Denton, TX
Marco.Nardelli@unt.edu

Abstract

Recent advances in theoretical methods combined with the advent of massively-parallel supercomputers allow one to reliably simulate the properties of complex materials and device structures from first-principles. We describe applications in two general areas: i) novel polymer composites for ultrahigh power density capacitors, necessary for pulsed power aplications, such as electric rail guns, power conditioning, and dense electronic circuitry, and ii) electron transport in DNA, for potential applications in sensor structures. Polyvinylidene Fluoride (PVDF) and its copolymers have recently been shown to combine high energy density, high breakdown field and fast speed. However, the microscopic origin of their non-linear dielectric response is not understood, which prevents their optimization to further increase the energy density and reduce loss. Using first-principles simulations, we uncover a cooperative transition path in PDVF-based polymers that connects its non-polar and polar phases. This path, which explores the complex torsional and rotational manifold of the PVDF system, is thermodynamically and kinetically accessible at relatively low temperatures. The resulting phase transition is responsible for non-linear dielectric properties and the high energy density in PVDF. Turning to bio-inspired devices, we are studying charge transport across matched and mismatched DNA strands. Recent experiments have shown that the conductivity of matched strands is much higher than that of mismatched ones. We can explain these trends in terms of alignment and mismatch of molecular orbitals of the individual bases, but also observe that the conductivity drastically depends on the presence of solvated ions.

1. Introduction

The usual means of storing electrical energy are either *battery*, where the current induces chemical reactions, or *capacitors*, where especially chosen *dielectrics* enhance the stored energy. Current batteries offer *energy densities* of about 10^2–10^3 Wh/kg, but their *power densities*, i.e., the speeds at which they can be discharged, are only 50–300 W/kg.[1] Fuel cells have energy densities[1] several times larger than batteries, but their power densities are significantly lower. Conventional capacitors have energy densities of only ~0.05 W/kg, but their power densities reach 10^5 W/kg and they can be fully discharged over 10^6 times, compared to 200–1,000 times for batteries. They can thus be used to release or store quick bursts of energy, which is required in pulsed power applications. Ultracapacitors, or electric double-layer capacitors, occupy a middle ground in energy storage. They achieve energy densities of ~10 Wh/kg and power densities of ~10^3 W/kg by using porous plates suspended in an electrolyte to attract negative/positive ions to the positive/negative electrodes, thereby dramatically shrinking the distance between the positive and negative "layers." This increases the capacitance per unit area, $c=\varepsilon_0 K/d$, where K is the dielectric permittivity and d is the distance between capacitor plates. However, the charge/discharge speeds of ultracapacitors are limited by ion diffusion in the electrolyte, while the porosity of the plates leads to small breakdown voltages within each cell. In contrast, by using ultrafast phase transitions as the primary mechanism of energy storage, PVDF-based capacitors reach the power densities and breakdown fields of conventional capacitors, while their energy densities are 10^2–10^3 larger, potentially rivalling those of ultracapacitors.

In traditional capacitors, the stored energy density is given by $U=1/2\ \varepsilon_0 K E_b^2$, where E_b is the electric breakdown field. Enhancement of K and/or E_b leads to increased energy storage. However, both are native properties that are constants for any chosen material. High K materials such as ferroelectric oxides have low E_b. Polymers typically have high E_b but very small K. Materials that have both high K and high E_b have not yet been found and attempts to create composites with the same characteristics have not yet met with success. An alternative avenue to enhance the stored energy is to search beyond "linear" dielectrics for materials where the effective K changes with the applied field. We describe the mechanism by which concerted atomic transformations induced by the electric field lead to phase transformations in PVDF-based polymers, resulting in storage densities rivalling those of ultracapacitors while maintaining the power densities associated with traditional capacitors. The phase-transition route could also be explored in other systems that undergo field-induced structural transformations, providing a different, largely unexplored route for optimizing materials with high power and storage densities.

Reference 2 has shown that an admixture of 9% CTFE impurities into PVDF results in an order of magnitude increase in energy density, reaching 17 J/cm^3. The dramatic non-linear dependence of energy density on CTFE concentration is a clear indication that a structural change must be involved. Indeed, we have previously postulated[3] that the greatly enhanced energy density is due to a reversible phase transition induced by the electric field. This assertion is supported by the fact that the non-polar state is indeed observed as the initial phase in x-ray spectra.[2] However, the atomistic mechanism of the transformation was not determined and previous work suggested that kinetic barriers for polymer transformations are ~1 eV,[3] precluding the fast kinetics observed experimentally. In contrast, the studies of multistage *concerted* motion described below uncover a novel mechanism in the PVDF family of polymers with modest energy barriers, easily capable of transforming their structure at room temperature. Furthermore, these barriers can be significantly modified by admixing appropriate copolymers, which provide a novel avenue for optimizing the properties of energy-storage materials for specific applications.

Charge transport in DNA has attracted much attention since initial works showed that the stacking of aromatic bases in DNA resembles one dimensional metallic molecular crystals.[4] Potential applications include bio/nano-electronics,[5] eletrochemical sequencing techniques[6] and possibility to detect DNA damage. Numerous experimental studies on charge transport in DNA have yielded a wide range of results from insulating,[7,8] semiconducting,[9] to metallic.[10] A recent state-of-the-art measurement[11] showed that a 15 base pair (bp) B-DNA has a very high conductance and that a single mismatch in the DNA decreases the conductance 300 times. Our goal is to understand these results and to determine key aspects of the mechanism responsible for charge transport in DNA.

2. Methods and Calculations

The band structure calculations are performed using PWscf[12] implementation of density functional theory and ultrasoft pseudopotentials. Depending on the system, the cutoffs are up to 35 Rydbergs and up to 32 k-points are used to ensure high level of convergence. All atoms are fully relaxed. For polymer calculations, the Born effective charges Z_i^* for each of the ions i are obtained using density functional perturbation theory.[13] Following Fu and Bellaiche[14] the term $-eZ_i^*E$ is added to the Hellmann-Feynman forces $\mathbf{f}_i$, where e is the elementary charge. The atomic structure is relaxed until the total force on each atom is close to zero, i.e., until $\mathbf{f}_i = -eZ_i^* E$.[14–16]

The quantum transport calculations are carried out using a massively parallel real-space multigrid implementation [17] of density-functional theory (DFT). Both the nanotube electrodes and the conductor parts are described at the same DFT footing, employing the generalized gradient approximation (GGA) and ultrasoft pseudopotentials. The conductance and nonlinear I-V characteristics are studied by non-equilibrium Green function (NEGF) method in a basis of optimally localized orbitals.[18,19] When an external bias is applied, the electronic structure is greatly affected, due to shifts in the electronic levels and non-equilibrium current. Full electronic self-consistency is thus critically important.[20] After it is reached, the transmission function and current are calculated at each bias voltage.

The transmission function is obtained from the Landauer formula

$$T(E,V) = \frac{2e^2}{h} Tr\left[\Gamma_L(E,V) G^R(E) \Gamma_R(E,V) G^A(E)\right]$$

where $G^{R,A}$ are the retarded/advanced Green functions of the conductor part and $\Gamma_{L,R}$ are the coupling functions to the left/right electrodes. The current through this system is calculated by integrating the transmission function over the bias window

$$I(V)=\int_{-\infty}^{\infty} T(E,V)\left[f(E-\mu_L)-f(E-\mu_R)\right]dE$$

where f is the Fermi distribution function, $\mu_{L,R}$ are the chemical potentials of the two electrodes, and the bias potential V is represented by the difference between the chemical potentials, $eV=\mu_L-\mu_R$.

Setup of our calculations mimics settings used in experiment[11] with DNA sandwiched between (5,5) carbon nanotube leads. DNA is connected to the leads via alkane linker –CONH-$(CH_2)_3$. The setup of the system is shown in Figure 1.

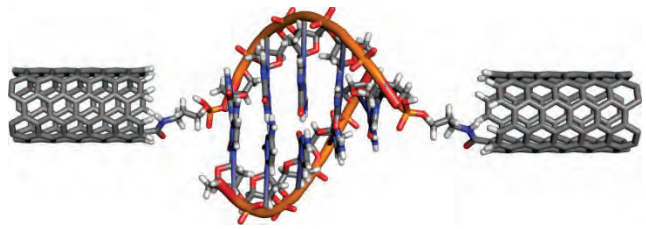


Figure 1. DNA fragment sandwiched between nanotubes. Carbon atoms are in gray, nitrogen in blue, hydrogen in white, oxygen in red, and phosphorous in orange.

3. Results and Discussion

3.1 High Energy Density Polymers

The structure of a single PVDF polymer chain is characterized by dihedral angles between the individual monomers: when the dihedral angles alternate between −57° and +57° the chain is in a trans-gauche conformation (TGTG'), see Figure 2, while when all the dihedrals are equal to 180°, the chain is in an all-trans conformation (TT). As the individual chains organize in a crystalline environment, they can form four different polymorphs, depending on the individual conformations of the chains and their relative orientation: i) anti-polar TGTG' (α-PVDF), ii) polar TGTG' (γ-PVDF) with polarization of 0.09 C/m^2 per unit cell, iii) anti-polar TT (δ-PVDF), and iv) polar TT (β-PVDF), with polarization of 0.19 C/m^2 per unit cell.

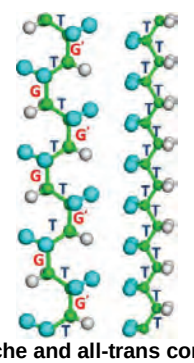


Figure 2. Trans-gauche and all-trans confirmations of PVDF

In our previous work,[3] we have shown that the thermodynamically preferred phase of P(VDF-CTFE) with a low CTFE concentration changes from the non-polar α to the polar β phase as the electric field increases. However, the thermodynamic preference cannot explain the strongly increasing displacement field and the large variation in the dielectric permittivity that is observed in experiments, and which eventually leads to high energy density. Moreover, a direct transformation from α- to β-PVDF, although thermodynamically feasible, would require unphysically large electric fields. Only by assuming a disordered multi-domain structure with a distribution of concentrations, the gradual non-linear dielectric response seen in experiments becomes reproducible. This assumption, albeit successful in interpreting the experimental data, falls short of providing a satisfactory microscopic explanation of the mechanism of high energy density storage in PVDF and its copolymers.

The fact that α and β chains of PVDF are related to each other by rotational transformations[5] has motivated the realization that transformations between the competing polymorphs can be described as a sequence of non-trivial but nevertheless easily characterized symmetry operations: i) in the rotational manifold of α-PVDF, rotating one of the polymer

chains by 180° yields γ-PVDF, and ii) β-PVDF is a torsional manifold of γ-PVDF. A clear topological pathway is thus available to the system to morph from one phase to the other, first by transforming from the anti-polar α phase to the polar γ phase (Figure 3), and then changing all of the dihedral angles from ±57° to 180° and reaching the β phase**.** As a consequence, PVDF can exhibit an adaptable response to electric fields by selecting an appropriate functional structure chosen from its large ensemble of interrelated topological states. However, this manifold of torsions and interchain rotations must be thermodynamically and kinetically accessible at relatively low temperatures for the transformations to occur. If a low-barrier transition path exists, it would lead to non-linear dielectric response and could consequently explain the high energy density observed in *some* PVDF-based materials.

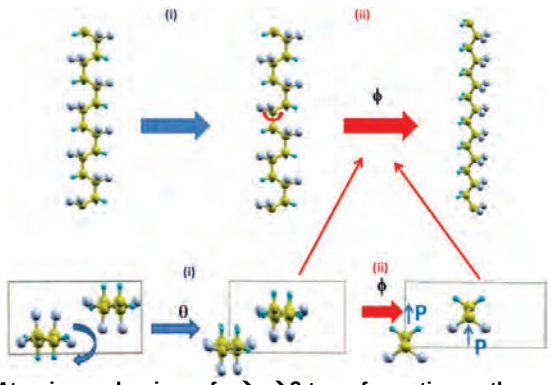


Figure 3. Atomic mechanism of α→γ→β transformation pathway (see text)

We evaluate the energies of the intermediate structures along the α→γ→β pathway for both PVDF and P(VDF-CTFE) 90/10 mol%, using first-principles calculations based on density-functional theory. In order to reach the minimum energy structures, we perform atomic and volume relaxations at all intermediate points along the path. Starting from the α structure, we gradually rotate one of the chains to sample the potential energy surface associated with the rotational manifold connecting the α and γ structures. The transition from γ to β involves a TGTG' to an all-trans (TT) transformation. We generate intermediate structures by alternatively changing the dihedral angles in steps of ±Δθ (Δθ=15°) from their values in the γ configuration (±57°) to the final β geometry (180°). The intermediate structures thus created are fully relaxed, under the constraint that the dihedral angles remain at their assigned values. The unit cell volume is also continuously optimized during the transformation. In Figure 4(a) we display the values of the total energy per C atom along the transformation pathway as a function of the two independent principal order parameters: the interchain angle (θ) and the dihedral angle (φ). First, θ changes from 0° to 180° and then φ changes from ±57° to 180° while holding θ=180°. Our results demonstrate the existence of a low barrier energy path, starting from the stable non-polar native α phase to the polar higher-energy β phase with a relative energy difference of ~24 meV/C-atom. The intermediate polar γ phase is metastable, with an energy 32 meV/C-atom greater than the α phase.

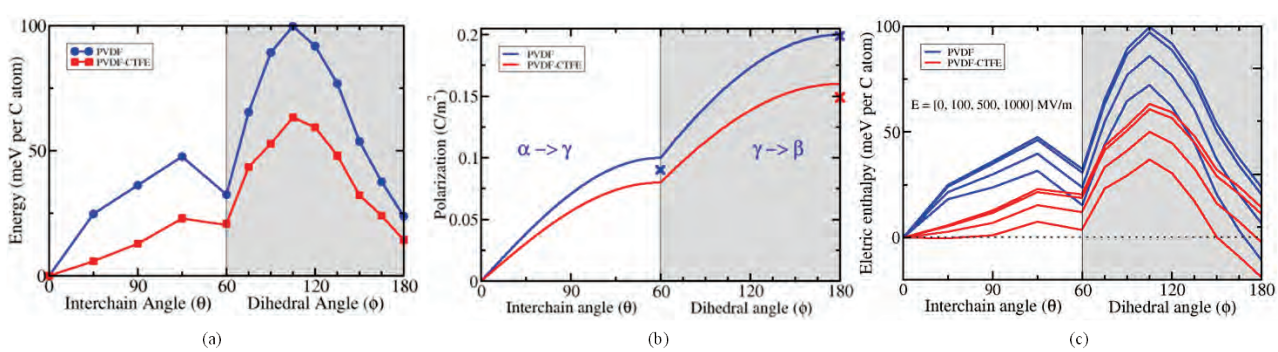


Figure 4. Energetics and polarization along the transformation pathway from the α to the β phase. The values for pure PVDF are in blue and for P(VDF-CTFE) 90/10 mol% in red. The change in interchain angle from 0° to 180° results in α→γ transition (white background), while the change in the dihedral angle from 60° to 180° describes the γ→β transformation (grey background). (a) Relative energies along the transition path. (b) Polarization based on analytic expression. The crosses are the value of polarization from Berry phase calculation. (c) Electric enthalpy for increasing values of the electric field along the path: 0, 100, 500 and 1,000 MV/m.

The energy barriers for the α→γ and the γ→β transitions are 50 and 100 meV/C-atom, respectively. In the same figure we also report results for the P(VDF-CTFE) 90/10 mol% case. The introduction of CTFE has a dramatic effect on the kinetics of the transformation. The energy barrier between α and γ is now reduced by almost a factor of 2 to less than 25 meV/C-atom, and the one between γ and β to less than 60 meV/C-atom. This drastic reduction in activation energies makes the phase transformation kinetically accessible at considerably lower temperatures and it dramatically lowers the critical electric field that will trigger the transformation. The calculated polarization along the α→γ→β transition path is plotted in Figure 4(b), together with the Berry phase values. The figure shows that polarization is only weakly affected by the 10% admixture of CTFE. Its large value leads to significant lowering of the enthalpy barriers separating the three phases. Figure 4(c) displays the electric enthalpy at different values of the electric field. At 500 MV/m, the activation enthalpies for α→γ and the γ→β transitions become 32 and 88 meV/C-atom, respectively, for pure PVDF, and 18 and 50 meV/C-atom for P(VDF-CTFE) 90/10 mol%. As expected from the $V\,\boldsymbol{P}\bullet\boldsymbol{E}$ term, for the highest values of the electric field the electric enthalpy of the β phase drops below that of α. Figure 4(c) clearly shows that a strong electric field both enables and accelerates the transformation to the β phase. Furthermore, the introduction of a small amount of CFTE has a dramatic effect on the transformation, because the polarization along the transition path does not change significantly, but the enthalpy barriers are reduced considerably. The combination of these two effects greatly enhances the kinetics and enables a fast response even at moderate temperatures.

In summary, we have uncovered the mechanism behind the non-linear dielectric response of certain PVDF-based polymers and showed how it can lead to ultrafast energy discharge *and* high energy density. This combination of properties is due to electric field-induced phase transition from a ground state nonpolar structure to a higher-energy polar structure. We showed that the relevant order parameters for the transition are collective changes of rotational and dihedral angles, rather than the formation and propagation of localized kinks. We also demonstrated that an addition of a small amount of a copolymer results in: i) a significant reduction of energy barriers, and ii) a small change in polarization. We believe that copolymers can also be chosen with the goal of controlling the morphology and microstructure as well as optimizing the energy hypersurface, in order to minimize hysteresis and loss.

3.2 Simulation of DNA Conductance

3.2.1 Dry DNA

Multiple theoretical studies[21,7], have established that dry DNA exhibits only small conductivity and has a large gap (~2 eV). Our calculations confirm these results. This is demonstrated on Figure 5, where partial density of states of 15 bp $(GC)_{15}$ DNA is shown. The figure shows that while HOMO states are well aligned, they are not extended and are localized on guanine groups. We have calculated current for a case of short 2 bp $(GC)_2$ DNA, where the effect of localization should be minimized, and find a value of only 7×10^{-9} nA for source bias of 50mV, while experiment[11] yielded values that are 11 order of magnitude higher. Also, transmission shows no significant peaks within 0.5 V of Fermi level (data not shown).

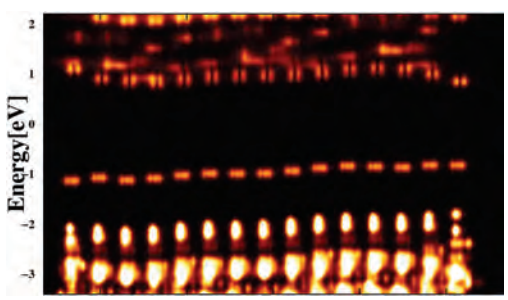


Figure 5. Local density of states of 15 bp $(GC)_{15}$ pair dry DNA

3.2.2 Solvated DNA

The presence of solvent has a substantial effect on DNA. Polarized water molecules cause phosphate groups in DNA backbone to deprotonate making DNA negatively charged. Water molecules screen that charge and thus contribute to stability of the double helix DNA. Another factor is the presence of ions in solution; positively charged ones – usually Na^+ - are electrostatically attracted to the DNA and may play a role in DNA conductivity.

In experiment by Guo, et al.[11], DNA was initially solvated in ionized solution and subsequently partially dried with N_2 gas. Thus the measured DNA likely contains trapped water molecules and positively charged ions, and thus has a charged back-bone.

We have investigated solvated DNA for the case of 4 bp. Including the first solvation shell (60 water molecules) and 8 sodium ions to balance the DNA charge, we find a transmission peak of 3.0×10^{-04} located at -0.08 eV below Fermi level. We also find that the distance of sodium atoms from the negatively charged group in DNA is important; if they are too close they effectively pacify DNA and the conductivity decreases close to that of dry DNA.

Conductance plots for different distances of positive ions relative to the negatively charged groups are shown in Figure 6. In agreement with a previous study[22] we find that the HOMO state of the DNA is pinned to the Fermi level of the leads. Note that the maxima shift towards Fermi level with increasing ion distance and thus should be relevant for lower biases.

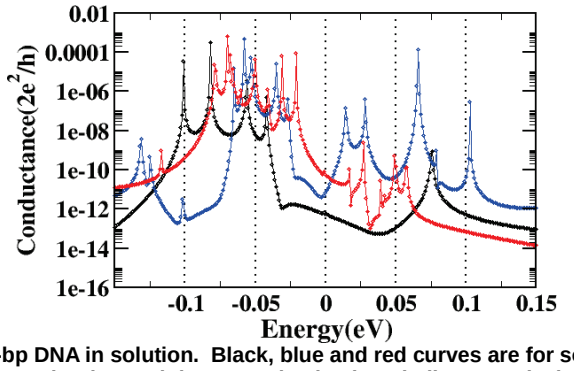


Figure 6. Transmission through 4-bp DNA in solution. Black, blue and red curves are for sodium ions being at the first and second or beyond the second solvation shell, respectively.

To elucidate effect of water, we calculated the local density of states of deprotonated DNA with counterions but without water and compared it to that of solvated DNA. This is shown in Figures 7–8. While this, admittedly, is an artificial system, the comparison shows that the presence of water does not affects the HOMO states at the Fermi level, suggesting that solvent does not play a direct role in DNA conductivity.

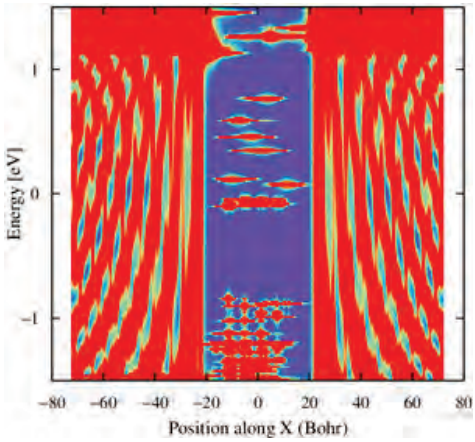


Figure 7. Local density of states of CNT-DNA-CNT in solution. The DNA includes four GC pairs; 60 water molecules and 8 sodium ions are placed in the first solvation shell.

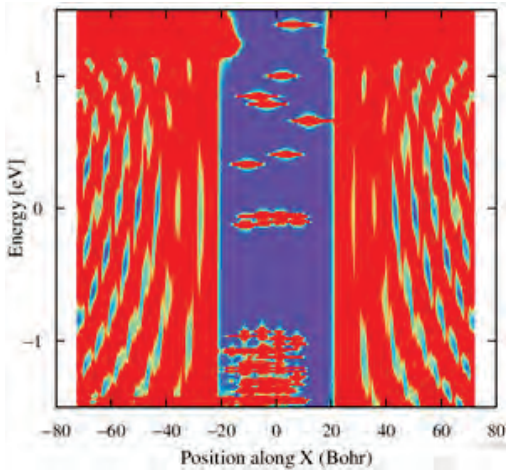


Figure 8. Local density of states of CNT-DNA-CNT. The sodium ions are in the first solvation shell and no water molecules are included.

3.2.3 Sequence and Mismatch Effect

Our work also addressed dependence of electron transport in DNA on base pair sequence and their mismatch. The results are summarized in Table 1 in terms of conductance maxima for two different positions of sodium ions. We find that introducing AC mismatch into the sequence substantially decreases conductivity in qualitative agreement with experiment.[11]

While the discrepancy between experimental and theoretical results for electron transport in DNA persists at present, our work on has uncovered several important aspects of this process. It was found that charged character of DNA in solution causes dramatic increase of its conductivity. The presence of positively charged ions is also important and their distance from DNA has a large influence on conductivity. In agreement with experiment, we find that the presence of a mismatch in DNA sequence causes substantial reduction in conductivity.

Table 1. Transmission peak of DNA with different sequence and ion locations

	Ions in first solvation shell	Ions in second solvation shell
GGGG CCCC	3.0E-04 G_0	4.5E-04 G_0
GAGG CTCC	2.5E-04 G_0	2.6E-04 G_0
GAAG CTTC	9.3E-05 G_0	7.1E-04 G_0
GAGG CCCC	7.5E-07 G_0	4.0E-05 G_0

References

1. Flynn, P.F., *Meeting the Energy Needs of Future Warriors*, National Academies Press, Washington D.C., 2004.

2. Chu, B., X. Zhou, K. Ren, B. Neese, M. Lin, Q. Wang, F. Bauer, and Q.M. Zhang, *Science*, 313, 334, 2006.

3. Ranjan, V., L. Yu, M. Buongiorno-Nardelli, and J. Bernholc, *Phys. Rev. Lett.*, 99, 047801, 2007.

4. Eley, D.D. and D.I. Spivey, *Transactions of the Faraday Society*, 58, 411, 1962.

5. Porath, D., G. Cuniberti, and R. Di Felice, *Long-Range Charge Transfer in DNA II*, vol. 237, G. B. Schuster, Ed. Springer Berlin/Heidelberg, pp. 183–228, 2004.

6. Marshall, A. and J. Hodgson, *Nature Biotechnology*, 16, 27, 1998.

7. de Pablo, P.J. F. Moreno-Herrero, J. Colchero, J. Gómez Herrero, P. Herrero, A.M. Baró, P. Ordejón, J.M. Soler, and E. Artacho, *Phys. Rev. Lett.*, 85, 4992, 2000.

8. Zhang, Y., R.H. Austin, J. Kraeft, E.C. Cox, and N.P. Ong, *Phys. Rev. Lett.*, 89, 198102, 2002.

9. Porath, D., A. Bezryadin, S. de Vries, and C. Dekker, *Nature*, 403, 635, 2000.

10. Yoo, K.-H., D.H. Ha, J.-O. Lee, J.W. Park, J. Kim, J.J. Kim, H.-Y. Lee, T. Kawai, and H.Y. Choi, *Phys. Rev. Lett.*, 87, 198102, 2001.
11. Guo, X., A.A. Gorodetsky, J. Hone, J.K. Barton, and C. Nuckolls, *Nat Nano*, 3, 163, 2008.
12. Giannozzi, P., et al., http://www.quantum-espresso.org.
13. Baroni, S., S. de Gironcoli, A. Dal Corso, and P. Giannozzi, *Rev. Mod. Phys.*, 73, 515, 2001.
14. Fu, H. and L. Bellaiche, *Phys. Rev. Lett.*, 91, 057601, 2003.
15. Sai, N., K.M. Rabe, and D. Vanderbilt, *Phys. Rev. B*, 66, 104108, 2002.
16. Antons, A., J.B. Neaton, K.M. Rabe, and D. Vanderbilt, *Phys. Rev. B*, 71, 024102, 2005.
17. Briggs, E.L. D.J. Sullivan, and J. Bernholc, *Phys. Rev. B, 54*, 14362, 1996.
18. Fattebert, J.L. and J. Bernholc, *Phys. Rev. B*, 62, 1713, 2000.
19. M. Buongiorno Nardelli, J. L. Fattebert, and J. Bernholc, *Phys. Rev. B*, 64, 245423, 2001.
20. Lu, W., V. Meunier, and J. Bernholc, *Phys. Rev. Lett.*, 95, 206805, 2005
21. Pemmaraju, C.D., I. Rungger, X. Chen, A.R. Rocha, and S. Sanvito, *Phys. Rev. B*, 82, 125426, 2010.
22. Mallajosyula, S.S., J.C. Lin, D.L. Cox, S.K. Pati, and R.R.P. Singh, *Phys. Rev. Lett.*, 101, 176805, 2008.

High-Fidelity Large Eddy Simulations of Supersonic Jet Noise for Complex-Geometry Nozzles

Guillaume A. Brès, Frank Ham, and Hung Lê
Cascade Technologies, Inc., Palo Alto, CA
{gbres, ham, hle}@cascadetechnologies.com

Joseph W. Nichols and Sanjiva K. Lele
Center for Turbulence Research, Stanford University, Stanford, CA
{jwn, lele}@stanford.edu

John T. Spyropoulos
US Naval Air Systems Command (NAVAIR), Patuxent River, MD
john.spyropoulos@navy.mil

Abstract

High-fidelity large eddy simulations (LES) are performed for the prediction of far-field noise from hot supersonic turbulent jets issued from complex nozzles. The first study focuses on round nozzles with chevrons at operating conditions relevant to tactical aircraft. The unsteady flow processes and shock/turbulence interactions in a hot, supersonic, over-expanded jet are investigated with the high-fidelity unstructured LES framework developed at Cascade Technologies. In this study, the complex geometry of the nozzle and chevrons are explicitly included in the computational domain, using unstructured grids and adaptive mesh refinement. Noise predictions are compared to experimental results carried out at the United Technologies Research Center for the same nozzles and operating conditions. The comparisons show excellent agreement in terms of spectra shape and levels for both the near-field and far-field noise. Additionally, a faceted military-style nozzle with a straight ramp diffuser developed at General Electric is considered in a second study. The chosen operating conditions correspond to a strongly heated, over-expanded supersonic jet. The high-fidelity LES reveals for the first time direct evidence of N-shaped waves embedded in the turbulent jet shear layers, linked to the phenomenon of "crackle" noise.

1. Introduction

The operation of modern tactical aircraft from aircraft carrier decks exposes military personnel to extreme noise environments that can result in significant and permanent hearing loss. Unlike commercial airplanes, the propulsion systems of combat aircraft are based on low bypass ratio turbo-jet engines. The exhaust jets are supersonic and hot, especially with the augmenter/afterburner in operation. In addition, because exhaust nozzles at take-off are typically operated in an over-expanded condition, the resulting pressure mismatch at the nozzle exit produces a shock cell system within the jet plume. Shock/turbulence interactions generate additional jet noise components, including broadband shock-associated noise, tonal screech noise, and in some cases, crackle noise.

Chevroned nozzles (see Figure 1) are known experimentally to reduce peak jet noise levels, although the detailed physics of this mechanism are still not fully understood. Further in-depth experimental investigations are costly and challenging, especially for heated supersonic pressure-mismatched jets relevant to military applications. In this context, large eddy simulation (LES), along with advancement in high-performance computing, is emerging as an accurate yet cost-effective computational tool for prediction of such high-speed propulsive jets and their acoustic fields.

To simulate the effect of chevrons on jet flow and noise, early approaches excluded the nozzle geometries from the simulations, to avoid the challenges of generating complex body-conformal meshes. Shur, et al. (2006) used an emulation procedure where the presence of the chevrons was modeled with a combination of sources at the nozzle exit. A few recent numerical studies have started investigating the effects of chevrons with the geometry included in the computational domain. Xia, et al. (2009) performed implicit large eddy simulations (ILES) of a subsonic cold jet flow exhausting from a round chevron nozzle using conformed multi-block structured mesh. The same configuration was investigated

independently by Uzun & Hussaini (2009) and by Paliath, et al. (2011) using ILES with multi-block, overset structured meshes. In contrast, Du & Morris (2011) used an immersed boundary method, rather than body-conformal mesh, to represent the chevron geometries. While the current results are encouraging overall, low-dissipative schemes are ultimately required for accurate predictions of jet noise from chevron nozzles. Here, high-fidelity unstructured LES is ideally suited for such aeroacoustic applications involving complex geometries.

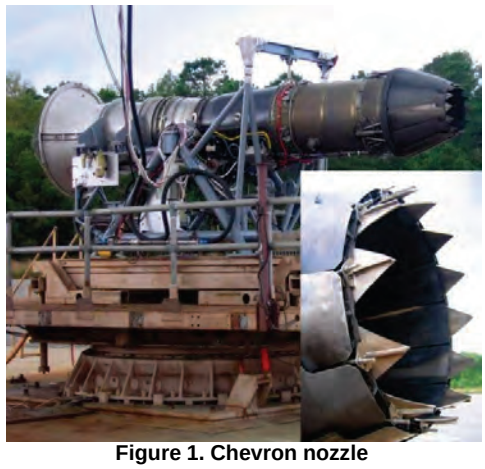


Figure 1. Chevron nozzle

The present work reports some recent simulation results for complex nozzles at operating conditions relevant to tactical aircraft. The first study focuses on a heated supersonic, over-expanded jet issued from a round nozzle with chevrons. The second study investigates the "crackle" noise phenomenon for a faceted military-style nozzle with a straight ramp diffuser, operating at strongly heated, over-expanded supersonic conditions. All the simulations are performed with the high-fidelity unstructured LES framework developed at Cascade Technologies. In both studies, the complex geometry of the nozzle is explicitly included in the computational domain, using unstructured grids and adaptive mesh refinement.

2. Numerical Methodologies

The LES software used here is composed of pre-processing tools (i.e., mesh adaptation), a flow solver "Charles," and post-processing tools. The mesh adaptation module produces high-quality yet economical unstructured grids suitable for capturing turbulence dynamics. The flow solver utilizes a low-dissipation numerical scheme designed to produce accurate results on unstructured meshes, in particular in the presence of hanging nodes and other transition type elements. The database generated by LES is post-processed for statistical analysis of flow and noise as well as for flow visualization.

The large eddy simulations described in this report were performed with the flow solver "Charles." Charles solves the spatially-filtered compressible Navier-Stokes equations on unstructured grids using a novel control-volume based finite volume method where the flux F is computed at each control volume face using a blend of a non-dissipative central flux and a dissipative upwind flux, i.e.,

$$F = (1-\alpha)F_{\text{central}} + \alpha F_{\text{upwind}}$$

where $0 \le \alpha \le 1$ is a blending parameter. This blending approach is often the basis of implicit approaches to LES, where the blending parameter is selected as a global constant with a value large enough to provide all the necessary dissipation (and potentially much more). Here, Charles does *not* use the "implicit" LES approach—an explicit sub-grid scale model is used to model the effect of the unresolved scales on the resolved flow. Furthermore, to minimize numerical dissipation relative to implicit LES approaches, the value of α is allowed to vary spatially such that it can be set to zero in regions where the grid quality is good and the scheme based on the central flux is discretely stable and non-dissipative. In regions of less-than-perfect grid quality, however, the central scheme can introduce numerical instabilities that must be prevented from contaminating/destabilizing the solution by locally increasing α. One significant advantage of this approach is that the blending parameter is purely grid-based, and can be pre-computed based on the operators only.

Because the underlying numerical method has minimal numerical dissipation, it is critical to employ a sub-grid model to account for the physical effects of the unresolved turbulence on the resolved flow. Different modeling options are available in the code. For the large eddy simulations reported in this work, we used the Vreman (2004) model with

constant coefficient set to the recommended value of c = 0.07, and constant turbulent Prandtl number Pr_t = 0.9 to close the energy equation.

Shocks, like sub-grid scale turbulence, are also sub-grid phenomena and thus require modeling to account for their effect on the resolved flow. However, unlike sub-grid scale turbulence, they are localized in the flow and a surgical introduction of modeling is potentially more appropriate. Charles uses a hybrid Central-ENO scheme to simulate flows involving shocks. The scheme has three pieces: a central scheme (described previously), a shock-appropriate scheme and a hybrid switch. For the scheme appropriate for computing a flux across a shock, Charles uses a fully unstructured 2nd-order ENO method to perform reconstructions (Shi, et al., 2002), and the HLLC approximate Riemann solver to compute the flux (Harten, et al., 1983). The hybrid switch, which detects where shocks are present in the flow and activates the shock-appropriate scheme, is based on the method developed originally by Hill & Pullin (2004), where the magnitudes of the smoothness parameters computed as part of the ENO reconstructions are compared to identify the presence of flow discontinuities.

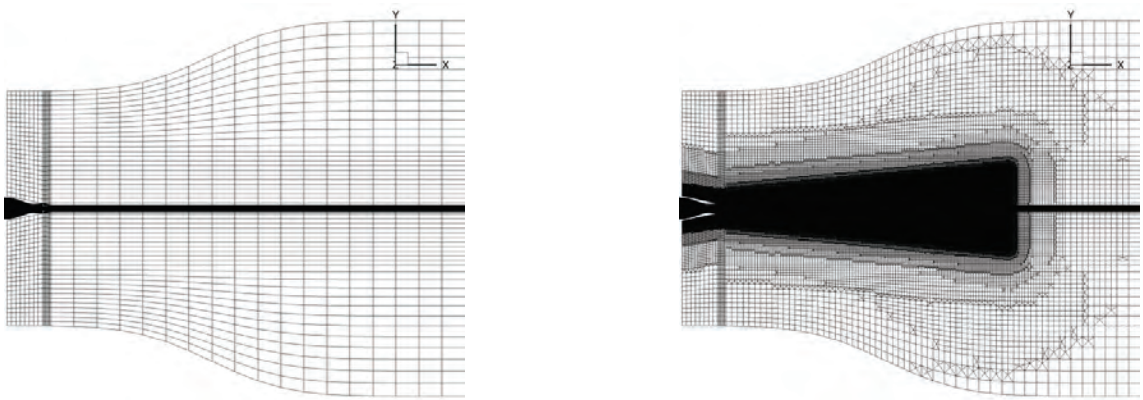
For far-field noise predictions, the accurate propagation of the small amplitude acoustic fluctuations from the near-field source region to the far-field microphones within the computational domain would be prohibitively expensive. The Ffowcs Williams-Hawkings (FW-H) (1969) equation is one of the most commonly used methods to overcome this difficulty. Sound at a far-field location can be computed from flow information on an arbitrarily-shaped surface S and the volume-distributed sources outside of S. As long as the flow is irrotational outside of the FW-H surface S, the computationally expensive (quadrupole) volume integral can be neglected, yielding an efficient procedure for calculating the far-field noise. In addition, jet flow configurations are typically considered in a fixed laboratory frame and, in the current study, in the presence of a uniform coflow. Since the distance between the noise sources (i.e., stationary surface enclosing the jet) and the observers (i.e., far-field microphones) is fixed and time-independent, there is no Doppler effect and an efficient frequency-domain formulation (Lockard, 2000) can be used. The details of the far-field noise module developed for the prediction of jet noise are presented in the work of Brès, et al. (2012a).

The generation of a high quality mesh for a complex geometry remains a pacing issue in high-fidelity flow simulation. LES places strict requirements on mesh resolution, element quality, and even the level of allowed mesh anisotropy. In addition, LES meshes are often too large to conveniently manage on even a high-end workstation. “Adapt" is a massively parallel tool recently developed in Cascade’s solver infrastructure that gives the user detailed control over the local mesh resolution in their grid. The underlying parallel refinement algorithm can refine elements locally to match a target length scale. This target length scale can vary throughout the domain, and can even be different in each direction. Specification of the target length scale can come from expert knowledge of the problem, a desired mesh size limited based on compute resources, or even from a solution on an unadapted or partially adapted mesh. Different from other adaptation tools, the directional refinement algorithm in “Adapt” only refines elements in the direction or directions necessary to meet target length scale requirements, dramatically reducing the overall mesh size and also preventing the addition of stiffness to the problem due to excessively small elements. The “Adapt” tool also provides a surface projection algorithm to respect non-planar mesh boundaries during refinement, ensuring accurate representation of the underlying geometry, and allowing the use of very coarse grids as a starting point.

3. Chevron Nozzle

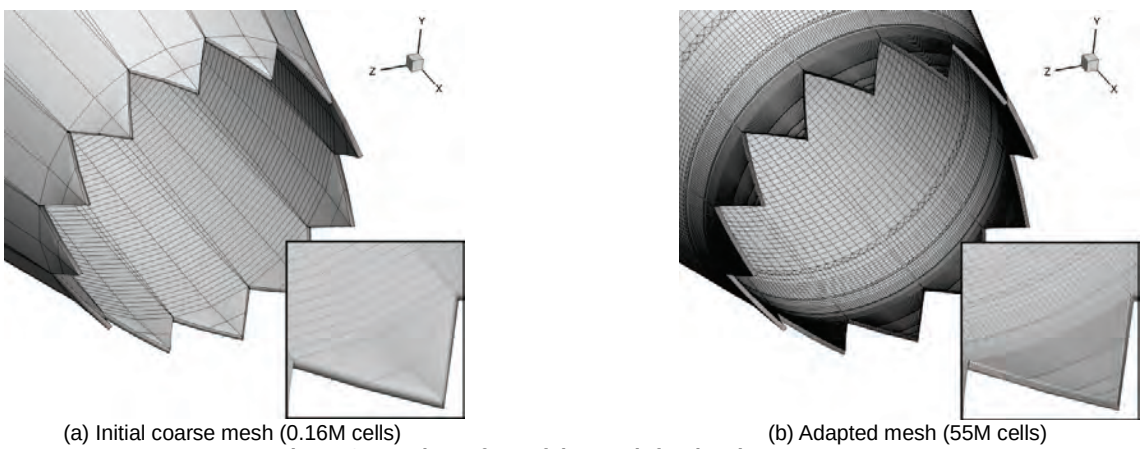
The simulation setup, referred to as $A3_{chevron}$, reproduces the heated over-expanded jet (B122 ODC, where ODC stands for “Off-Design with Chevrons”) conditions tested at the UTRC acoustic research tunnel facility. The configuration consists of 12 chevrons (6° penetration) appended to the round nozzle exit. The round nozzle geometry corresponds to UTRC converging-diverging (CD) nozzle designed using a method-of-characteristics to provide ideally-expanded (i.e., shock-free) flow at the nozzle exit. The design Mach number of the nozzle is M_d = 1.5. The operating conditions are defined by the nozzle pressure ratio $NPR = P_0/P_\infty$ = 2.97 and the nozzle temperature ratio $NTR = T_0/T_\infty$ = 2.53, where the subscripts 0 and ∞ refer to stagnation properties inside the nozzle and static properties in the ambient fluid, respectively. These conditions produce a heated over-expanded supersonic jet. The fully expanded jet Mach number is $M_j = U_j/c_j$ = 1.35 and the acoustic Mach number is $M_a = U_j/c_\infty$ = 1.84. The jet temperature ratio is T_j/T_∞ = 1.85, and the Reynolds number is Re_j = 130000. Here the subscript j refers to the fully expanded jet conditions, and c is the speed of sound, and U is the mean streamwise jet velocity. The computational results are compared to available near-field and far-field experimental data at the same operating conditions. The study presented in this paper was performed using “blind” comparisons, in the sense that all the simulations, post-processing and analysis were performed without prior knowledge of the experimental data.

In the present simulations, the chevrons are explicitly included in the computational domain with a body-fitted mesh. The new adaptation approach has been used to generate the grids, as shown in Figure 2. The starting point is a very coarse mesh with a paved core containing about 0.16M control volumes. Several embedded zones of refinement are defined and enforced by the adaptation tool. The bulk of the mesh for the downstream jet plume is discretized by cells with edge lengths Δx no greater than 0.04 D where D is the jet diameter. The finest resolution corresponds to the near nozzle and chevrons with $\Delta x < 0.005D$. Here, azimuthal resolution is not sacrificed with increasing distance away from the centerline, as is the case with purely cylindrical meshes. The current adaptive refinement strategy promotes mesh isotropy in the acoustic source-containing regions, as previous studies have shown that it yields the best performance (Nichols, et al., 2011b). A smoothing algorithm is then applied to avoid sharp grid transitions between different refinement zones. Finally, a surface projection algorithm is used to respect non-planar mesh boundaries during refinement, ensuring accurate representation of the underlying geometry. The chevron surface mesh is shown in Figure 3. While triangle elements appear in the figures, this is merely an artifact of the plotting algorithm's treatment of cells with subdivided faces. The actual mesh used for the calculations contained approximately 55 million unstructured control volumes, mostly hexahedral.



(a) Initial coarse mesh (0.16M cells) (b) Adapted mesh (55M cells)

Figure 2. Midsection cut of the mesh for the chevron case



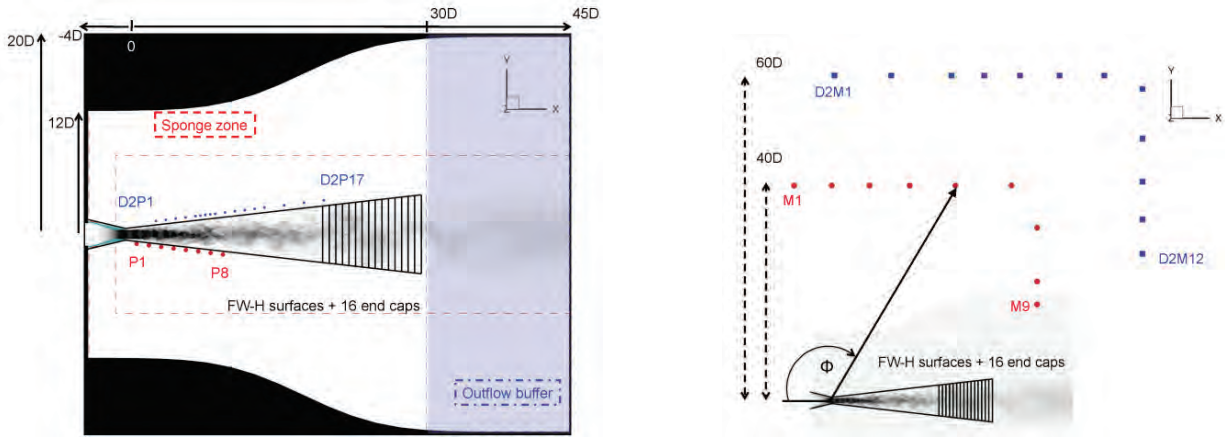
(a) Initial coarse mesh (0.16M cells) (b) Adapted mesh (55M cells)

Figure 3. Nozzle surface of the mesh for the chevron case

The jet configuration is presented in Figure 4, including the FW-H surface used to compute the far-field sound. The method of end-caps (Shur, et al., 2005) is applied from $x = 20D$ to $x = 30D$ to eliminate the uncorrelated spurious sound. A constant plug-flow is applied to the inlet of the nozzle such that the desired Mach number and the temperature ratio are achieved at the nozzle lip. It should be noted that we assume that the flow issued from the nozzle is laminar.[1] Consequently, the grid resolution inside the nozzle is only adequate for a laminar flow. Figure 4 also show the two sets of near-field (D2P1-D2P17) and far-field (D2M1-D2M12) microphones in the UTRC experimental configuration, which match the LES microphone stations. The jet exhausts into an anechoic chamber which is subject to a wind tunnel flow with

[1] In the experiment, it is not known whether the flow issued immediately after nozzle is laminar or turbulent.

Mach number M_∞ = 0.1, to avoid overheating the chamber. The same M_∞ = 0.1 free-stream flow condition is used in the simulation. Additional details about the computational setup can be found in the work of Brès, et al. (2012b).



(a) Computational domain (b) Far-field setup

Figure 4. Schematics of LES computational domain and FW-H surfaces with far-field microphones

An instantaneous snapshot of the temperature field is shown in Figure 5. The adaptive mesh refinement strategy allows us to place sufficient azimuthal resolution close to the nozzle lip so that the shear layers support complicated three-dimensional disturbances indicative of turbulence very close to the nozzle exit, as shown by Figure 5. The turbulence is visible at x/D = 0.5 and fully developed at one diameter from the nozzle exit. The turbulent flow mixes as it develops farther downstream of the nozzle. The shock cells and detailed turbulent structures are clearly visible in this figure. The development of these fine turbulent structures, from the chevrons nozzle exit to the end of the potential core, is highlighted in Figure 6. In this figure, the isosurface of vorticity magnitude is colored by the streamwise velocity.

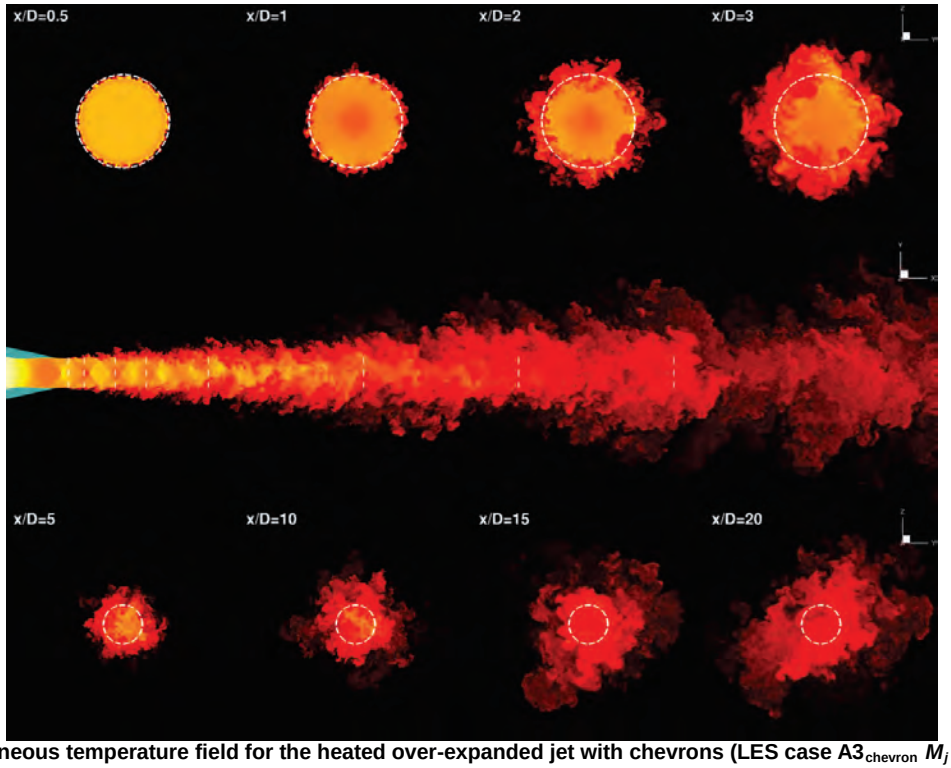


Figure 5. Instantaneous temperature field for the heated over-expanded jet with chevrons (LES case A3$_{chevron}$ M_j = 1.35, T_j = 1.85, Re_j = 130000). The axial cuts of temperature are shown in top and bottom subfigures. The colored contours show T from 1(black) to 2.5 (white). The dashed circular line corresponds to the nozzle exit outline.

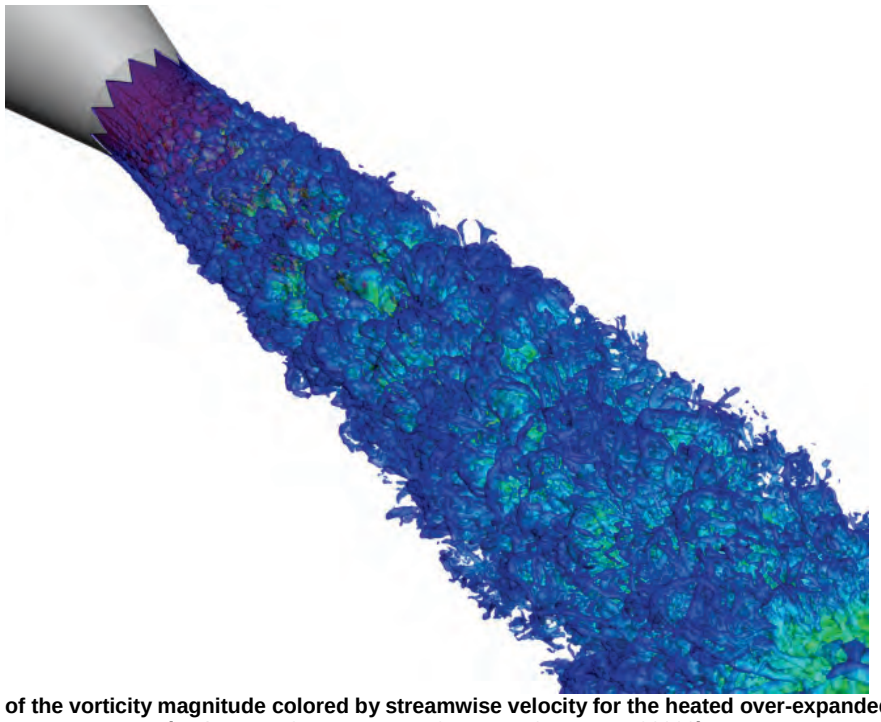


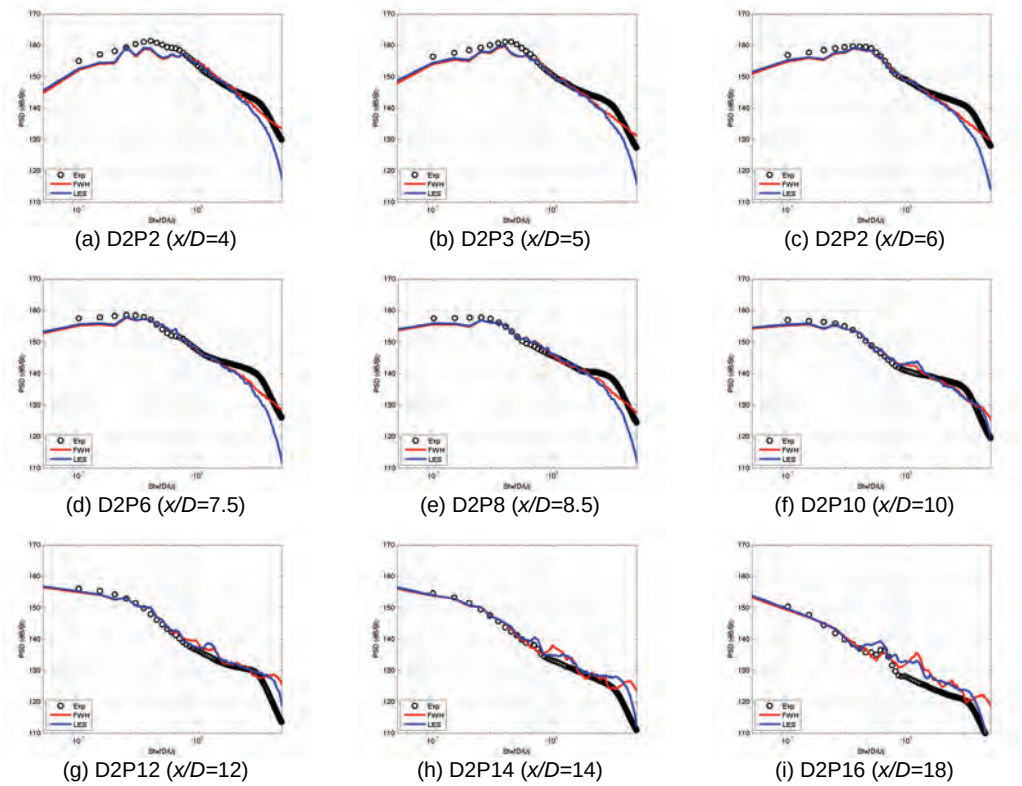
Figure 6. Isosurface of the vorticity magnitude colored by streamwise velocity for the heated over-expanded jet with chevrons (LES case A3$_{chevron}$, M_j = 1.35, T_j = 1.85, Re_j = 130000)

Two different methods are used to compute the noise at the experimental near-field probes. The first method corresponds to the computation and propagation of the pressure fluctuations directly in the simulation (i.e., recording the pressure signal from simulation at specified locations). The second method makes use of the FW-H solver previously discussed. Since there is no "far-field" approximation in the current algorithm, aside from the quadrupole terms being neglected, the far-field noise solver can be used to predict the noise at any location outside of the FW-H surface. Here, all the near-field probes are located outside the FW-H surface as shown in the schematic in Figure 4(a). To reproduce the presence of the coflow in the LES computation, the FW-H predictions are performed with the same coflow, namely $M_\infty = 0.1$.

Figure 7 presents the near-field sound spectra and the blind comparison to experimental measurements at nine different probe stations. Overall, the comparison is very good, from the low frequency of $St = f D/U_j = 0.05$ up to the experimental cutoff frequency of $St \approx 3$. Similar agreement is obtained for the other probes locations. For most probes, the FW-H prediction matches exactly the LES results up to $St = 2$. The under-prediction at the higher frequency is likely due to the contributions from the quadrupole (i.e., volume terms) which are not negligible in the near-field but not computed with the current formulation.

Figure 8 shows the blind comparisons of the far-field sound at the experimental microphone stations shown in Figure 4(b). The spectra span the range of dominant jet noise directivity angles in the aft quadrant: from the most aft measurement angle 150° corresponding to microphone D2M11 to 90° at microphone D2M1. Note that the far-field experimental results are corrected for atmospheric attenuation and but not for refraction through the open jet shear layer. The effect of the coflow can also be included in the present FW-H calculations, such that a uniform coflow $M_\infty = 0.1$ is assumed between the noise sources and the microphones. The agreement is again very good for these aft angles, in particular for microphones D2M9 (ϕ = 140°), D2M10 (ϕ = 145°), and D2M11 (ϕ = 150°) which correspond to the loudest angles. For most microphones, the agreement extends from $St = 0.05$ to $St = 5$.

The majority of calculations were carried out on "garnet" and "raptor," CRAY XE6 machines at DoD supercomputer facilities in ERDC and AFRL. The simulations are run on 2,048 cores and yield large transient flow field databases, between 10TB and 30TB each. These databases reside in archival storage at ERDC and are currently continuing to be used for post-processing and analysis, probing the physics of jet noise production and the effect of chevrons.

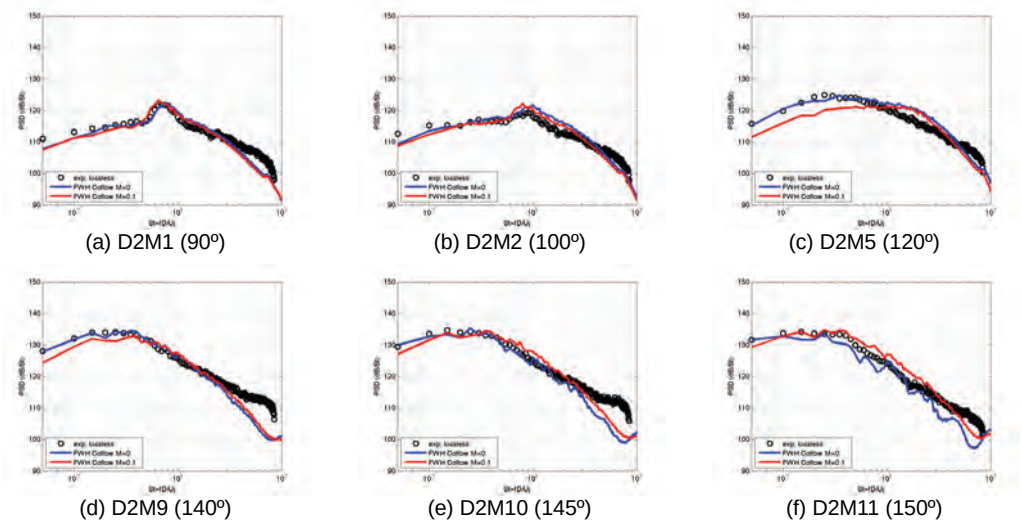


(a) D2P2 (*x/D*=4) (b) D2P3 (*x/D*=5) (c) D2P2 (*x/D*=6)

(d) D2P6 (*x/D*=7.5) (e) D2P8 (*x/D*=8.5) (f) D2P10 (*x/D*=10)

(g) D2P12 (*x/D*=12) (h) D2P14 (*x/D*=14) (i) D2P16 (*x/D*=18)

Figure 7. Sound spectra at different near-field probe locations for the heated over-expanded jet case with chevron A3$_{chevron}$ (M_j = 1.35, T_j = 1.85, Re_j = 130000). Legend – symbol: experiment; blue line: LES; red line: FWH.



(a) D2M1 (90º) (b) D2M2 (100º) (c) D2M5 (120º)

(d) D2M9 (140º) (e) D2M10 (145º) (f) D2M11 (150º)

Figure 8. Sound spectra at different far-field probe locations for the heated over-expanded jet case with chevron A3$_{chevron}$ (M_j = 1.35, T_j = 1.85, Re_j = 130000). Legend – symbol: experiment; blue line: FWH Coflow, M=0; red line: FWH Coflow, M=0.1.

4. Crackle Noise

Crackle noise from heated supersonic jets is characterized by positive pressure spikes arriving intermittently at observer locations. Because it involves sudden compressions of the acoustic pressure, crackle is extremely obnoxious when it occurs. There has been much scientific debate as to its cause over the last 37 years, beginning with a paper by Ffowcs Williams, et al. (1975). The LES code Charles was recently used by Nichols, et al. (2012a) to investigate the

sources of crackle noise generated by a hot over-expanded supersonic jet. The present calculations have opened a new window into this difficult problem. The section below summarizes their results.

Figure 9 shows a military-style nozzle with faceted straight-ramp diffuser used in the simulation. This geometry matches exactly that of an experimental nozzle designed by GE and used by Martens, et al. (2011) for acoustic testing. The design Mach number is M_d = 1.65; and the simulation used a nozzle pressure ratio $NPR = P_0/P_\infty = 4.0$ and nozzle temperature ratio $NTR = T_0/T_\infty = 3.65$. This operating point corresponds to a fully expanded jet Mach number $M_j = 1.56$ and acoustic Mach number $M_a = 2.44$ and so produces a strongly heated, over-expanded supersonic jet. The actual mesh used for the calculations contained 331 million purely hexahedral control volumes. Most of these simulations were run on 4,000 cores and up to 8,192 cores on "raptor," a CRAY XE6 machine at DoD supercomputer facilities in AFRL.

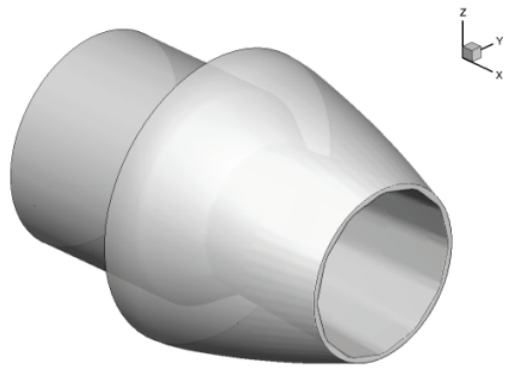


Figure 9. Military-style nozzle designed by GE for acoustic testing, and used for the crackle simulations

A snapshot of the instantaneous temperature and pressure fields at an axial cross-section of the flow is presented in Figure 10. The turbulent structures in the jet plume are made visible by the black-red-yellow color scale of the temperature contours. Also observed downstream of the nozzle exit is the double-diamond pattern of the shock cells. Outside of the jet plume, the blue-white color scale of the pressure contours is used to visualize the near-field acoustics generated from the jet. In the figure, the green circles indicate stations at which time histories of pressure are recorded.

Figure 11 shows a pressure time signal of outer probe 2 in which "crackle events" are identified by the spikes in pressure, the strongest of which is denoted by the red arrow. At this instant, the pressure exceeds four standard deviations (dashed lines) above the signal mean (solid line). These pressure excursions are observed to consist of a sudden rise followed by a more gradual decline, leading to the "N-shaped" waves characteristic of "crackle."

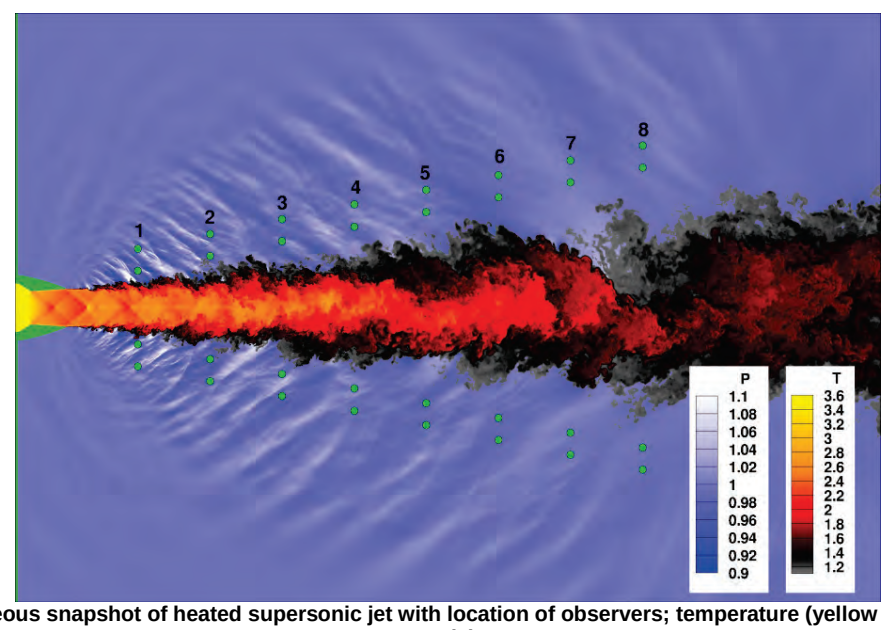


Figure 10. Instantaneous snapshot of heated supersonic jet with location of observers; temperature (yellow scale), pressure (blue scale)

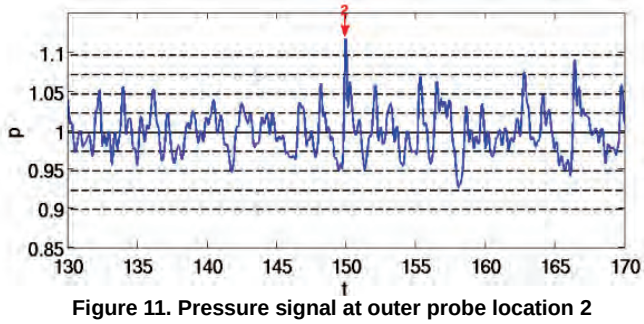


Figure 11. Pressure signal at outer probe location 2

In the following figure, Figure 12, visualization of the entire flow field at 3 instants leading to the crackle event in Figure 11 is shown with temperature in grayscale, while pressure contours are in color. The cyan circle indicates the location of outer probe 2. The magenta curve is the contour where the acoustic Mach number is equal to unity, separating the supersonically moving fluid inside from the subsonic fluid outside. In all three frames (a)-(c), crackle waves are identified by the green color and a particularly strong crackle wave is indicated by the black arrow. The green crackle wave progresses in the time sequence and reaches the probe location in frame (c) which coincides with the pressure spike identified in Figure 11.

With these simulations, Nichols, et al. (2012a) show that crackle waves emerge from the jet turbulence, and appear to be produced at indentations of the sonic line where cool ambient fluid is entrained into the jet core. The N-shaped waves associated with crackle are embedded directly in the jet turbulence, and therefore are not generated by nonlinear propagation effects alone. Figure 13 shows a three dimensional visualization of the crackle waves, which was produced on the Utility Server at AFRL.

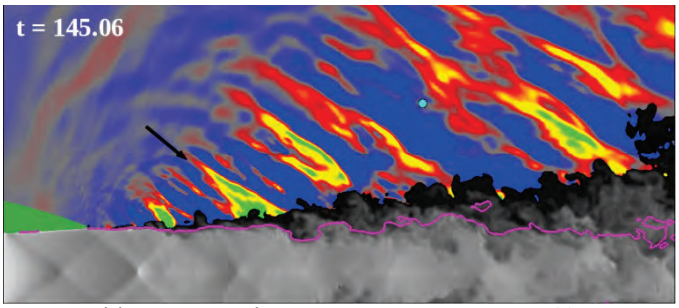
In general, simulations of a crackling jet present a significant computational challenge, as they correspond to intense operating conditions (i.e., strongly heated, pressure-mismatched, supersonic jet) and require treatment of a range of scales from the large energy-containing eddies to thin shocklets associated with N-shaped waves. In this context, the unstructured LES solver “Charles” shows very encouraging results and much potential towards meeting this challenge.

5. Summary of Results and Future Plans

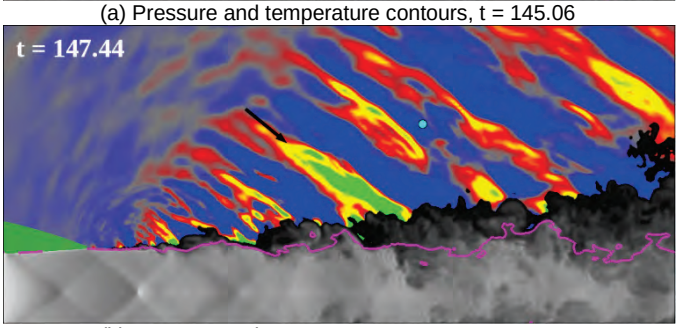
In the previous seven months, we have used approximately 5.5 million core-hours at the ERDC and AFRL supercomputer centers, running high fidelity LES calculations for the prediction of supersonic jet noise from complex geometries. In particular, we have focused our efforts upon chevron nozzles as well as military-style faceted nozzles. Chevron (or serrated) nozzles are known to significantly alter the noise produced by supersonic jets, although the physical mechanism responsible is not yet fully understood. Our high-fidelity LES provides access to complete flow fields (see Figure 6) allowing us to shed new light on this problem. Blind comparisons between simulation and laboratory experiment were very encouraging in terms of both near-field and far-field acoustic spectra (Brès, et al., 2012a; Brès, et al., 2012b).

A portion of the CPU time was also used to complete a campaign of rectangular supersonic simulations that was begun during a previous DoD HPCMP Capability Applications Project discussed at the previous HPCMP User's Group Conference (Nichols, et al., 2011c). In particular, a rectangular jet with and without chevrons was simulated (Nichols, et al., 2011; Nichols, et al., 2012b). Experience gained from these calculations aided in the mesh design and acoustic post-processing steps used for the round chevron jets presented above.

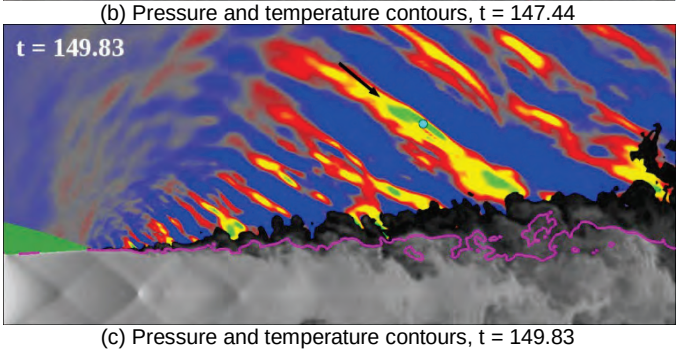
Finally, unstructured LES was applied to a turbulent supersonic jet issuing from a faceted military-style nozzle designed by GE Global Research (Nichols, et al., 2012a). Simulations are performed at operating conditions matching the strongest heating level able to be tested in the laboratory environment, as well as at operating conditions involving an even higher level of heating. Skewness levels were measured from the direct near acoustic field at various axial and radial positions. It was found that the pressure perturbations were highly skewed very close to the jet. Furthermore, we observed crackle waves emerging directly from the jet turbulence, with characteristic N-shaped pressure waveforms. Therefore, while nonlinear propagation effects may eventually steepen waves yet further, it is not a necessary component, as steep crackle waves are produced directly at the jet.



(a) Pressure and temperature contours, t = 145.06



(b) Pressure and temperature contours, t = 147.44



(c) Pressure and temperature contours, t = 149.83

Figure 12. Identification of a crackle event, outer probe 2; (a)-(c) sequence of instantaneous temperature contours (grayscale) and pressure contours (color) leading up to the crackle event

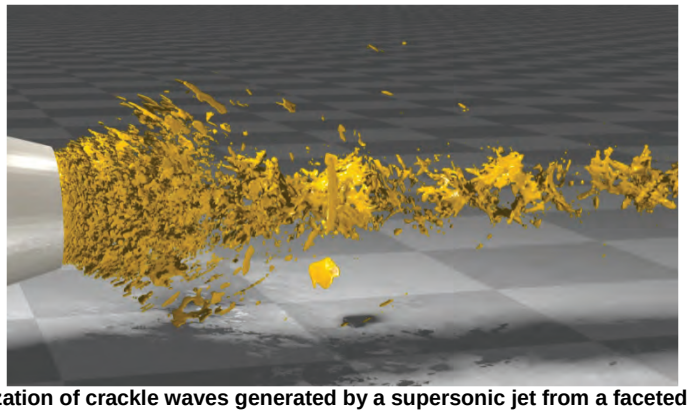


Figure 13. 3D visualization of crackle waves generated by a supersonic jet from a faceted military-style nozzle

The round chevron nozzles have been simulated using 55M unstructured grid cells. In order to ascertain the accuracy of the results, we believe it will be necessary to introduce one more level of refinement, resulting in as many as 200M grid cells. To complement these high fidelity runs, a series of lower fidelity runs are also planned, whereby different mesh clustering schemes are tested for this complicated geometry, in order to obtain best practices for acoustic prediction for general 3D geometries. Recently it has been shown (Zaman, 2011) that the initial state of the boundary layer as it leaves the nozzle can influence the far-field sound. In order to address this issue in our own simulations, we will run a series of simple conical nozzles corresponding to a laboratory experiment with and without a wall model active inside of the nozzle.

Analysis of the crackle simulations has raised questions about the role of temperature in the formation of N-shaped waves in the jet turbulence. For a fixed nozzle pressure ratio, heating increases the supersonic jet velocity relative to the ambient fluid. It is also possible to increase the jet velocity by increasing the nozzle pressure ratio without heating. Such additional simulation will be performed to investigate whether temperature plays a key role in crackle, or whether it is determined by jet velocity alone. While the mesh used for the calculations presented in this paper contained 331 million control volumes, a refined simulation with size increased to 650 million is currently ongoing. It is also planned that NAVAIR personnel will perform simulations of realistic nozzle geometries that cannot be shared with academic partners. We expect these computations to be similar in size and complexity to the faceted military-style nozzles used for the crackle simulations.

Acknowledgments

This research is supported by NavAir STTR Contract Number N68335-11-C-0026. Computational resources were provided by the HPC Challenge Project NAWCP30952C5E, "Large Eddy Simulations of Supersonic Jet Noise and Combustor & Augmentor Spray Atomization." The majority of calculations were carried out at DoD supercomputer facilities in ERDC and AFRL. The CD round nozzle, chevron geometries and corresponding experimental results were provided by Robert H. Schlinker, John Simonich and Ramons A. Reba of United Technologies Research Center. The geometry of the military-style nozzle with faceted straight-ramp diffuser was provided by Steve Martens of GE Global Research.

References

Brès, G.A., Nichols, J.W., Lele, S.K., and Ham, F., "Towards Best Practices for Jet Noise Predictions with Unstructured Large Eddy Simulations," *42nd AIAA Fluid Dynamics Conference and Exhibit*, New Orleans, LA, 2012a, June 2012.

Brès, G.A., Nichols, J.W., Lele, S.K., Ham, F., Schlinker, R.H., Reba, R.A., and Simonich, J., "Unstructured Large Eddy Simulation of a Hot Supersonic Over-Expanded Jet with Chevrons." *18th AIAA/CEAS Aeroacoustics Conference*, 2012b, Colorado Springs, CO, June 2012.

Du, Y. and Morris, P.J., "Noise simulations of supersonic hot jets for chevron nozzles," *AIAA paper 2011-2787*, 2011.

Ffowcs Williams, J.E., and Hawkings, D.L, "Sound Generation by Turbulence and Surfaces in Arbitrary Motion," *Royal Society of London Philosophical Transactions Series A*, 264, pp. 321–342, May 1969.

Ffowcs Williams, J.E., Simpson, J., and Virchis, V.J. "Crackle: an annoying component of jet noise," *Journal of Fluid Mechanics*, 71(2), pp. 251–271, 1975.

Harten, A., Lax, P., and Van Leer, B., "On upstream differencing and godunov-type schemes for hyperbolic conservation laws," *SIAM review*, 15, pp. 35–61, January 1983.

Hill, D.J. and Pullin, D.I., "Hybrid tuned center-difference-WENO method for large eddy simulations in the presence of strong shocks," *Journal of Computational Physics*, 194(2), pp. 435–450, 2004.

Lockard, D., "An efficient, two-dimensional implementation of the Ffowcs Williams and Hawkings equation," *Journal of Sound and Vibration*, 229(4), pp. 897–911, 2000.

Martens, S., Spyropoulos, J.T., and Nagel, Z., "The effect of chevrons on crackle – engine and scale model results," *Proceedings of the ASME Turbo Expo 2011*, GT2011-46417, Vancouver, Canada, 2011, June 2011.

Nichols, J.W., Ham, F.E., Lele, S.K., and Moin, P., "Prediction of supersonic jet noise from complex nozzles," *Center for Turbulence Research Annual Research Briefs*, Stanford University, 2011a.

Nichols, J.W., Ham, F.E., and Lele, S.K., "High-fidelity large-eddy simulation for supersonic rectangular jet noise prediction," *AIAA paper 2011-2919*, 2011b.

Nichols, J.W., Ham, F.E., Lele, S.K., Khalighi, Y., and Spyropoulos, J.T., "High-Fidelity LES of Asymmetric Supersonic Jets," *Proceedings of the HPCMP User Group Conference*, Portland, OR, 2011c, June 2011.

Nichols, J.W., Lele, S.K., Ham, F.E., Martens, S., and Spyropoulos, J.T., "Crackle noise in heated supersonic jets," *In Proceedings of the ASME Turbo Expo 2012*, GT2012-69624, Copenhagen, Denmark, 2012a, June 2012.

Nichols, J.W., Ham, F.E., Lele, S.K., Moin, P., and Bridges, J.E., "Large-eddy simulation for supersonic rectangular jet noise prediction: effects of chevrons," *18th AIAA/CEAS Aeroacoustics Conference*, Colorado Springs, CO, 2012b, June 2012.

Nichols, J.W., Ham, F.E., Lele, S.K., and Bridges, J.E., "Aeroacoustics of a supersonic rectangular jet: Experiments and LES predictions," *AIAA paper 2012*-678, 2012c.

Paliath, U., Shen H., Avancha, R., and Shieh, C., "Large Eddy Simulation for Jets from Chevron and Dual Flow Nozzles," *AIAA paper 2011-2881*, 2011.

Shi, J., Hu, C., and Shu, C., “A technique of treating negative weights in WENO schemes,” *Journal of Computational Physics*, 175, pp. 108–127, January 2002.

Shur, M.L., Spalart, P.R., and Strelets, M.K., “Noise prediction for increasingly complex jets. Part I: Methods and tests,” *International Journal of Aeroacoustics*, 4(3-4), pp. 213–246, 2005.

Shur, M.L., Spalart, P.R., Strelets, M.K., and Garbaruk, A.V., “Further step in LES-based noise prediction for complex jets,” *AIAA paper 2006-485*, 2006.

Uzun, A. and Hussaini, Y.M., “Simulation of Noise Generation in Near-Nozzle Region of a Chevron Nozzle Jet,” *AIAA Journal*, 47(8), 2009.

Vreman, A., “An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications,” *Physics of Fluids*, 16:3570, January 2004.

Xia, H., Tucker, P.G., and Eastwood, S., “Large-eddy simulations of chevron jet flows with noise predictions,” *International Journal of Heat and Fluid Flow*, 30, pp. 1067–1079, 2009.

Zaman, K.B.M.Q., “Effect of nozzle exit conditions on subsonic jet noise,” *AIAA paper 2011-2704*, 2011.

Hot Supersonic Jet Noise Predictions using High-Resolution Large Eddy Simulation

James P. Erwin, Neeraj Sinha, and Gregory P. Rodebaugh
Combustion Research and Flow Technology (CRAFT Tech), Pipersville, PA
{jerwin, sinha, grodebaugh}@craft-tech.com

Abstract

Predicting and reducing supersonic jet noise emissions is an important goal for the United States Navy. Large-Eddy Simulation (LES) has become a widely-used tool to predict the noise from supersonic jets. Three LES calculations are described that solve an over-expanded, hot, supersonic jet flowfield with and without chevrons for noise reduction. The principal objective is to use LES to guide the development and testing of noise reduction devices for full scale, realistic engines with complex geometries. Using the Ffowcs-Williams and Hawkings equation method, the noise in the far-field was computed for the baseline and chevron jet with an LES of approximately 5.5 million cells. The chevron devices are shown to reduce the noise in all directions to the jet axis, and by up to 2–3 dB in the downstream direction. Non-Contiguous Block Interfaces (NCBI) is implemented to refine the structured grid around the chevrons and reduce computational cost in the far-field of the jet. The finer LES grid consists of approximately 128 million cells with 117 blocks and was distributed among 1,958 processors on both the Navy DoD Supercomputing Resource Center (DSRC) Cray XT5 and the Arctic Region Supercomputing Center (ARSC) Cray XE6 system. The calculation has used approximately 880,000 hours of allocation between both machines, and is shown to have better than linear speedup across 1,958 processors. Solving the flow on both machines was possible by transferring a restart file between them when hours became available. The time-averaged flow agrees reasonably well with the lower-resolution chevron calculation. However, differences are identified in the turbulent flowfield and also the far-field noise, especially in the upstream noise direction. This calculation is significant to the Navy because it allows for the development of noise reduction devices which require extensive grid resolution to affect the noise-generating mechanisms of the flowfield.

1. Introduction

Large-Eddy Simulation (LES) has become a widely-used tool to predict the noise from supersonic jets. With sufficient grid resolution, the LES can directly resolve the noise-generating mechanisms in a wide range of turbulent flowfields. The noise sources of supersonic free jets are mostly understood. High-intensity Mach wave radiation is emitted at downstream angles from the jet. This is due to the supersonic convection of turbulent structures relative to the ambient air. Broadband shock-associated noise (BBSAN) is generated when the shock cells in the jet core scatter the turbulence in the shear layer. In addition to these sources, the onset of jet screech and crackle is also able to be captured with LES. Because LES directly predicts the noise generated from resolved turbulence, it is well-suited to predict the impact of noise reduction devices that can be installed on the trailing edge of the nozzle. An inherent feature of LES; however, is the high computational cost associated with spatiotemporally resolving these turbulent structures. With the Department of Defense's (DoD's) High Performance Computing Modernization Program (HPCMP), extremely high-resolution calculations can be completed within days or weeks turnaround time with thousands of processors.

The United States Navy is reducing the noise emitted from hot military gas turbine engines by installing noise reduction devices on the nozzle trailing-edge. Before the devices are fabricated, tested, and ultimately installed on operational aircraft, it would be beneficial to evaluate them using LES to help guide the design process. In this paper, we describe a methodology that we use to predict the far-field noise for a hot supersonic military gas turbine engine. The calculations are performed for a baseline faceted-geometry, as well as geometry with representative chevrons. The chevrons are infinitesimally thin, equilateral, non-penetrating triangles and were not optimized or designed though any US Navy-supported program. They are only intended for a demonstration of the computational tools available. Three

calculations are described that were completed on both the NAVY DoD Supercomputing Resource Center (DSRC) Cray XT5 and the Arctic Region Supercomputing Center (ARSC) Cray XE6 system. The CRAFT CFD® LES solver, as well as the far-field acoustic code has been described by the authors in detail (see Sinha, 2009, 2010; Erwin, 2011, 2012). This paper will primarily focus on the logistics of the high-resolution calculations, as well as the computational resources required to complete them. Development cases were run on grids consisting of only 5.5 million cells and will be referred to as the low-resolution (LR) cases. The high-resolution (HR) calculation presented consists of 128 million cells and was performed with 1,958 processors.

2. Numerical Method

For the LES calculations, we use a structured-grid Navier-Stokes solver, CRAFT CFD® which has been used extensively for the evaluation of jet noise attenuation concepts for military gas turbine engines (Sinha, 2008). A low-dissipation fifth-order upwind biased reconstruction procedure is used with Roe's shock capturing, approximate Reimann solver (Kannepalli, 2002). A Direct Numerical Simulation (DNS) study of subsonic and supersonic turbulent free-shear flow demonstrates the capability of the fifth-order reconstruction scheme (Calhoon, 2002). Unique to CRAFT CFD® is a Hybrid Reynolds-averaged Navier-Stokes (RANS)/LES (HRLES) option (Arunajatesan, 2003) that allows for seamless transition between a k-ε RANS model in the internal nozzle and an LES calculation of the exhaust region. In HRLES mode, the code assumes that every point in the flowfield satisfies an empirical form of the turbulent kinetic energy spectrum. The eddy viscosity from the k-ε RANS model is then reduced based on the portion of this energy spectrum sustainable by the local mesh resolution.

The acoustic signatures at far-field microphone locations are calculated through a transformation of the near-field LES solution. The transformation solves the Ffowcs Williams and Hawkings (FW-H) equation using Farassat's Formulation 1A (FF1A) (Farassat, 2007). The FW-H equation is an exact rearrangement of the continuity and Navier-Stokes equations into an inhomogeneous wave equation. An in-depth analysis of the FW-H and Kirchhoff methods is given by Brentner and Farassat (Brentner, 1998). During the LES, flow variables are recorded on a fictitious surface tightly surrounding the shear-layer. In an LES post-processing step, these variables are then used in the FW-H transformation for the far-field noise predictions.

2.1 LES Grid Topology

The identical grid is used for both the baseline and chevron calculations. Figure 1 is an illustration of the grid with the nozzle nacelle and chevrons highlighted yellow. The identical grid can be used for both the chevron and baseline case by modifying the boundary condition along the triangle domains that form the chevron shape. Figure 2 is an illustration of how we simulate the baseline and chevron nozzle by adjusting the boundary condition along the chevron triangles to be inter-block communication or no-slip walls, respectively. We have developed a method to resolve the triangular chevron shape with quadrilateral domains which are required to form a structured- grid topology. Each triangle is divided into three quadrilaterals. In Figure 2(a), the blue-shaded quadrilaterals are the chevron surfaces, and the grey-shaded quads are the valley area between each chevron. Figure 2(b) is the same grid but with the chevron domains now turned from a no-slip boundary condition to an inter-block boundary condition. We then transform, using non-contiguous block interfacing (NCBI), six quadrilateral domains into a parallelogram shape (outlined in red in Figure 2) which can more cleanly be continued away from the chevrons into the far-field of the LES grid.

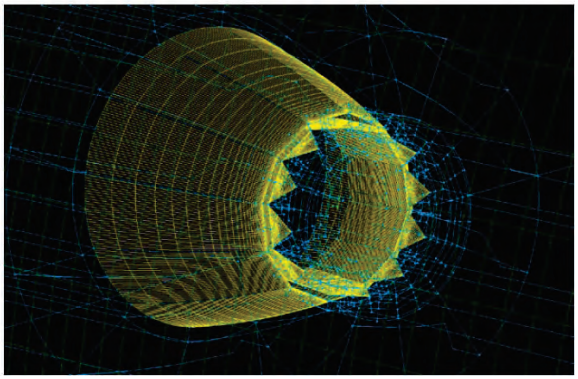


Figure 1. Structured Grid for LES of hot supersonic jet with Chevrons

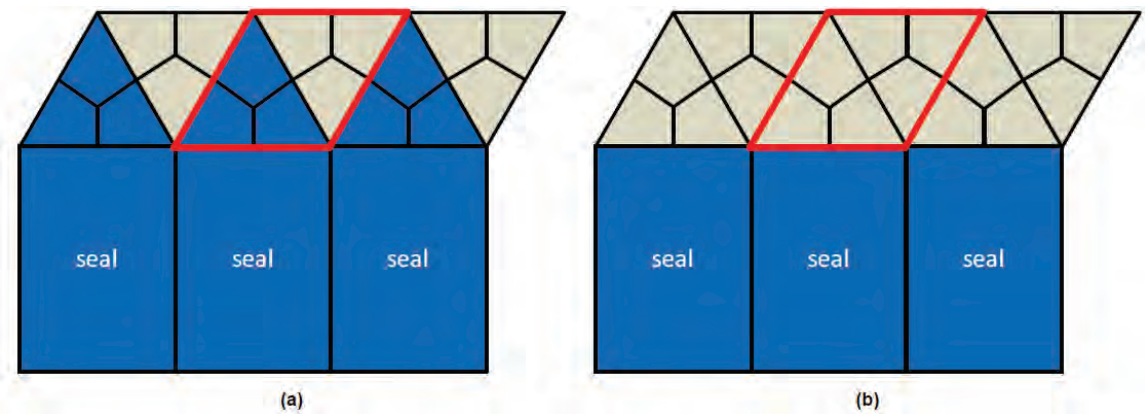


Figure 2. Schematic of grid topology for 3-d Chevron grid. Three seals (blue) and Chevrons (blue) are Unwrapped and shown in planar view. Valleys between Chevrons are shown in grey. A non-contiguous block interface is outlined in red.

The NCBI also allow the resolution around the chevrons to be increased without inherently increasing the resolution in the far-field, which is a traditional limitation of structured-grid topologies. An edge of the chevron was resolved by 111 points for the HR grid compared to only 34 points for the LR grid. A qualitative comparison of the two grid resolutions is shown in Figure 3. A similar type of refinement was performed in the immediate region downstream of the chevrons. In addition to a more densely resolved chevron region, the minimum grid-spacing around the chevrons was reduced to 10% of the minimum spacing in the LR grid. This results in an allowable time-step of roughly $1/10^{th}$ the LR time-step. The grids extended approximately thirty jet diameters (D_J) downstream before we applied a buffer zone to reduce spurious reflections from the outflow boundaries.

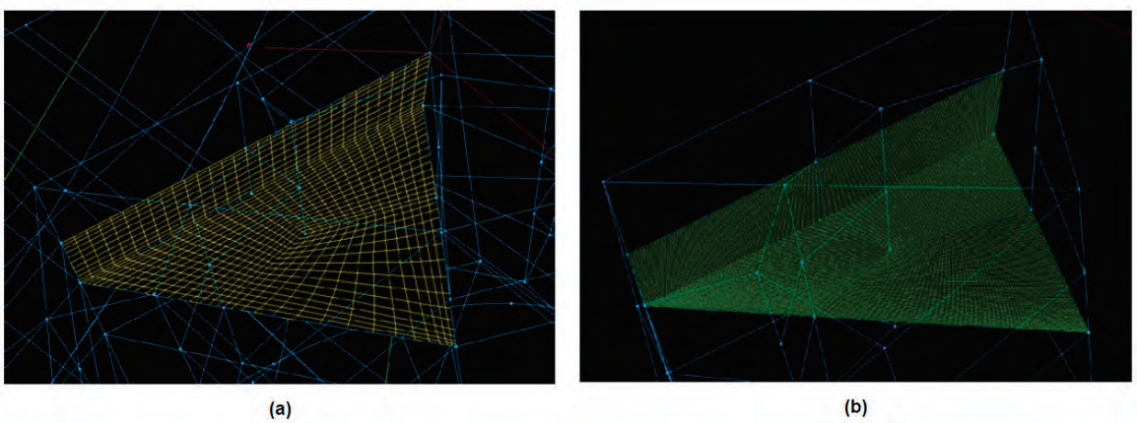


Figure 3. Qualitative grid resolution illustration along Chevron for (a) LR grid (b) HR grid

3. Noise Predictions of Hot Supersonic Military Gas Turbine Engines

In this section we present laboratory scale flowfield and noise predictions for the baseline and chevron jets at low-resolution, as well as chevron case performed at the high-resolution. The jets are over-expanded, operating at approximately Mach 1.55.

3.1 Low-Resolution LES of Baseline versus Chevron Jet

Figure 4 and Figure 5 are instantaneous snapshots of temperature for the baseline and chevron case, respectively. There is a large variation in the shock cell centerline alignment in the baseline case starting at approximately the third cell downstream, whereas the chevrons tend to straighten (align) the shock cells. The movement of the shock cell structure in the baseline case contains a combination of azimuthal and flapping modes, where the chevron jet has a much more stationary core region. Figure 6 and Figure 7 are isometric views of the temperature field for the two cases. The axial cut is located at one jet radius (R_J) downstream. Even for the baseline case, there is a significant amount of shear-layer

turbulence resolved at this location. The axial cut of the chevron jet shows a substantial penetration of the entrained flow into the shear-layer of the jet, due to the pressure imbalance at the nozzle exit. Because the pressure at the nozzle exit is lower than ambient pressure, the entrained flow is pulled through the chevron valley quite strongly.

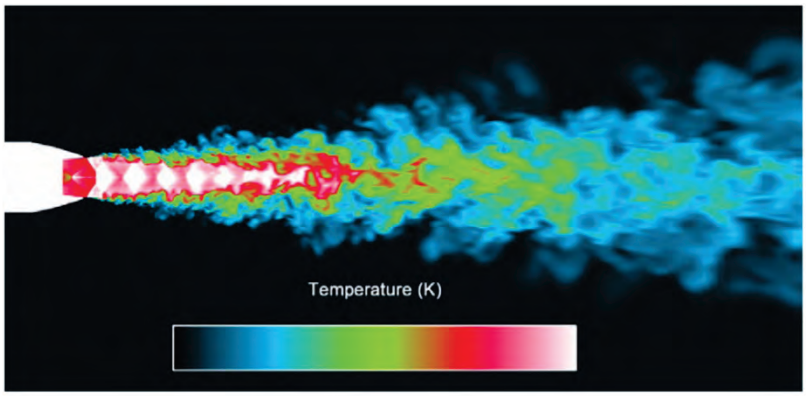


Figure 4. Instantaneous LES temperature (Baseline)

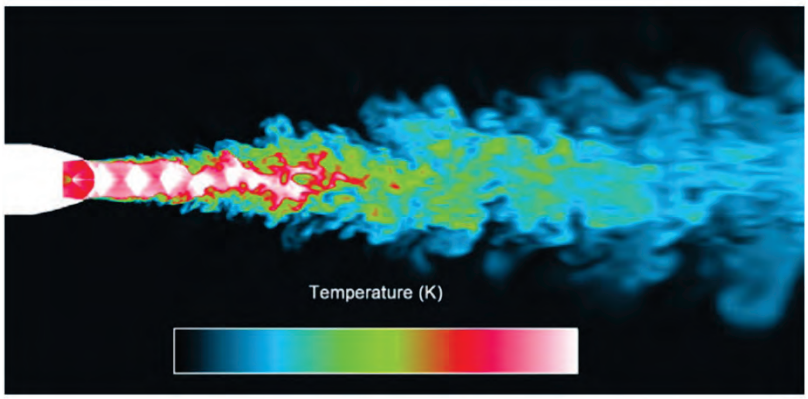


Figure 5. Instantaneous LES temperature (Chevrons)

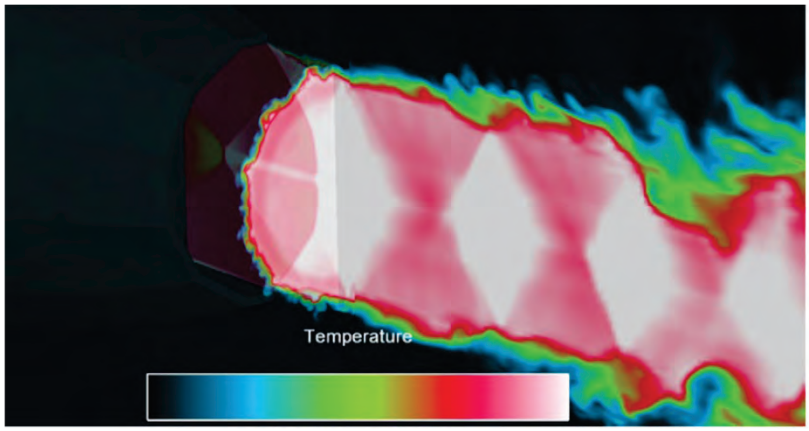


Figure 6. Instantaneous LES Temperature (Baseline, Isometric View)

The pressure RMS contours in Figure 8 indicate that the chevrons may have a significant impact on noise reduction. The peak-levels for the baseline jet are much lower than the chevron jet. The high-spikes along the outer edge of the core for the baseline case are due to the shock cell motion discussed earlier. This phenomenon was similarly found by Nichols, et al. (Nichols, 2012) when performing high-resolution LES of rectangular Mach 1.4 jets. Nichols observed that the peak RMS levels were located along the shear-layer region when the jets were flapping, and more centralized for non-flapping jets.

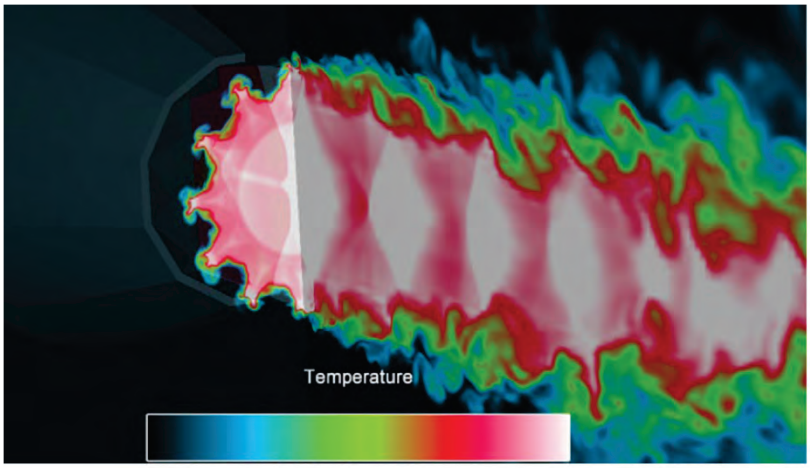


Figure 7. Instantaneous LES temperature (Chevrons, Isometric View)

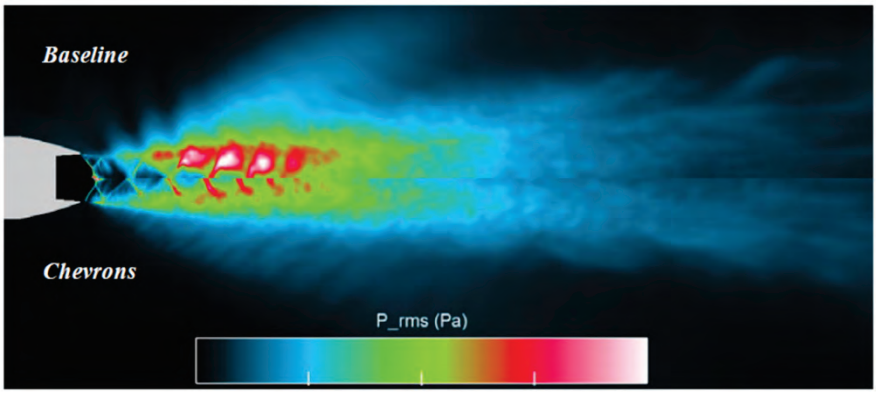


Figure 8. RMS pressure (Baseline vs. Chevrons)

3.2 Low-Resolution Noise Predictions of Baseline versus Chevron Jet

The LES/FW-H method was applied to the baseline and chevron jets. Acoustic data was written on the FW-H surfaces at a sampling rate of 200 kHz for 0.043 seconds. Shown in Figure 9 is the Overall Sound Pressure Levels (OASPL) for the baseline and chevron jet located on a circular arc approximately 100 D_J from the nozzle exit. The downstream direction is zero degrees and the upstream direction is 180 degrees. The chevrons appear to reduce the downstream noise by as much as 3–4 decibels (dB) and the noise in the transverse direction a fairly uniform 1 dB. Narrowband spectral comparisons are made in the upstream (135 degrees) and downstream (40 degrees) directions in Figure 10 and Figure 11, respectively. A BBSAN peak occurs in the upstream direction around 8–9 kHz. One can see the chevrons have an effect on the noise in both directions. In the upstream direction the reduction is most notably in the lower-frequencies, potentially due to the reduction of a low jet-flapping mode. With the chevrons there is significant reduction of broadband noise in the downstream direction, as well as elimination of the screech tone that was predicted for the baseline case at approximately 4.5 kHz. This tone is attributed to the intense flapping of the shock cells and their interaction with the shear-layer, as discussed in the previous section. Figure 12 shows 1/3 Octave spectra in the upstream, transverse, and downstream directions, highlighting the noise reduction across a wide range of frequencies in all directions. Upstream, the noise reduction is most noticeably at frequencies below the BBSAN peak. After the BBSAN peak the levels are quite similar. In the transverse direction, the noise reduction can start to be seen above the BBSAN peak. Finally, in the downstream direction, the reduction is primarily in the higher-frequencies above the peak frequency and also the tone is eliminated.

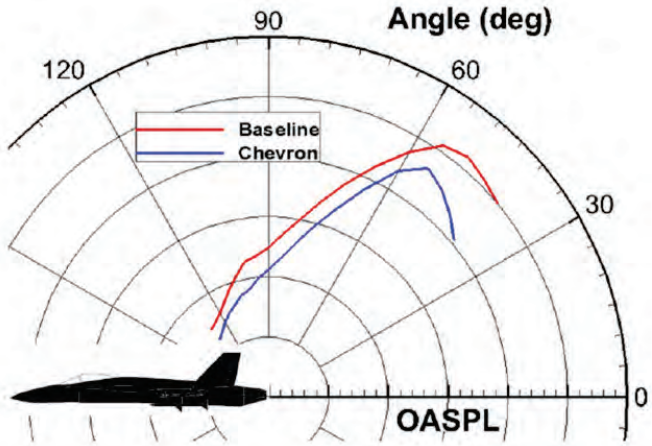


Figure 9. OASPL directivity for laboratory scale hot military gas turbine engine with and without Chevrons

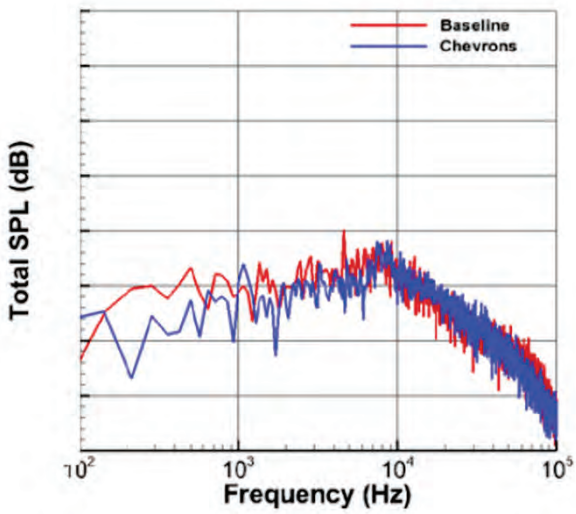


Figure 10. SPL Spectra 135 degrees (Upstream)

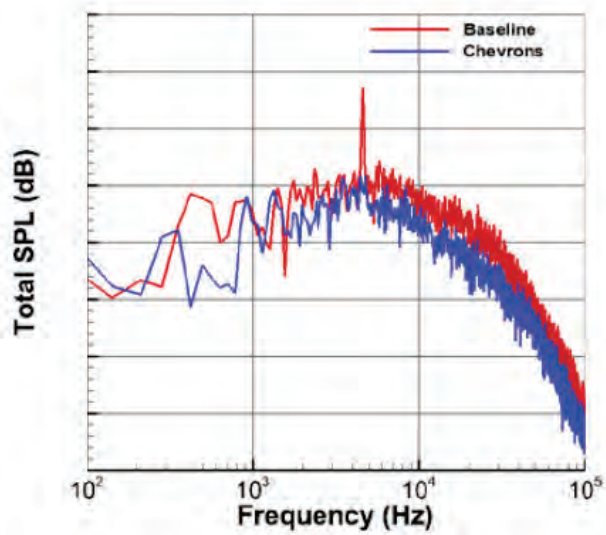


Figure 11. SPL Spectra 40 degrees (Downstream)

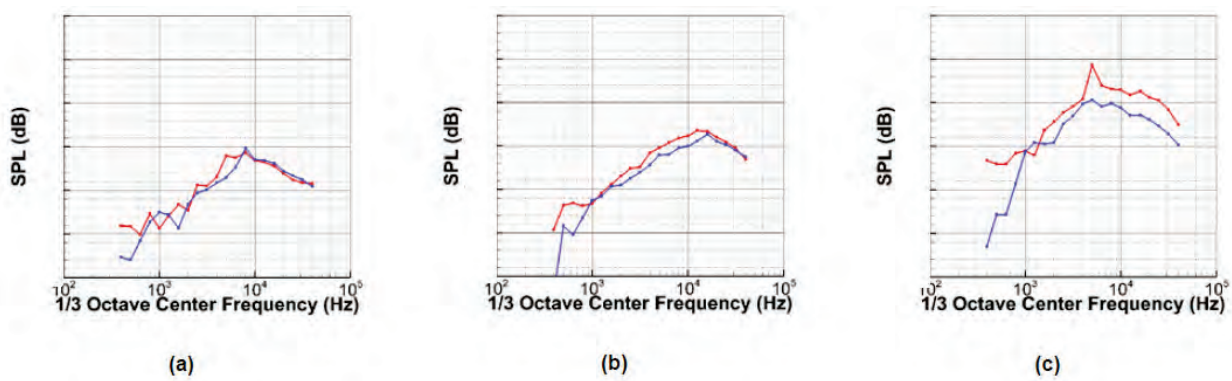


Figure 12. 1/3 Octave SPL spectra for hot military gas turbine engine with and without Chevrons (a) Upstream 115 degrees (b) Transverse 90 degrees (c) Downstream 40 degrees

4. High-Resolution LES of Jet with Chevrons

Many differences in the noise between the baseline and chevron jet were identified at the low-resolution in the previous sections. However, noise reduction devices most likely influence the strength and growth of small-scale turbulence, and requires increasingly fine-LES grid resolution to capture their impact on the flowfield. In this section we present a high-resolution (HR) calculation consisting of 128 million grid-cells. The solution from the LR chevron case was transferred to the HR grid for the flow initialization. The transfer was based on a nearest-node search, and therefore thousands of time-steps were required to flush out the high-frequency modes and error associated with the nearest-node approximation. This still was more efficient than initializing the entire HR flowfield to ambient conditions. Each restart file (8 bytes FORTRAN double-precision) used 6.7GB of disk space and each solution variable (4 bytes FORTRAN single-precision) required approximately 0.48GB per-time-step saved. During this calculation we saved pressure, temperature, Mach number, and sub-grid scale viscosity at approximately 200 time-steps, requiring a total of 384GB of storage for flow data.

Time-averaged LES centerline Mach number plots are shown in Figure 13. The core of the HR calculation is approximately 2–3 D_J longer than the LR calculation. However, the close-up in Figure 13(b) shows that the first 6–7 shock cells agree very closely in magnitude, as well as spacing. The HR calculation has a couple more shock cells present, but they are low in magnitude. The HR calculation predicts a longer core length, as expected, because the growth of large-scale structures, that are known to break down the jet core, is delayed with finer grid-resolution. Instantaneous LES contours of temperature for the HR and LR case are shown in Figure 14. Many more fine-scale features can be seen in the HR calculation, as expected. Also of interest is the additional unsteadiness resolved by the HR calculation in the core. At the first normal shock outside of the nozzle, one can see structures shedding off the centerline of the shock cells. Although the overall penetration depth of the entrained flow in the HR case is similar to the LR case, the finer scales are better resolved, which directly impact the high-frequency noise.

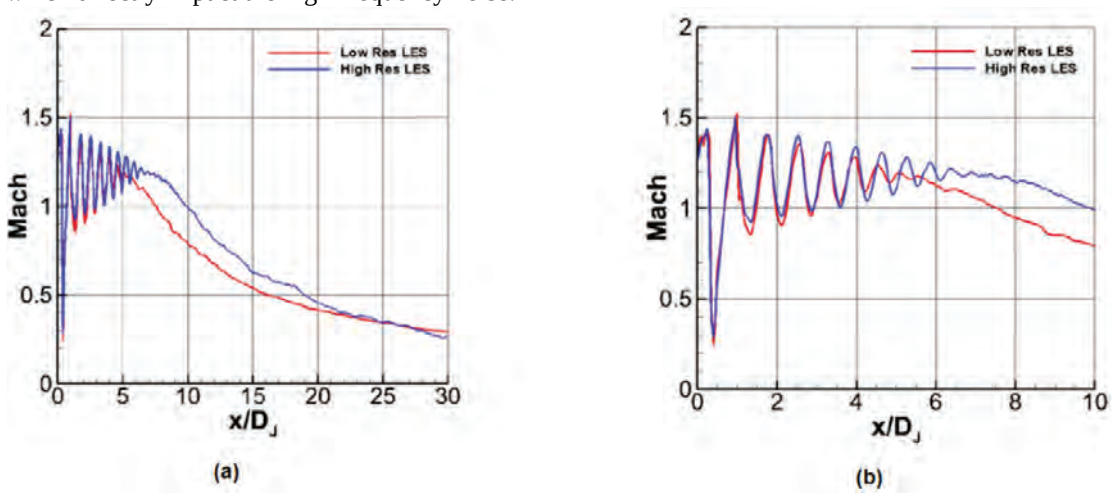


Figure 13. Centerline profile, time-averaged Mach number for Chevron, Fine vs. Coarse LES (a) 0 to 30 D_J (b) 0 to 10 D_J

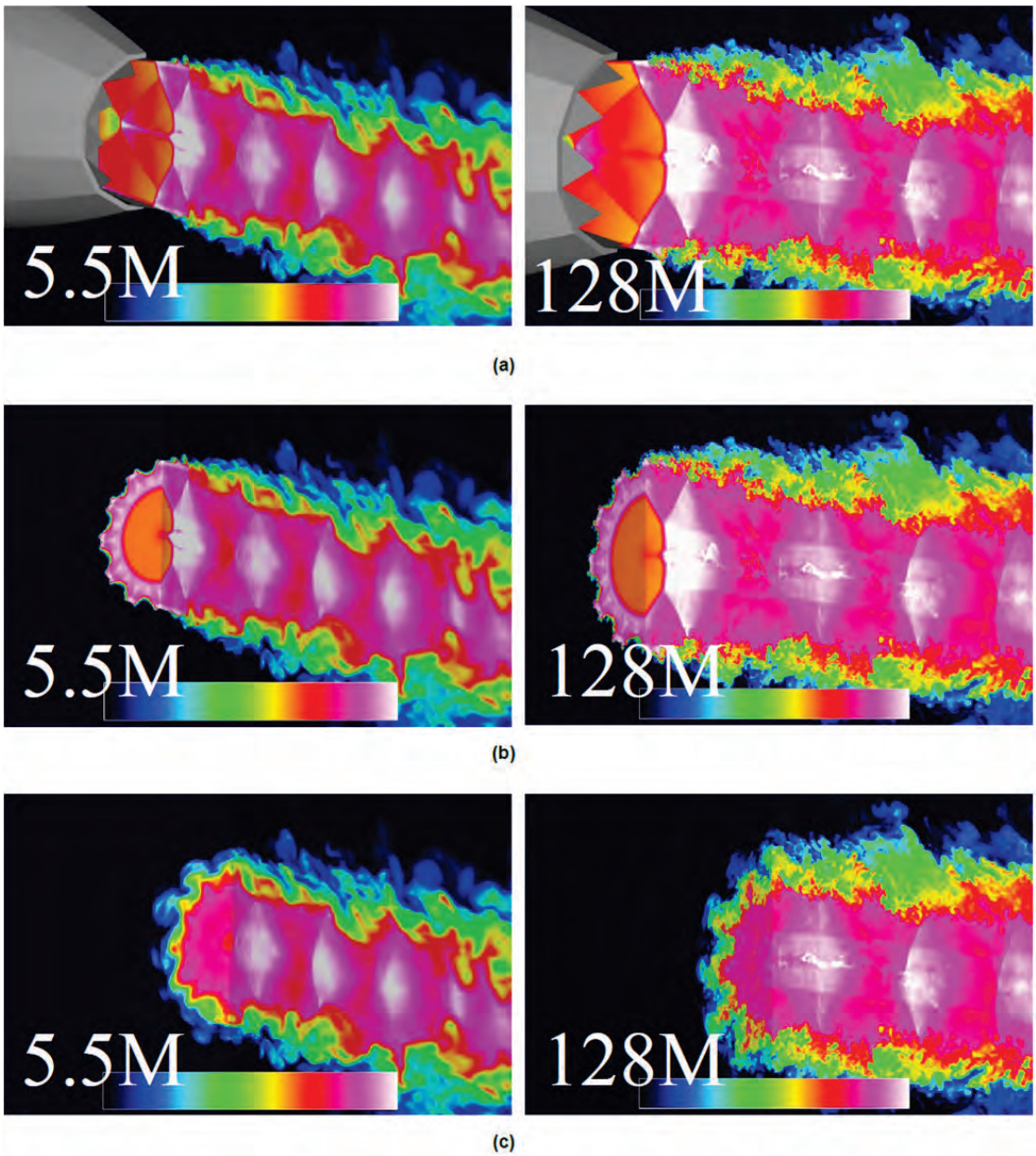


Figure 14. Instantaneous temperature contours for Chevron, LR (5.5M) vs. HR (128M) (a) z-Cut at Centerline (b) z-cut with x-cut at ~0.2 D_J (c) z-cut with x-cut at ~1.0 D_J

4.1 High-Resolution Far-field Noise Predictions

Far-field noise predictions were performed for the HR chevron case. Acoustic data was written to the FW-H surface at a sampling rate of 227 kHz for a total time of 0.00462 seconds. A comparison of the instantaneous pressure field for the LR and HR calculations is shown in Figure 15. High wavenumber content can be seen throughout the entire shear-layer region and around the shocks in the HR calculation. These high-frequency waves are BBSAN. Pressure RMS contours are shown in Figure 16 for both resolutions. The HR calculation has much more distinct RMS regions than the LR calculation, which has a more continuous band of high-RMS throughout the shear-layer. This is also an indication that the HR noise predictions will contain higher frequency content, because each shock cell is producing these high-amplitude peaks in the RMS contours. There are also more peaks resolved further downstream than the LR calculation.

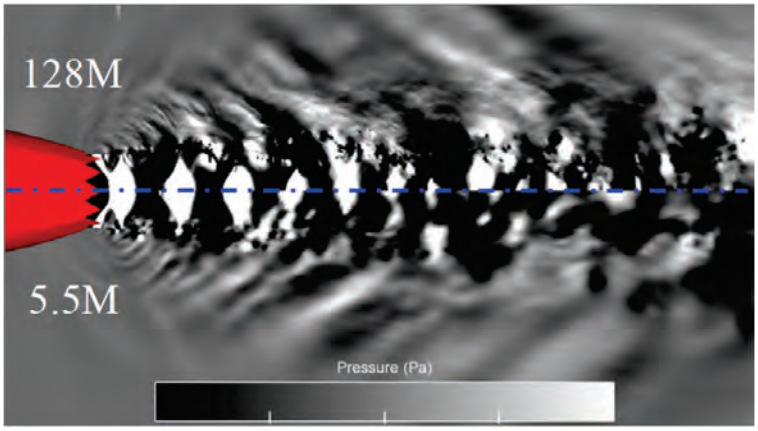


Figure 15. Instantaneous pressure contours for Chevron, Fine vs. Coarse LES

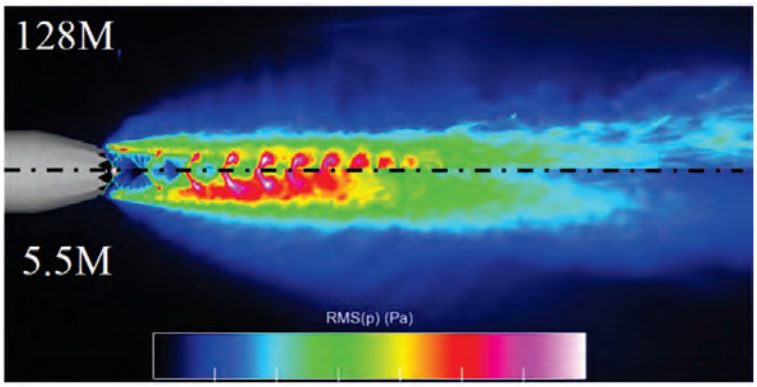


Figure 16. RMS pressure contours for Chevron, Fine vs. Coarse LES

The far-field narrowband spectra levels are compared in Figure 17 for the upstream, transverse, and downstream directions. In the upstream direction it is clear that the HR case increases the noise levels throughout the entire frequency range, and especially at levels above the BBSAN peak. This is attributed to the increased high-wavenumber resolution, especially in the shock cell region. In the transverse direction, the levels also increase above the BBSAN peak, but the levels agree fairly-well below the peak. In the downstream direction where the peak noise-levels exist due to Mach wave radiation, the HR grid only increases the high-frequency content above a Strouhal number of around 2–3, which is well beyond the peak frequency of interest in this direction. It should be noted that the HR calculation was simulated for much less physical time than the LR calculation due to the time-step constraint. Therefore, the low-frequency resolution of the HR predictions is not as low as the LR predictions.

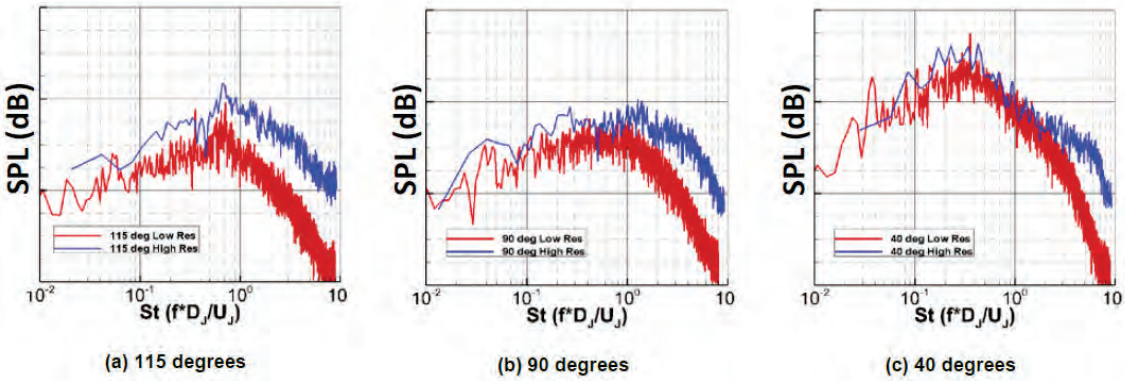


Figure 17. Coarse vs. Fine LES/FW-H noise predictions for hot military gas turbine engine with Chevrons

4.2 Conclusions from High-Resolution Chevron Calculation

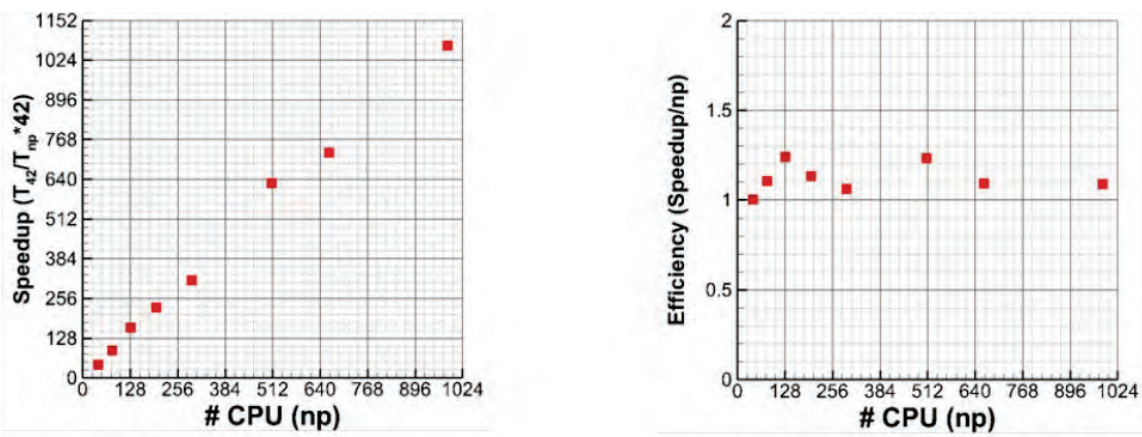
The HR chevron calculation was a successful first attempt at using almost two thousand processors with a coupled LES/FW-H noise prediction. We were able to run the simulation long enough to obtain a time-averaged flowfield comparison with the low-resolution case. Also, the far-field sound levels increase the most in the upstream direction throughout the entire frequency range, which is expected due to much higher resolution of BBSAN. In the downstream direction where the peak noise-level exists, the HR calculation only increased the noise-levels above a Strouhal number of between 2 and 3, which is beyond the peak-level frequency. If the peak noise in the downstream direction is of primary interest when designing noise reduction concepts, the HR grid is most likely not needed. However, if it is necessary to predict high-frequency noise directed upstream, it is clear that the LR grid is insufficient and needs to be refined.

4.3 High Performance Computing Resources

The HR chevron case was simulated on the NAVY Cray XT5 system (Einstein), as well as the ARSC Cray XE6 system (Chugach). The calculation was started locally at CRAFT Tech and subsequently transferred the case to the HPC clusters. We were able to consistently submit and run jobs using 1,958 processors, requesting 5–6 hours of wall-clock time per submission. The lower wall-clock time requests allowed jobs to start sooner, even with the high 1,958 processor count. When the queue slowed on one machine, we transferred the current restart file to the other machine and ran the job there in an attempt to minimize turnaround time.

Each LES time-step using 1,958 processors required approximately 3.1 seconds. This equates to approximately 1.7 CPU hours per-time-step. On Einstein, we were able to use 96% of the 600,000 hours allocated to the jet noise project. On Chugach, we used 51% of the 600,000 hours allocated.

Figure 18 shows the efficiency and speedup of a CRAFT CFD® LES case over a wide range of processors for a 30 million cell grid. The scalability study was performed on the Einstein system with a baseline of 42 minimum processors. We were also able to demonstrate scalability for the HR chevron case using up to 1,958 processors on Einstein, as shown in Figure 19. For this 128M case, we expect the speedup to level-off once the communication cost outweighs the computational cost per processor. When the surface area (inter-block communication) to volume-ratio of each processor increases beyond a certain point, we will not be able to achieve more speedup with more processors.



(a) Parallel speedup (b) Parallel efficiency

Figure 18. Scalability of CRAFT CFD® code on a demonstration 30 million cell case (a) Parallel speedup (b) Parallel efficiency

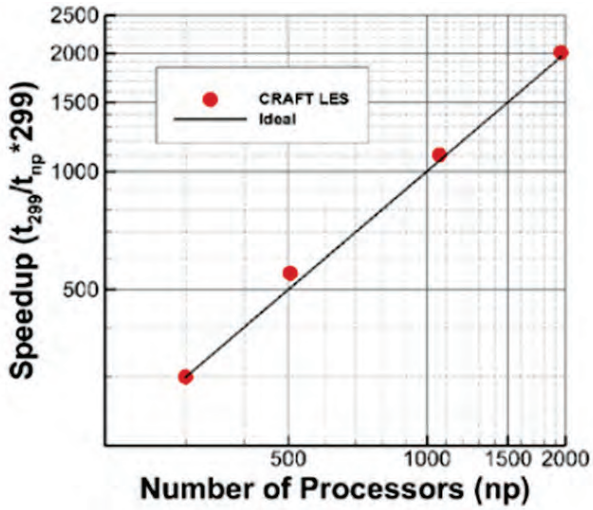


Figure 19. Scalability of CRAFT CFD® code on 128M Chevron case on Cray XT5 system (Einstein)

5. Conclusion

With the DoD's High Performance Computing Modernization Program (HPCMP), high-resolution LES calculations can be completed within days or weeks turnaround time with thousands of processors. In this paper, we describe hot supersonic jet-noise calculations using an innovative grid-topology. Using the identical grid, we were able to simultaneously compare the flowfield and far-field noise for a baseline jet and a jet with representative chevrons. The chevrons are infinitesimally thin, equilateral, non-penetrating triangles, and were not optimized or designed though any US Navy supported program.

We described low-resolution LES and noise predictions consisting of a modest 5.5 million cells. With the identical grid we conclude that the chevrons, at over-expanded power settings, have the potential to reduce the noise at least 1–2 dB, and possibly more in the downstream peak direction. We then greatly increased the resolution for a chevron calculation to 128 million cells and compared with the low-resolution calculation. The high-resolution grid increases noise-levels in the upstream direction due to enhanced resolution of BBSAN. For realistic, full-scale engines where these fine-scale turbulent structures can dominate the noise at upstream directions, the high-resolution could be required. However, in the downstream direction, the high-resolution only increases the noise above a Strouhal number of 2–3, which is well beyond the peak-level frequency dominating the acoustic field.

Using 1,958 processors on both the NAVY Cray XT5 system (Einstein) as well as the ARSC Cray XE6 system (Chugach), we were able to run the high-resolution calculation for a total of 0.00462 seconds. We have also demonstrated exceptional scalability of CRAFT CFD® up to this number of processors, and expect even better speedup as load-balancing improves with more processors. More importantly, we have shown the ability to predict the impact of noise-reduction devices on hot supersonic jets of interest to the US Navy.

Acknowledgements

This work was supported by NAVAIR Propulsion & Power SBIR awards. Computing time was provided by the DoD Supercomputing Resource Center and the Arctic Region Supercomputing Center. We would like to thank Dr. John Spyropoulos for his guidance with the project and Ms. Kathy Hollyer for her assistance in allocating computing time.

References

Arunajatesan, S. and Sinha, N., "Hybrid RANS/LES Modeling for Cavity Aeroacoustics Predictions." *Journal of Aeroacoustics*, 2(1), pp. 65–93, 2003.

Brentner, K. and Farassat, F., "An Analytical Comparison of the Acoustic Analogy and Kirchhoff formulation for Moving Surfaces." *AIAA Journal*, 36(8), pp. 1379–1386, 1998.

Calhoon,W.J., Kannepalli, C., Arunajatesan, S., and Dash, S., “Analysis of Scalar Fluctuations at High Convective Mach Numbers.” *AIAA-2002-1087*, 2002.

Erwin, J. and Sinha, N., “Near- and Far-field investigations of Supersonic Jet Noise Predictions using a coupled LES and FW-H equation method.” *ASME Turbo Expo GT2011-45210*, 2011.

Erwin, J. and Sinha, N., “Large eddy simulation of supersonic impinging jets.” *AIAA-2012-0679*, AIAA Aerospace Sciences Meeting, 2012.

Farassat, F., “Derivation of Formulations 1 and 1A of Farassat.” *Tech. Rep. NASA/TM-2007-214853*, Langley Research Center, Hampton, VA, 2007.

Kannepalli, C., Arunajatesan, S., and Dash, S.,”RANS/LES Methodology for Supersonic Transverse Jet Interactions with Approach Flow.” *AIAA-2002-1139*, 2002.

Nichols, J., Ham, F., Lele, S., and Bridges, J., “Aeroacoustics of a supersonic rectangular jet: Experiments and LES predictions.” *AIAA 2012-0678*.

Sinha, N., Kenzakowski, D., Ungewitter, R., Dash, S., and Seiner, J., “Computational and Experimental Investigations of Jet Noise Reduction Concepts for Low Bypass Ratio Military Gas Turbine Engines.” *ASME Turbo Expo 2008, GT2008-50091*.

Sinha, N., Arunajatesan, S., Kannepalli, C., Erwin, J., and Dash, S., “Unified RANS/LES Engineering Framework for Predicting Jet Structure and Noise for Realistic Aircraft Exhausts.” *ONR Advanced Propulsion Conference*, 2009.

Sinha, N., Erwin, J., Kannepalli, C., and Arunajatesan, S., “LES Predictions of Noise Emissions from a Low-Bypass Ratio Military Gas Turbine Engine.” *ASME Turbo Expo 2010, GT2010-22191*.

Numerical Investigation of Three-Dimensional Separation in External Flows

A. Gross, R. Jacobi, and H.F. Fasel
Department of Aerospace and Mechanical Engineering, University of Arizona, Tucson, AZ
{agross, rjacobi, faselh}@email.arizona.edu

Abstract

Flow separation for Navy applications is highly-undesirable. Separation not only deteriorates hydrodynamic performance, but also leads to the generation of energetic coherent structures that can create unsteady loads and increase acoustic signature. Because of the typically-low aspect-ratio of Navy applications separation is almost always three-dimensional. While the understanding of the time-dependent flow physics and underlying hydrodynamic instability mechanisms for two-dimensional separation is relatively well-developed, more research is needed for understanding three-dimensional separation because of its greater complexity. We are employing numerical simulations for investigating three-dimensional separation bubbles both on a flat-plate and on a hemisphere-cylinder body. By inducing separation on a flat-plate, flow curvature effects are excluded. The hemisphere-cylinder geometry was chosen because a large separation bubble develops on the fore-body at angle-of-attack. An added advantage arises from the fact that water tunnel measurements are available for both geometries. Our direct numerical simulations for the flat-plate bubble show that the displacement body aspect-ratio has a profound effect on the separation topology. Also, it was found that tunnel free-stream turbulence had to be accounted for to recover the bubble characteristics observed in the experiment. Direct numerical simulations for a hemisphere-cylinder model at Re=5,000 are in good agreement with the experiment and show a pronounced fore-body separation and distinct leeward vortices. Results obtained with two hybrid turbulence models for Re=50,000 are noticeably different indicating the need for more reliable and accurate turbulence models. The mean-flow topology for both models is; however, identical and the same as for Re=5,000.

1. Introduction

Separation for Navy-relevant geometries (submarines, torpedoes, fins, low-aspect-ratio lifting or control surfaces) is always three-dimensional (3D) and associated with considerable unsteadiness. Coherent structures that originate in the separated shear-layer as a consequence of hydrodynamic instabilities influence separation and reattachment. In particular, coherent structures influence the size and shape of the separated region, and thus the lift and drag characteristics. Coherent structures can also add to the acoustic signature. An improved understanding of the flow physics governing 3D separation in general and the dynamics of such structures in particular is desirable, as this may lead to novel devices or strategies that may help prevent or control separation, and the associated coherent structures.

We are employing Direct Numerical Simulations (DNS) and water tunnel experiments for investigating 3D separation bubbles on a flat-plate generated by the proximity of a 3D displacement body [Kremheller and Fasel, 2010; Jacobi, et al., 2011]. The separated region on the flat-plate is induced by the close proximity of an axisymmetric displacement body. Because separation is generated on a flat-plate, flow curvature effects are deliberately excluded. Insight into the hydrodynamic instability mechanisms can be obtained when instabilities are excited directly through the introduction of controlled disturbances.

In parallel, we are investigating 3D separation on axisymmetric bodies [Gross, et al., 2012]. We chose a hemisphere fore-body, because earlier research indicated considerable flow separation for this geometry [Bippes 1986; Wang and Hsieh, 1992]. At low-Reynolds number conditions and for large angles-of-attack, a separation bubble forms on the fore-body. In addition, the cross-flow causes the boundary-layers to separate, roll-up, and form a pair of stream-wise vortices on the leeward-side of the body (leeward-vortices). Three types of separation lines are common for hemisphere-cylinders: the bow separation line, and the primary and secondary separation lines. The latter two are associated with the leeward vortices. At small angles-of-attack, foci terminate the bow separation line, and horn vortices are present.

The considerable computational expense of fully-resolved DNS can be avoided when all or parts of the turbulence kinetic energy spectrum are modeled. For example, in Reynolds-averaged Navier-Stokes (RANS), all unsteady motion is modeled; while in unsteady RANS (URANS), all but the dominant large-scale structures are modeled. In Large-Eddy Simulations (LES), the filter cutoff is located in the inertial sub-range, and only the statistically more random turbulence is modeled. Hybrid models blend between RANS and LES according to the local grid-resolution and flow properties. For example, near walls they revert to pure RANS models. This property is advantageous, as modern RANS models are well-suited for the computation of attached boundary-layer flows. Away from walls, a typical LES or DNS is recovered.

In this paper, we first discuss the computational methods. We then show results for the flat-plate model geometry and the hemisphere-cylinder model. Finally, a short summary and conclusions are provided.

2. Computational Methods

We employ a two-pronged computational approach: i) An incompressible finite-difference code is used for simulating 3D separation on a flat-plate, and ii) A compressible finite-volume code is employed for simulating the flow over an hemisphere-cylinder model at incidence.

The higher-order-accurate incompressible-finite-difference code employed for the flat-plate model geometry simulations was developed in our laboratory [Meitz and Fasel, 2000]. Derivatives in the stream-wise and wall-normal directions are discretized with fourth-order-accurate compact differences. A pseudo-spectral method is employed in the span-wise direction and a fourth-order-accurate Runge-Kutta scheme is employed for time-integration. The code is parallelized over the spectral modes using the Message Passing Interface (MPI) software.

For simulations of the axisymmetric bodies, we are employing a higher-order-accurate compressible finite-volume code that was also developed in our laboratory [Gross and Fasel, 2008]. The Navier-Stokes equations are discretized with a ninth-order-accurate upwind scheme for the convective terms and a fourth-order-accurate discretization for the viscous terms. The turbulence model transport equations are discretized with a second-order-accurate upwind scheme for the convective terms and a second-order-accurate discretization for the viscous terms. The code was parallelized using MPI.

3. Three-Dimensional Separation on a Flat Plate

Three-dimensional flow separation on a flat-plate is generated by placing a 3D axisymmetric displacement body at a distance, h, from the flat-plate (Figure 1). The cone-shaped displacement body has a hemispherical front-end of diameter, d, and boundary-layer suction was employed in the rear-part of the body to prevent flow separation from the displacement body (for more details regarding the experiments see [Kremheller and Fasel, 2010]). The cone opening angle is 20deg. The local boundary-layer displacement thickness at separation can be varied by changing the downstream distance, s, of the body from the flat-plate leading-edge or by varying the free-stream velocity and thus the Reynolds number. The results shown in this paper are for h=0.1m, s=0.25m, and d=0.1m.

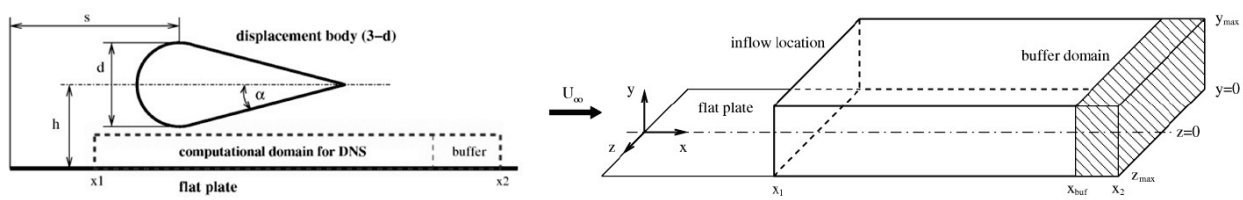


Figure 1. Experimental set-up and computational domain for 3D separation bubble simulations

The computational domain for the DNS is located underneath the displacement body (Figure 1). At the inflow boundary, steady velocity and vorticity profiles are imposed. At the free-stream boundary, y_{max}, the flow is assumed to be irrotational and all vorticity components and their derivatives are set to zero. A Dirichlet boundary condition is applied for the wall-normal velocity, which is obtained from a potential flow solution for the flow over the displacement body. The potential flow solution is mirrored in the wall-normal direction to account for the presence of the flat-plate and in the span-wise direction for simulating the tunnel side-walls. At the wall, y=0, no-slip and no-penetration conditions are imposed. To prevent reflections at the outflow boundary at x_2, the flow is "relaminarized" in a buffer domain starting at x_{buf}. Inside the buffer domain, the disturbance vorticity components are gradually ramped-down to zero. The dimensions of the computational domain are 0.768m×0.048m×0.64m in the stream-wise, wall-normal, and span-wise direction, respectively.

The computational grid has 769×129 grid points in the stream-wise and wall-normal directions, respectively, and 161 modes (257 collocation points) in the span-wise direction.

Figure 2 shows the separation topology for an axisymmetric displacement body at Re_D=11,000 and a displacement body with aspect ratio AR=0.75 at Re_D=5,000. The aspect ratio, AR, is defined as the ratio of the span to body length, and thus the aspect ratio for the axisymmetric displacement body is 0.5. The streamline and skin- friction line patterns indicate that the two bubbles have very similar topologies despite the considerable difference in Reynolds number. The singular point pattern in the reattachment region is the same. Both bubbles feature a secondary separation in the bubble wake. Also, the mid-span pressure gradient was found to increase with increasing displacement body-aspect ratio which led to a larger separation bubble height.

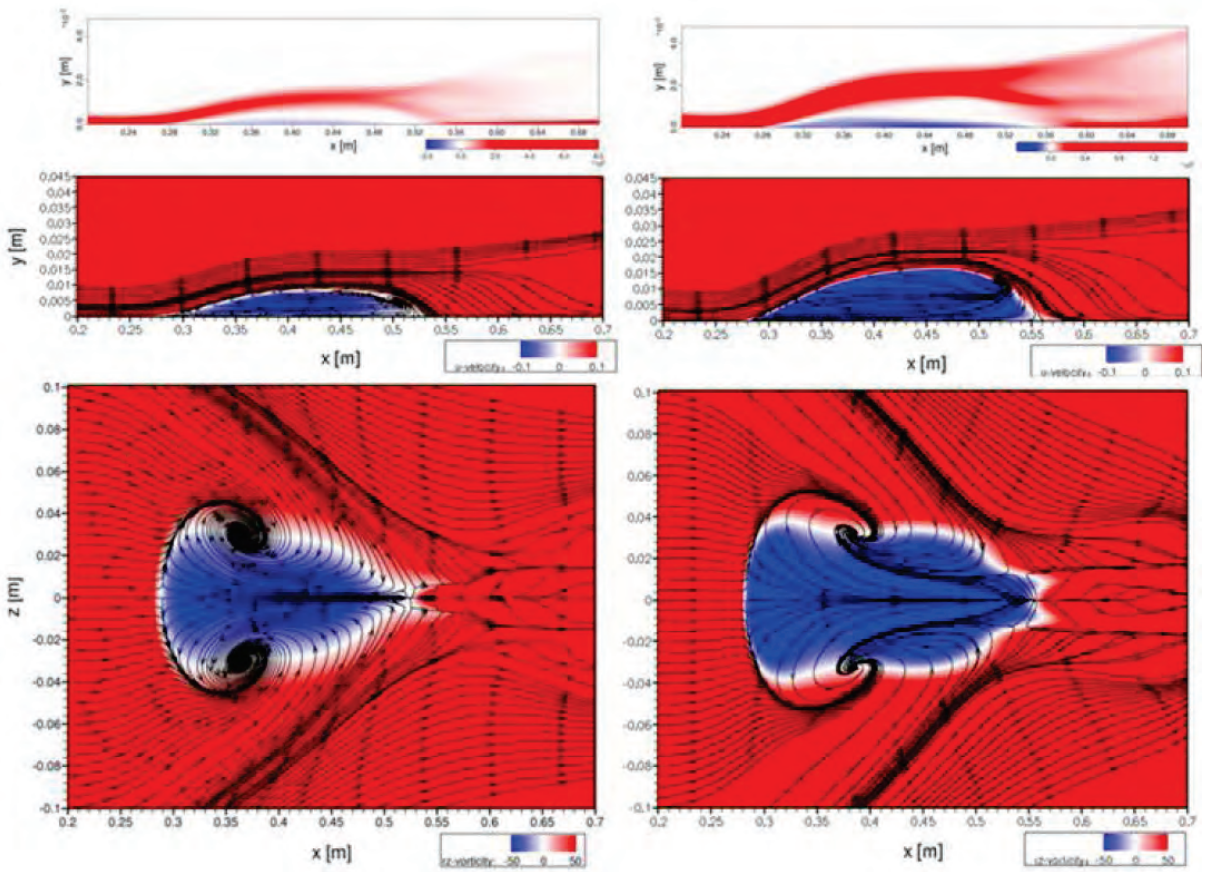


Figure 2. Comparison of time-averaged flow field for axisymmetric displacement body at Re_D=11,000 (left) and non-axisymmetric displacement body with aspect ratio 0.75 at Re_D=5,000 (right). Top: color contours of span-wise vorticity in symmetry plane, center: streamlines in symmetry plane, color contours of stream-wise velocity, bottom: skin-friction lines on the surface, color contours of span-wise vorticity (blue indicates reverse flow).

Figure 3 provides visualizations of the 3D flow fields. The 3D streamlines computed from the time-averaged flow data illustrate how the separation stream-surface rolls-up into a pair of horn vortices that connect to the coherent stream-wise structures in the wake identified by iso-surfaces of the Q vortex identification criterion [Hunt et al., 1988], as well as the iso-contours of stream-wise vorticity in the cross-flow plane. The secondary separation seen in the time-averaged skin-friction lines in Figure 2 downstream of the separation bubbles appears to be caused by these stream-wise structures. Note that these longitudinal coherent structures in aggregate exhibit the same sense of rotation as the leeward vortices seen for hemisphere-cylinder models at incidence. The mean-flow topology is corroborated by our water-tunnel experiments [Kremheller and Fasel, 2010]. Dye flow visualizations and Particle Image Velocimetry (PIV) measurements for Re=5,000 and AR=0.75 are shown in Figure 4. All present evidence, numerical and experimental, supports the conclusion that the separation topology is very similar to what was conjectured by Bippes and Dallmann, et al. [1986] for the flow over a hemisphere cylinder (Figure 5).

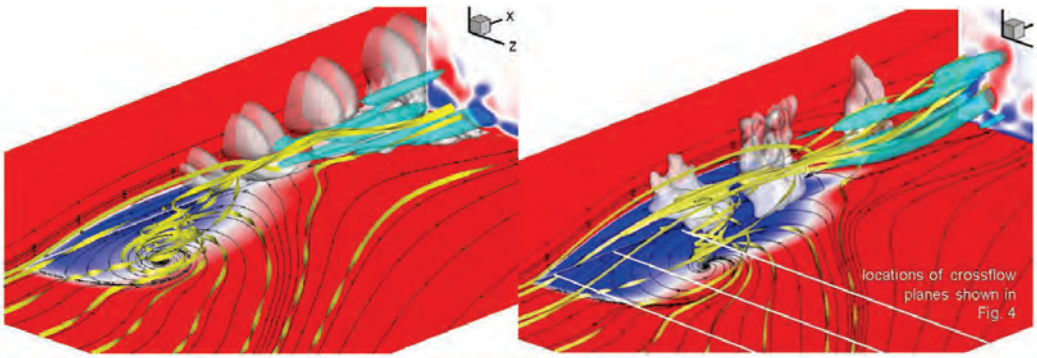


Figure 3. Unsteady separation topology for axisymmetric displacement body at Re_D=11,000 (left) and displacement body with aspect ratio 0.75 and Re_D=5,000 (right). The time-averaged flow field is visualized in the symmetry plane and on the wall by streamlines and skin-friction lines, respectively (black), and color contours of the stream-wise velocity (blue indicates reverse-flow). Also shown are 3D streamlines (yellow), iso-surfaces of the Q vortex identification criterion (cyan), and iso-contours of the stream-wise vorticity in the rear (blue=counter-clockwise, red=clockwise rotation). Grey iso-surfaces illustrate the two most energetic POD modes.

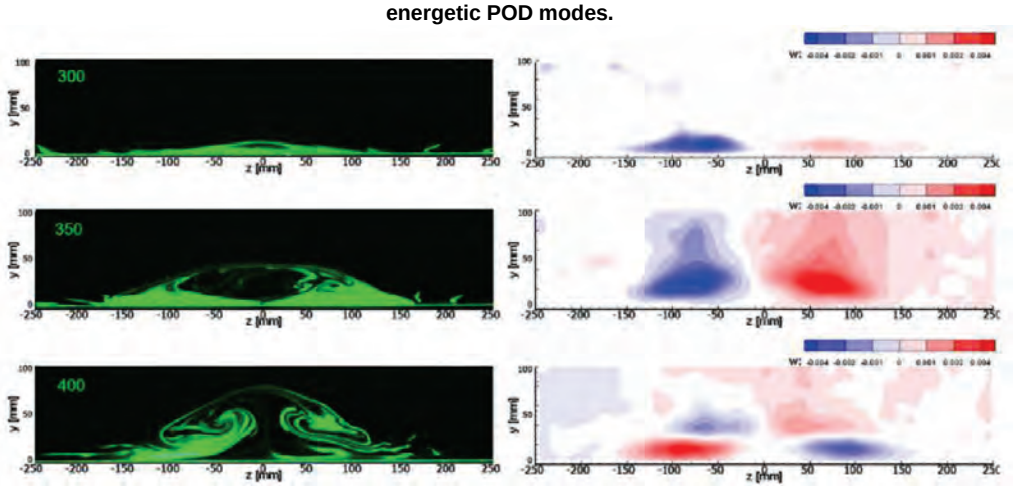


Figure 4. Cross-flow planes inside separation bubble for a displacement body with aspect ratio 0.75 and Re_D=5,000. Water-tunnel fluorescent dye visualizations (left) and color contours of span-wise velocity obtained from Particle Image Velocimetry (PIV) measurements at the downstream locations indicated in Figure 3.

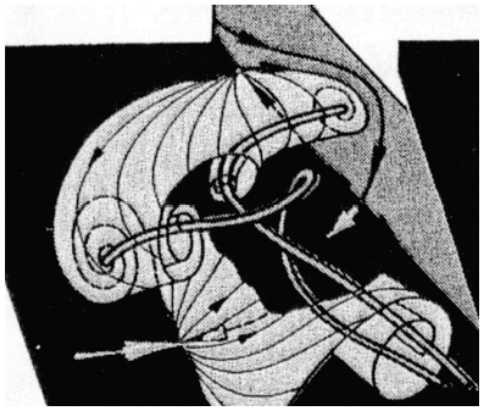


Figure 5. Stream surface topology conjectured by Bippes and Dallmann, et al. [1986] for flow over a hemisphere-cylinder

A Proper Orthogonal Decomposition (POD) [Sirovich, 1987] was carried out for identifying the dominant unsteady coherent flow structures. Figure 3 shows iso-surfaces of the wall-normal velocity component for the two most energetic POD modes. The modes are shifted by a quarter wave-length in the downstream direction which indicates that they represent downstream travelling coherent structures. The modes are nestled in between the strong stream-wise vortices in the wake of the bubble.

Figure 6a provides a comparison between water-tunnel measurements and DNS results for an axisymmetric displacement body at Re_D=10,000. The bubble was shedding in the experiment, but steady in the simulation. When randomized low-amplitude forcing ($v_{max}=5\times10^{-4}\ U_\infty$) was employed to mimic the effect of free-stream turbulence the bubble began to shed in the simulation, and a much better agreement with the experimental mean-flow data was obtained. This result demonstrates how important the exact specification of free-stream turbulence is for numerical simulations where hydrodynamic instabilities have a considerable impact on the separation and transition behavior. Finally, iso-contours of the span-wise vorticity component in the mid-span plane are shown in Figure 6b. With randomized forcing strong time-periodic shedding sets in. The shedding completely alters the bubble mean-flow topology.

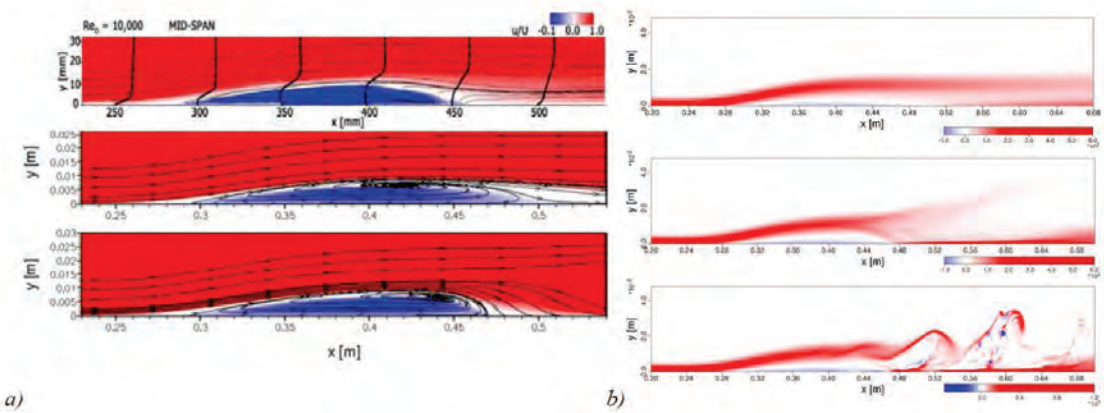


Figure 6. Axisymmetric displacement body at Re_D=10,000. a) Comparison of PIV and DNS data in mid-span plane. Shown are streamlines overlaid with color contours of the stream-wise velocity (blue indicates reverse-flow) for time-averaged PIV data (top), steady DNS without forcing (center), and time-averaged DNS with randomized forcing (bottom). b) Color contours of span-wise vorticity in mid-span plane. DNS without (top) and with randomized forcing (center: time-average, bottom: instantaneous flow field).

4. Separation on Hemisphere-Cylinder Model

Direct numerical simulations and hybrid turbulence model simulations were carried out for a hemisphere-cylinder model. All length-scales were made dimensionless with the body diameter. The body diameter and length were 1 and 4.81, respectively. A Poisson grid-generator was employed to generate an O-grid that was then extruded in the azimuthal direction (Figure 7). The outer (free-stream) boundary of the computational domain was located 50 body diameters from the center of the body. Finally, H-grids were inserted at the front and rear to remove the grid singularities associated with the axisymmetric grid. Table 1 provides the block grid resolutions. A characteristics-based non-reflecting boundary condition was applied at the free-stream boundary and symmetry was enforced at the mid-span plane. Simulations were carried out for Reynolds numbers based on body diameter of Re=5,000 and 50,000. The angle-of-attack was 30deg. The first Reynolds number, Re=5,000, is small enough to allow for DNS. The one-equation Renormalization Group (RG) model by de Langhe, et al. [2008] and the filter-based RANS model by Johansen, et al. [2004] were employed for hybrid simulations at Re=50,000 to lower the computational expense. The filter-based RANS model was based on the 1998 Wilcox k-ω model [Wilcox, 1998].

Figure 7. Computational grid for hybrid simulations of hemisphere-cylinder at Re=50,000

Table 1. Block grid resolutions (number of cells in downstream, wall-normal, and azimuthal direction) and total number of cells

Block1	386×200×64
Block2	32×200×16
Block3	32×200×16
total	5,145,600

Instantaneous flow visualizations for Re=5,000 are shown in Figure 8. Iso-surfaces of the Q vortex identification criterion by Hunt [1988] indicate areas where rotation dominates strain. The flow can be characterized by a laminar separation bubble on the fore-body, two pronounced leeward vortices, and considerable wake unsteadiness. The leeward vortices appear to counteract separation by transporting high-momentum fluid towards the wall. The same flow topology was also identified in water tunnel experiments at the Hydrodynamics Laboratory at the University of Arizona. Dye flow visualizations for Re=5,000 and α=30deg are shown in Figure 9a. The dye is seen to spiral around the cores of the leeward vortices. Dye is also slowly recirculated in the fore-body separation bubble. This is in agreement with the simulation (Figures 9b,c). Figure 10 provides a more quantitative comparison of the simulation data with water tunnel PIV measurements for the symmetry plane. The bubble is slightly shallower in the experiment, probably as a result of tunnel free-stream turbulence, which is not accounted for in the simulation. The good agreement with the water tunnel data for Re=5,000 led to the decision to carry out hybrid turbulence model simulations for a larger Reynolds number, Re=50,000. This Reynolds number is still orders-of-magnitude lower than typical Reynolds numbers for torpedoes or submarines. Nevertheless, by gradually increasing the Reynolds number, we can validate different hybrid turbulence modeling approaches and also investigate how the flow physics change with Reynolds number.

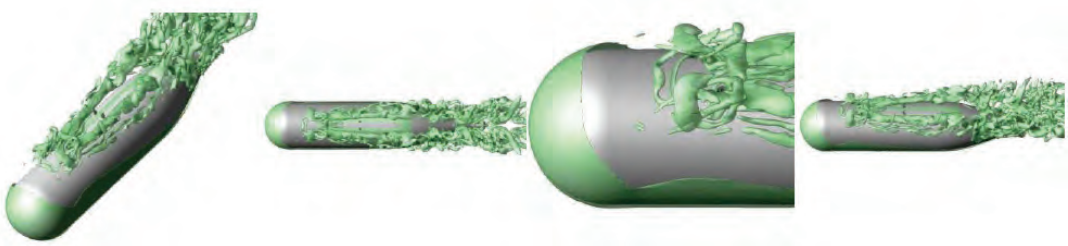


Figure 8. Hemisphere-cylinder at Re=5,000 and α=30deg; Iso-surfaces of Q=10

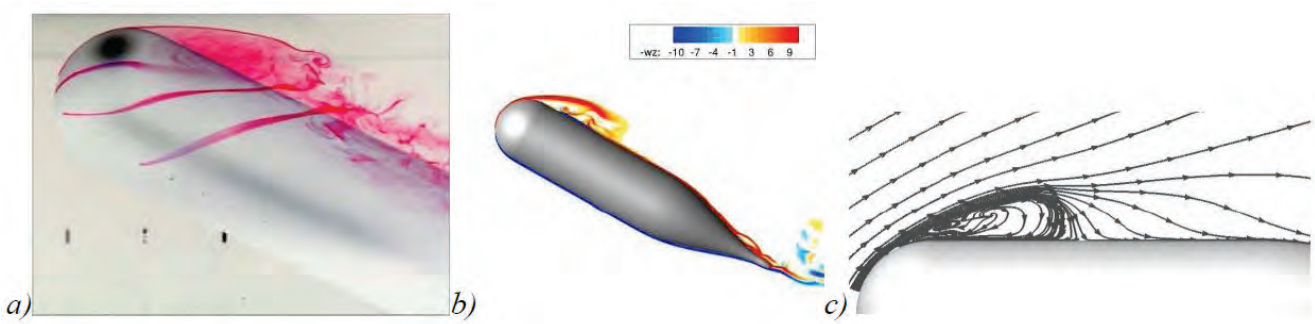


Figure 9. Hemisphere-cylinder at Re=5,000 and α=30deg. a) Water tunnel dye flow visualization (inverted color scale) [Gross, et al., 2012]. b) Iso-contours of span-wise vorticity, ω_z, and c) streamlines in symmetry plane obtained from simulation.

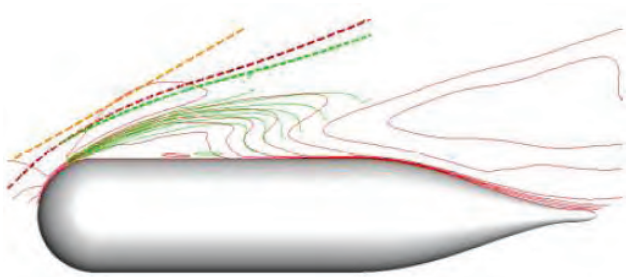


Figure 10. Hemisphere-cylinder at Re=5,000 and α=30deg. Iso-contours of u-velocity (−0.2<u<1.2) in symmetry plane. Red lines: Simulations; Green lines: Measurements. Red dashed lines: Streamlines extracted from simulations; Green dashed lines: Streamlines extracted from PIV measurements; Orange dashed lines: 30deg angle reference line.

Iso-contours of the unresolved-eddy viscosity, μ_{Tu}, and the instantaneous span-wise vorticity component, ω_z, in the symmetry plane are presented in Figure 11. With filter-based RANS model a fore-body separation bubble is obtained which is shedding violently. The unresolved-eddy viscosity in the unsteady wake of the model is about 5-times larger than the molecular viscosity. With the RG model, all of the unsteady motion is modeled and a steady flow field is obtained. This is not desirable since the available grid resolution would clearly allow for some resolved turbulence, and since the RG model was not designed as a pure RANS model.

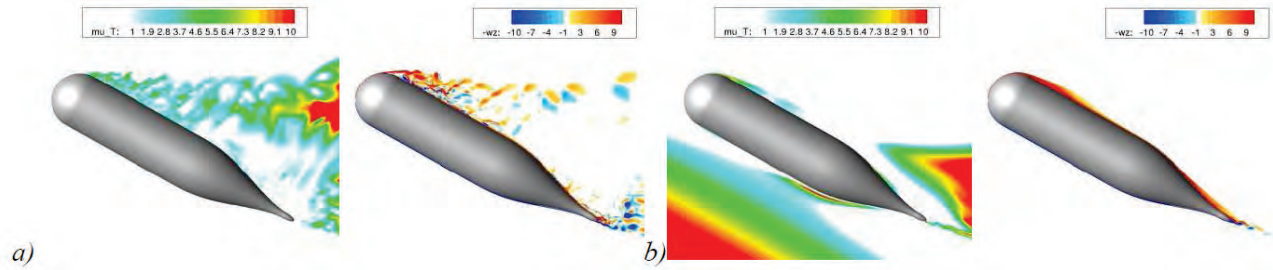


Figure 11. Hemisphere-cylinder at Re=50,000 and α=30deg. Iso-contours of unresolved-eddy viscosity, μ_T, and spanwise vorticity, ω_z, in symmetry plane obtained from a) filter-based RANS simulation and b) RG simulation.

Iso-contours of the turbulence model contribution function, f, and the unresolved turbulence kinetic energy, k_u, for the filter-based RANS model are provided in Figure 12. The turbulence model contribution in the wake is about 20%. Larger model contributions are expected for simulations at higher-Reynolds numbers. The unresolved turbulence kinetic energy is mostly concentrated near the downwind end of the fore-body separation bubble.

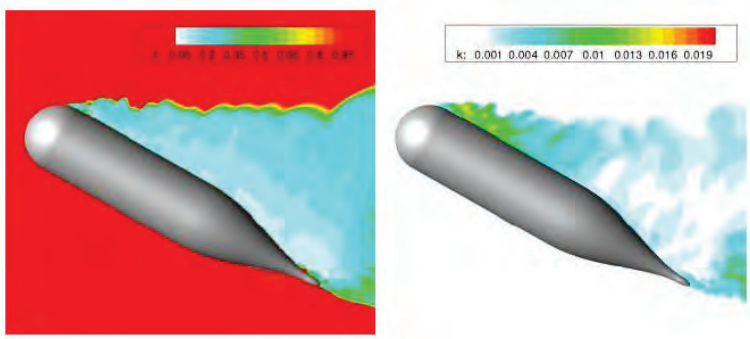


Figure 12. Hemisphere-cylinder at Re=50,000 and α=30deg. Iso-contours of model contribution, f, and unresolved turbulence kinetic energy, k_u, in symmetry plane obtained from filter-based RANS simulation.

Instantaneous visualizations of the vortex identification criterion are presented in Figure 13. With the filter-based RANS model a surprising amount of small-scale unsteadiness is resolved, even in comparison with the DNS for Re=5,000 (Figure 8). The leeward vortices remain present; however, they appear embedded in small-scale motion and must therefore be understood as steady coherent structures in a turbulent flow. With the RG model all flow unsteadiness is modeled, except for near the very stern of the model.

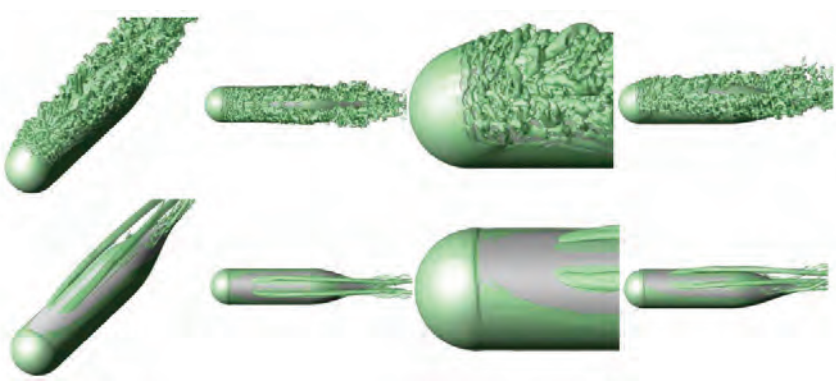


Figure 13. Hemisphere-cylinder at Re=50,000 and α=30deg. Instantaneous iso-surfaces of Q=10 for filter-based RANS (top) and RG model (bottom).

In Figure 14, time-averages for the 3 different cases are compared. In all instances, the leeward vortices remain the dominant feature that characterizes the flow field. The fore-body separation is shortened as the Reynolds number is increased. The separation topology of the fore-body separation becomes more complicated for Re=50,000 compared to Re=5,000.

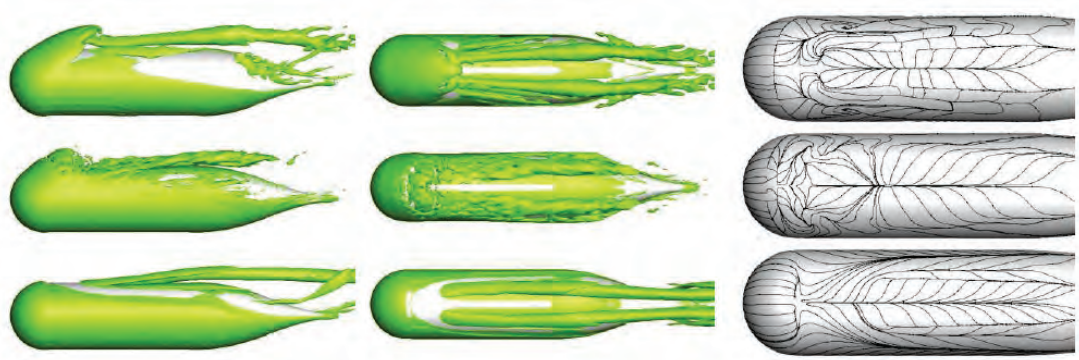


Figure 14. Hemisphere-cylinder at α=30deg. Iso-surfaces of Q=5 (left and center) and skin-friction lines (right) computed from time-averaged data for Re=5,000 (top) and Re=50,000 (center and bottom). Filter-based RANS model (center) and RG model (bottom).

5. Conclusion

Direct numerical simulations were carried out for investigating steady and unsteady 3D separation bubbles on a flat-plate generated by the close proximity of displacement bodies with different aspect ratios, AR. The separation topology for a displacement body with aspect ratio 0.75 at Re=5,000 and for an axisymmetric displacement body (AR=0.5) at Re=11,000 were found to be very similar. A comparison of simulation data with water tunnel measurements for an axisymmetric displacement body at Re=10,000 revealed that tunnel free-stream turbulence had to be accounted for in order to obtain a good match of the time-averaged bubble length and height. Without the randomized forcing that was employed for modeling the tunnel turbulence, the 3D separation bubble was steady which is in stark contrast to the experiment. This interesting finding illustrates that the realistic modeling of free-stream turbulence is a necessary prerequisite for realistic simulations of laminar 3D separation bubbles.

In addition, direct numerical simulations and hybrid turbulence model simulations were carried out for a hemisphere-cylinder model at 30deg angle-of-attack and for Reynolds numbers based on diameter of Re=5,000 and 50,000. At Re=5,000 a laminar separation bubble forms on the fore-body and pronounced leeward vortices develop that appear to suppress shedding. The agreement of the simulation data with water tunnel dye flow visualizations and measurements for Re=5,000 was good and encouraged hybrid simulations for Re=50,000. Two different hybrid turbulence models were employed. For the filter-based RANS model, the model contribution was relatively low and a large amount of unsteady flow structures was resolved. With one-equation renormalization group model all of the unsteadiness was modeled. This is a clearly undesirable feature, as the grid resolution is sufficient for resolving parts of the energy spectrum. Also, the hybrid model was not designed as a pure RANS model. Nevertheless, both simulations indicate that the flow topology at Re=50,000 remains similar to the flow topology at Re=5,000.

6. Significance to DoD

Flow separation deteriorates hydrodynamic performance, increases structural loads, and may result in an undesirable acoustic signature. For Navy applications, separation is mostly three-dimensional and unsteady. Our simulations of generic 3D separation bubbles advance the understanding of 3D separation, which paves the way for improved vehicle designs and the development of separation control devices that will increase hydrodynamic loading and reduce drag and acoustic signature.

Acknowledgments

This research was supported by the US Office of Naval Research (ONR) under grant number N00014-10-1-0404 with Dr. Ronald Joslin serving as Program Manager. Compute time provided by Challenge Project ONRDC07692C4Y is highly appreciated. This work was supported in part by a grant of HPC resources from the Arctic Region Supercomputing Center.

References

Bippes, H., "Experimental Investigation of Topological Structures in Three-dimensional Separated Flow." Boundary-Layer Separation IUTAM Symposium, London, 1986.

De Langhe, C., Bigda, J., Lodefier, K., and Dick, E., "One-equation RG hybrid RANS/LES computation of a turbulent impinging jet." *Journal of Turbulence*, Vol. 9, No. 16, pp. 1–19, 2008.

Gross, A. and Fasel, H., "High-Order Accurate Numerical Method for Complex Flows." *AIAA J.*, Vol. 46, No. 1, pp. 204–214, 2008.

Gross, A., Jagadeesh, C., and Fasel, H.F., "Numerical Investigation of Three-Dimensional Separation on Axisymmetric Bodies at Angle of Attack." *AIAA-2012-97*, 2012.

Hunt, J.C.R., Wray, A.A., and Moin, P., "Eddies, stream, and convergence zones in turbulent flows." *Report CTR-S88*, Center for Turbulence Research, Stanford, CA, 1988.

Jacobi, R., Gross, A., and Fasel, H.F., "Numerical Investigation of Three-Dimensional External Flow Separation." *HPCMP Users Group Conference 2011*, Portland, OR, 2011.

Johansen, S.T., Wu, J., and Shyy, W., "Filter-based unsteady RANS computations." *International Journal of Heat and Fluid Flow*, Vol. 25, No. 1, pp. 10–21, 2004.

Kremheller, A. and Fasel, H.F., "Water Tunnel Experiments on Three Dimensional Separation Bubbles on a Flat Plate." *AIAA-2010-4738*, 40th AIAA Fluid Dynamics Conference and Exhibit, Chicago, IL, 2010.

Meitz, H.L. and Fasel, H., "A Compact-Difference Scheme for the Navier-Stokes Equations in Vorticity-Velocity Formulation." *J. Comp. Phys.*, Vol. 57, No. 1, pp. 371–403, 2000.

Sirovich, L., "Turbulence and the Dynamics of Coherent Structures." *Quarterly of Applied Mathematics*, Vol. XLV, No. 3, pp. 561–590, 1987.

Wang, K.C. and Hsieh, T., "Separation Patterns and Flow Structures about a Hemisphere-Cylinder at High Incidences." *AIAA-92-2712*, 1992.

Wilcox, D.C., *Turbulence Modeling for CFD*, 2nd edition, DCW Industries, Inc., La Canada, CA, 1998.

Structural and Magnetic Properties of $EuTiO_3$ from First Principles

Yurong Yang and L. Bellaiche
Physics Department and the Institute for Nanoscience and Engineering, University of Arkansas, Fayetteville, AR
{yyrwater, laurent}@uark.edu

Abstract

The structural stability and magnetic exchange interaction of $EuTiO_3$ bulks are studied from first-principles. Results of different numerical schemes, namely the so-called PBE+U, PBEsol+U, and hybrid-functional Heyd-Scuseria-Ernerhof (HSE), are discussed. We find that the PBEsol+U approach can give structural parameters and magnetic exchange interactions that are in remarkable agreement with experiment, and at a much lower computational cost than HSE. Using the PBEsol+U approach, we further predict three stable structures (with space groups I4/mcm, Imma, and C2/c) that have a much lower total energy than the cubic paraelectric state.

1. Introduction

Multiferroic materials, in which electric and magnetic orderings coexist and are coupled to each other, have attracted great attention, partly because of various potential applications[1]. The search for novel multiferroics, therefore, constitutes an important research field direction nowadays. Interestingly, while $EuTiO_3$ (ETO) in its bulk form is not multiferroic (it exhibits a G-type antiferromagnetic (AFM) ordering at temperature of 5.3 K[2], but adopts a quantum paraelectric (PE) behavior[3,4], down to the lowest temperatures, implying that it does not possess an electric polarization), experimental studies on ETO *films* show the occurrence of ferroelectricity (and, therefore, make ETO film multiferroic) for an epitaxial tensile strain of about 1.1%. ETO also possesses a strong spin-lattice coupling and large magnetoelectric (ME) effects, as evidenced by the fact that a 7% change in the dielectric constant upon the application of a magnetic field of 2 T has been demonstrated[5–9].

Previous first-principles studies predicted that the magnetic ground-state of ETO bulk changes from an AFM state to a ferromagnetic (FM) state when increasing the cell volume. These computations also predicted that multiferroicity appears when placing ETO under epitaxial compressive and tensile strains[10,11]. Furthermore, using hybrid function, K Akamatsu[12] studied the antiferromagnetic super-exchange in cubic ETO. Recently, oxygen octahedral rotation in ETO was also studied by first-principles[13]. However, these latter calculations used the so-called GGA+U method that usually overestimates the lattice constants. It is therefore difficult, using this GGA+U method, to give reasonable prediction for the spin-lattice coupling in ETO and for physical properties of this system when under epitaxial strain. It is thus desired to find a numerical method that not only (i) gives excellent agreement with measurements for ETO; but (ii) that is also not too computationally demanding. In this paper, we report that the so-called PBEsol+U provides a solution to these two requirements (i) and (ii).

2. Methods

All the calculations are performed in the framework of density-functional theory as implemented in the Vienna *ab initio* simulation package[14,15] with a plane-wave of 550 eV. The so-called PBE+U[16], PBEsol+U[17] and hybrid-functional HSE[18] calculations are carried out using the same set of projector-augmented wave (PAW) potential to describe the electron-ion interaction[19,20]. The Dudarev's approach[21] with values of the on-site Coulomband exchange parameters for the Eu ion, U=5.7 eV and J=1 eV for PBE+U *versus* U ranging between 5.1 and 6.1eV, and J=1 eV for PBEsol+U, are used here. For the $\sqrt{2}\times\sqrt{2}\times\sqrt{2}$ super-cell (10 atoms) of cubic ETO bulk with space group $Pm\overline{3}m$, a 6×6×6Γ -centered

k-point mesh is used. For $\sqrt{2}\times\sqrt{2}\times 2$ super-cell (20 atoms) for ETO bulk with space groups I4/mcm, Imma, and C2c, a 6×6×4 Γ-centered k-point mesh is used. Both the atomic positions and super-cell lattice vectors are allowed to relax until the force on each atom is less than 0.5meV/Å. Only G-type antiferromagnetic (AFM) and ferromagnetic orderings are considered here.

Table 1. Volume (Å^3), equilibrium lattice constant (a_{eq}) and magnetic exchange constant (J_1), from PBE, PBEsol, HSE, and experiment for cubic ETO with AFM

	GGA-PBE (U=5.7)	GGA=PBEsol (U=5.57eV)	HSE	EXP[4,5,22]
V_0 (Å^3)	61.16	58.86	59.32	59.32
a_{eq} (Å)	3.94	3.89	3.90	3.90
J_1 (K)	0.015	−0.014	−0.014	−0.014[5], −0.02[4]

3. Results and Discussion

We start our calculation from the PBE+U approach with U=5.7 eV and J=1 eV, which was recently used in the literature[10,11,13]. The results for the cubic, paraelectric ETO are given in Table 1. The equilibrium lattice constant from PBE+U is 3.94 Å, that overestimates by about 1% the experimental value of 3.90Å. The magnetic exchange parameter is found to be 0.015 K for this theoretical lattice constant, that differs considerably in magnitude, and even sign from the experimental results of −0.014 K. (Note that this magnetic exchange parameter is computed by mapping the energy difference between different magnetic configurations of this system to the Heisenberg Hamiltonian as $H=-2\sum_{i>j}^{N} J_{ij}S_i\cdot S_j$.) This explains why, to get reasonable magnetic exchange parameter, the experimental lattice constant is typically used in the PBE+U method. However, doing so renders difficult an accurate prediction of the low-energy states and phases under epitaxial strain.

The Heyd-Scuseria-Ernzerhof (HSE) hybrid function, where the exchange mixing parameter and other parameters are used to split the Coulomb interaction kernel into short- and long-range pieces while retaining the mixing only on the short-range component, is known to provide satisfactory performance in insulators[23] and semiconductor/oxides interfaces[24]. From Table 1, one can see that the lattice constant and magnetic exchange parameter predicted by HSE calculations perfectly agree with the experimental results in cubic ETO. However, HSE calculations are computationally very costly, which makes the prediction of phases associated with larger super-cells very difficult.

That is the reason why we decided to turn to PBEsol+U calculation. Interestingly, using the same values of U and J, as in the aforementioned PBE+U calculation, gives a lattice constant of 3.89 Å (see Figure 1), that is much closer to the experimental one of 3.90 Å than the one given by the PBE+U calculations (about 3.94 Å). We are now wondering if PBEsol+U can also provide a reasonable magnetic exchange interaction. For that, we calculated the energy difference between the ferromagnetic and antiferromagnetic states of cubic ETO as a function of U, as shown in Figure 2, for the equilibrium lattice constant corresponding to these U values. To get the magnetic exchange parameter J_1 predicted by HSE calculations and determined from experiments, the U value in PBEsol+U should be precisely equal to 5.57 eV. From Figure 3, one can also see that the equilibrium lattice constant linearly increases when increasing U in these PBEsol+U calculations. Interestingly, when U=5.57 eV, the lattice constant of the AFM state is 3.8912 Å. The results of PBEsol+U with U=5.57 eV are summarized in Table 1. They agree rather well with those of the HSE calculations and experiment.

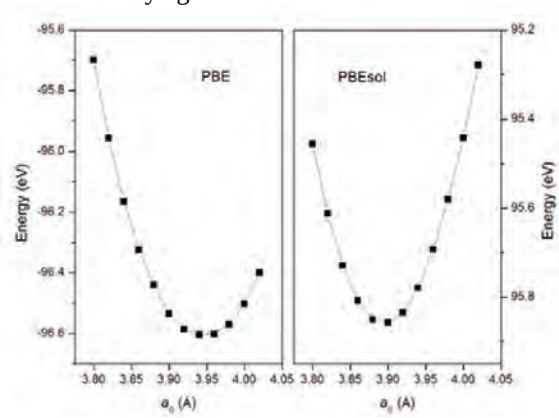


Figure 1. Energies as a function of lattice constant (a_0) for PBE+U and PBEsol+U with U=5.7 for cubic ETO with AFM

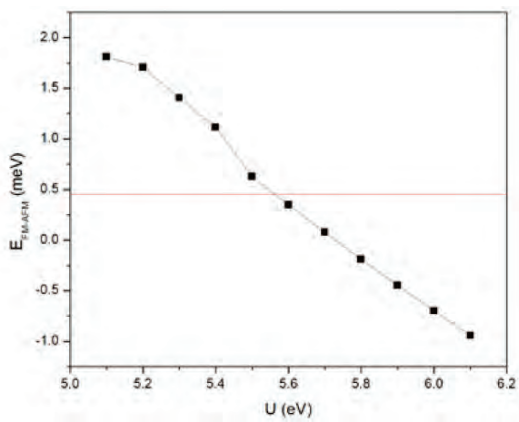


Figure 2. Energy differences between ferromagnetic and antiferromagnetic state as a function of U for PBEsol+U calculations. The horizontal line shows the energy difference predicted by HSE calculations for cubic ETO.

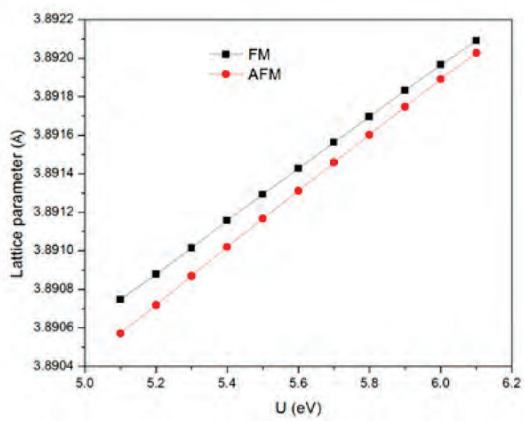


Figure 3. Equilibrium lattice constant as a function of U for the ferromagnetic and antiferromagnetic state in PBEsol+U calculations for ETO cubic

Let us thus now use PBEsol+U calculations, with U=5.57 eV, to predict properties of phases other than the cubic state, namely those having the following space groups: I4/mcm, Imma, and C2/c. Interestingly, these three phases are found to be more stable than the cubic paraelectric phase! Their energy and equilibrium lattice parameters are shown in Table 2. The I4/mcm, Imma, and C2/c states have similar total energies, which are all lower than that of $Pm\bar{3}m$ by about 36 meV per 5 atom. The I4/mcm phase has the lowest energy, and is characterized by antiphase antiferrodistortive patterns about the z direction (see Figure 4b). Imma has an energy higher than I4/mcm by only 1meV and has antiphase antiferrodistortive patterns about x and y directions (see Figure 4c). Finally, C2/c has an energy higher than I4/mcm by 2meV and has antiphase antiferrodistortive patterns about x, y, and z directions (see Figure 4d). The tilting angles are also shown in Table 2.

Table 2. Total energy, equilibrium lattice constants and antiferrodistortive pattern of oxygen octahedral tilting (expressed in Glazer's notation[25]) in the $Pm\bar{3}m$, I4/mcm, Imma, and C2/c phases.

	$Pm\bar{3}m$ (cubic)	I4/mcm	Imma	C2/c
E (meV/f.u.)	36	0	1	2
Lattice constraints (Å)	(3.89, 3.89, 3.89)	(5.47, 5.47, 7.84)	(5.53, 5.49, 7.74)	(5.51, 5.48, 7.78)
Tilting	$a^0a^0a^0$(0, 0, 0)	$a^0a^0c^-$ (0, 0, 8.2°)	$a^-a^-c^0$ (6.2°, 6.2°, 0)	$a^-b^-c^-$ (4.3, 4.2, 5.3°)

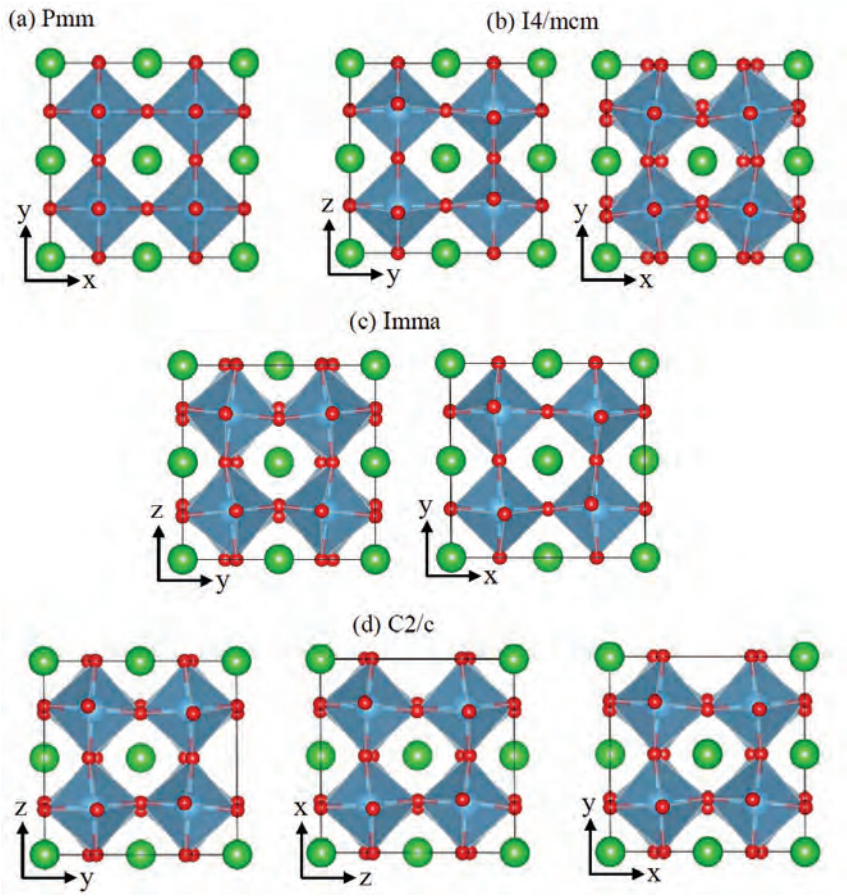


Figure 4. Schematic representation of ETO for the (a) $Pm\bar{3}m$, (b) I4/mcm, (c) Imma, and (d) C2/c states

4. Conclusion

$EuTiO_3$ bulk has been studied by first-principles using the PBE+U, PBEsol+U, and HSE hybrid function. HSE calculations give the best results when comparing to measurements. Interestingly, PBEsol+U calculations also provide very similar results than HSE calculation and experimental results, when tuning the U value. Finally, using the PBEsol+U method with the "right" U value, we further predict that three structures of ETO, namely those of I4/mcm, Imma, and C2/c symmetries, are more stable than the paraelectric, cubic $Pm\bar{3}m$ state. Our results indicate that multiferroic properties of ETO films under epitaxial strain should be studied by first-principles within the PBEsol+U method, unlike what has been done so far.

Acknowledgments

We acknowledge the Office of Naval Research (ONR) Grants N00014-11-1-0384 and N00014-08-1-0915, and the computational support provided by the ONR Grant N00014-07-1-0825 (DURIP), and a Challenge Grant from the Department of Defense (HPCMP RPN: ONRDC: 2756-3-262). This time was made available at US Army Engineer Research and Development Center (ERDC).

References

1. Wang, Y., J. Hu, Y. Lin, and C.-W. Nan, *NPG Asia Mater*, 2, pp. 61–68, 2010.

2. McCarthy, G., W. White, and R. Roy, *Journal of Inorganic and Nuclear Chemistry*, 31(2), pp. 329–339, 1969.

3. Chien, C.-L., S. DeBenedetti, and F.D.S. Barros, *Physical Review B*, 10 (9), pp. 3913-3922, 1974.

4. McGuire, T.R., M.W. Shafer, R.J. Joenk, H.A. Alperin, and S.J. Pickart, *Journal of Applied Physics*, 37 (3), pp. 981–982, 1966.

5. Katsufuji, T. and H. Takagi, *Physical Review B*, 64 (5), 054415, 2001.

6. Gong, S.J. and Q. Jiang, *Physica Status Solidi (B)*, 241 (13), pp. 3033–3040, 2004.

7. Jiang, Q. and H. Wu, *Journal of Applied Physics*, 93 (4), pp. 2121–2125, 2003.

8. Wu, H. and W.Z. Shen, *Solid State Communications*, 133 (8), pp. 487–491, 2005.

9. Wu, H., Q. Jiang, and W.Z. Shen, *Physics Letters A*, 330 (5), pp. 358–364, 2004.

10. Fennie, C.J. and K.M. Rabe, *Physical Review Letters*, 97 (26), 267602, 2006.

11. Lee, J.H., L. Fang, E. Vlahos, X. Ke, Y.W. Jung, L.F. Kourkoutis, J.–W. Kim, P.J. Ryan, T. Heeg, M. Roeckerath, V. Goian, M. Bernhagen, R. Uecker, P.C. Hammel, K.M. Rabe, S. Kamba, J. Schubert, J.W. Freeland, D.A. Muller, C.J. Fennie, P. Schiffer, V. Gopalan, E. Johnston–Halperin, and D.G. Schlom, *Nature*, 466 (7309), pp. 954–958, 2010.

12. Akamatsu, H., Y. Kumagai, F. Oba, K. Fujita, H. Murakami, K. Tanaka, and I. Tanaka, *Physical Review B*, 83 (21), 214421, 2011.

13. Rushchanskii, K.Z., N.A. Spaldin, and M. Ležaić, *Physical Review B*, 85 (10), 104109, 2012.

14. Kresse, G. and J. Hafner, *Physical Review B*, 47 (1), pp. 558–561, 1993.

15. Kresse, G. and J. Furthmüller, *Physical Review B*, 54 (16), pp. 11169–11186, 1996.

16. Loschen, C., J. Carrasco, K. M. Neyman, and F. Illas, *Physical Review B*, 75 (3), 035115, 2007.

17. Perdew, J.P., A. Ruzsinszky, G.I. Csonka, O.A. Vydrov, G.E. Scuseria, L.A. Constantin, X. Zhou, and K. Burke, *Physical Review Letters*, 100 (13), 136406, 2008.

18. Heyd, J., G.E. Scuseria, and M. Ernzerhof, *The Journal of Chemical Physics*, 124 (21), pp. 219906–219901, 2006.

19. Blöchl, P.E., *Physical Review B*, 50 (24), pp. 17953–17979, 1994.

20. Kresse, G. and D. Joubert, *Physical Review B*, 59 (3), pp. 1758–1775, 1999.

21. Dudarev, S.L., G.A. Botton, S.Y. Savrasov, C.J. Humphreys, and A.P. Sutton, *Physical Review B*, 57 (3), pp. 1505–1509, 1998.

22. McGuire, T.R., M.W. Shafer, R.J. Joenk, H.A. Alperin, and S.J. Pickart, *Journal of Applied Physics*, 37 (3), pp. 981–982, 1966.

23. Gou, G., I. Grinberg, A.M. Rappe, and J.M. Rondinelli, *Physical Review B*, 84 (14), 144101, 2011.

24. Shaltaf, R., G.M. Rignanese, X. Gonze, F. Giustino, and A. Pasquarello, *Physical Review Letters*, 100 (18), 186401, 2008.

25. Glazer, A., *Acta Crystallographica Section* B, 28 (11), pp. 3384–3392, 1972.

6.1 Basic Research

Other DoD Agencies

Multi-Scale Simulation of Deflagration-to-Detonation Transition in Turbulent Dense Two-Phase Reactive Mixtures

Kalyana C. Gottiparthi and Suresh Menon
School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA
{kalyan.gottiparthi, suresh.menon}@gatech.edu

Abstract

The process of transition of a deflagration-to-detonation (DDT) in stoichiometric H_2–Air mixture in channels with and without obstacles has been simulated and studied. The channel width has been varied from 2, 4, 8, and 16 cm, with a constant blockage ratio 0.5 to simulate the quasi-detonation and the detonation regimes observed in experiments and earlier computational results. The effect of variation of volume-fraction (αp_0) of the inert particles on DDT in channels is analyzed by introducing steel particles into the accelerating flame, in a channel of width 16 cm and blockage ratio 0.5. The energy and the momentum absorbed by the particles results in the deceleration of the leading shock-wave and the flame, and the interaction of the particles with the flame enhance the instabilities, thereby increasing the flame-speed. These two conflicting effects, in combination, result in three detonation-initiation scenarios. In the first scenario, transition-to-detonation occurs through formation of a hot-spot, due to the leading shock-obstacle interaction and in the second scenario, the transition occurs due to leading shock-flame coupling at the wall of the channel. In the third scenario, the detonation-initiation occurs due to the leading shock-reflection from the obstacle and subsequent interaction of the reflected shock and the flame. Thus, the distance and time for deflagration to detonation is observed to vary with $\alpha_{p0,}$ based on the initiation scenario.

1. Introduction

Decomposition or neutralization of the chemical/biological (CB) agent via its entertainment into flames/hot spots, ignition/combustion by blast-waves, dispersion of particles/droplets, and venting and plume formation from the damaged structure are all highly-transient phenomena with a wide-range of characteristic time and length scales. Geometry of the target structure can also introduce unsteady flow complexities, such as re-circulating flow behind obstacles, blast-wave reflections and refractions that can affect mixing and combustion processes. Most of the Agent Defeat (AD) scenarios involve these complex processes, the knowledge of which is vital for effective agent neutralization. One of the fundamental problems which occur in multiple AD scenarios and very challenging to resolve is Deflagration-to-Detonation Transition (DDT). Confined flames in reactive mixtures are known to generate detonations and effect the survival of CB agents in the flow. Thus process of DDT in channels with gas-particle mixtures, with primary focus on the effect of presence of inert particles on DDT, is studied to address the processes which explain the flame acceleration and transition to detonation.

Generally, there are two ways to initiate detonations in explosive systems, either by direct initiation or by deflagration-to-detonation transition (Lee, 2008). When an ignition source of sufficient strength, which can couple the reaction and the shock wave and cause detonation immediately, is used, direct initiation occurs. However, if a weak ignition source is used, a deflagration is formed and under suitable initial and boundary conditions the deflagration accelerates to form a detonation. This process of transition of a deflagration-to-a-detonation involves several complicated multi-scale phenomena. Even though several investigations, both experimental and numerical, have been performed to understand these phenomena (Ciccarelli and Dorofeev, 2008), most of them involve pure gaseous reactive mixtures. Past studies involving DDT in channels with and without obstacles provide insights into the processes involved in DDT and enable to estimate the duct-length and time required for DDT (Bradley, et al., 2008; Silvestrini, et al., 2008). However, the knowledge of the vital mechanisms which control DDT in gas-particle mixtures is still lacking.

Very few investigations have been conducted on DDT in multi-phase mixtures (Wolanski, 1990; Kapila, et al., 2001) in comparison to pure gaseous phase DDT studies. The classical work by Baer and Nunziato (1986) provided a two-phase theory for DDT in reactive granular materials. Using the theory they were able to predict the distances for the transition to detonation. Investigations of DDT in an end multi-phase slug using experiments and numerical analysis were conducted by Zhang, et al. (1998). Their results suggest that the presence of a dense cloud of particles ahead of an accelerating flame would lead to the formation of reflected shocks which facilitate rapid transition to detonation. Also, the reflected detonations, from the end-wall, are observed to achieve peak pressures, which are twice than that achieved in pure gaseous phase cases. Experimental investigation of effect of sand particles of radius of the order of 0.1–1.0 mm on DDT in stoichiometric $H_2{-}Air$ mixture in channels without obstacles showed that the increase in particle volume fraction results in decrease in the time and distance to DDT (Borissov, 1991). The particles in the experiment, which are much larger than the micron-size particles considered in the present study, enhance the flame acceleration by generating small-scale turbulence and bow shocks. DDT in dust mixtures have also been reported earlier (Wolanski, 1990). These studies suggest that the transition from deflagration-to-detonation in grain dust mixtures occurs based on the reactivity and the concentration of the particles. However, in all these investigations, the processes involved in DDT in gas-particle mixtures, where particle-size is in orders-of-microns, are not explored and to the authors' knowledge there are no results available in literature on the influence of particles, either inert or reactive, on DDT in channels, with or without obstacles.

Due to the unique nature of the threat posed by CB agents, there is a critical need to develop a first-principle simulation methodology that can be used *a priori* to evaluate the threat, and also design and optimize possible threat countermeasures. As a part of a multi-year research, sponsored by Defense Threat Reduction Agency (DTRA), novel approaches to simulate destruction of CB agents using fuel mixtures that can mix and burn with aerosolized materials behind strong blast waves created by targeted detonation, are being developed. These simulation strategies aim to provide accurate and reliable predictions of impact of AD scenarios when implemented on actual targets, as well as, to develop new strategies to handle unforeseen scenarios. DDT modeling covers the entire gamut of the problem areas of direct interest, and is also a challenge in its own right. The ability to capture DDT numerically in multi-phase mixtures is, in itself, extremely challenging since shock capturing, combustion, multi-phase and multiple-scale interactions all have to be simulated accurately. In this paper, the results obtained during the first seven months of the challenge project are presented. The main focus of the initial phase of the work has been to conduct high-resolution simulations of flame acceleration and detonation initiation in channels with and without obstacles. Both pure gaseous and two-phase mixtures have been considered.

This paper is organized as follows. The second section of this paper summarizes the numerical method used and the performance of the code (LESLIE3D). The pure gas phase simulations, which form the base for the two-phase simulations, and the effect of volume-loading on flame acceleration and DDT, are discussed in the third section. Finally, sections four and five provide the conclusions and the significance of the current work to the Department of Defense (DoD), respectively.

2. Formulation and Performance of the Code

2.1 Numerical Method

In the present study, the unsteady, compressible, multi-phase Navier-Stokes equations for reacting gas flows are used to solve for the gas phase, and Lagrangian tracking approach is used to solve for the position, the mass, the velocity and the temperature of the solid particles (Balakrishnan, et al., 2010). Further, in all the simulations presented here, every particle in the flow is deterministically tracked, without using any approximation or modeling. A single-step Arrheius kinetics based on the parameters described in Table 1 is employed (Gamezo, et al., 2008). The momentum and the energy transfer between the two phases is evaluated based on the empirical expressions for the drag coefficient (C_D) and the Nusselt number (Nu) (Gottiparthi and Menon, 2012).

The governing equations for the gas phase are solved using approximate Riemann solver with Monotone Upstream-centered Schemes for Conservative Laws (MUSCL) reconstruction and monotonized-central limiter. The Riemann solver used is a hybrid solver (Einfeldt, 1988) which employs the HLLC flux solver everywhere except in the shock-capturing region, where in the direction transverse to the shock the HLLE Riemann solver is used. The scheme is second-order-accurate in both time and space, although its spatial accuracy decreases in the region where the discontinuity is present. Detailed validation of this approach for shock-turbulence interaction (Génin and Menon, 2010) and detonations (Gottiparthi and Menon, 2012) has already been reported. The governing equations for solid phase are integrated in time using 4th-order Runge-Kutta scheme. In order to account for dense loading effects, Euler gas-phase fluxes in each cell are modified using the Discrete Equations Method (DEM) (Abgrall and Saurel, 2003). DEM allows for the treatment of volume fraction of gas phase as a piece-wise continuous variable and the inter-cell interfaces are evaluated using distinctive Riemann

problems. Details of the DEM approach for Eulerian-Lagrangian formulation, used in the current study, are presented elsewhere (Balakrishnan, et al., 2010).

2.2 Numerical Setup

The computational domain used is shown in Figure 1 and is discretized using a structured-grid. The grid is uniformly fine in the regions (of length L_f given in Table 1), where the flame and shock exits and is coarsened in the regions where the fuel is completely burnt or completely unburnt with no flow property gradients. The total channel length, L, and the number of obstacles considered in each case are shown in Table 1. The minimum grid resolution Δ_{min}=1/144 cm is used which corresponds to 5 points per flame thickness (x_1=0.035 cm) and 3 points per half-reaction zone thickness ($L_{1/2}$=0.02 cm) of the detonation in stoichiometric H_2–*Air* mixture. The grid-resolution is chosen based on resolution studies performed earlier (Gamezo, et al., 2007; Gottiparthi and Menon, 2011).

The particles are considered to have properties of steel (i.e., C_P and ρ_p). Tables 2 and 3 list the initial solid-phase conditions, the various cases, the number of particles, and the initial volume-fraction of the solid-phase for each case. In this paper, the radius of the particles (r_p) is 16 µm. Cases with r_p=2, 4, and 8 μm have been already reported (Gottiparthi and Menon, 2011) albeit in dilute flows i.e., $\alpha_p \leq 0.01$ everywhere in the flow. The flame is ignited by placing a hot region of burnt mixture, where the temperature is $2.0T_b$, where T_b=$7.289T_0$ is the post-flame temperature. This mode of ignition is 'soft' mode, i.e., the energy provided for the flame ignition is of the order of the energy release due to the flame. The boundary conditions used are also shown in Figure 1. The surfaces of the walls and obstacles are adiabatic no-slip reflecting surfaces. At the outlet of the channel, the properties are extrapolated as the flow becomes supersonic. When the simulation is performed in half of the channel, symmetry condition at the half-channel width is employed. These boundary conditions are similar to the conditions used in previous studies (Gamezo, et al., 2007; 2008).

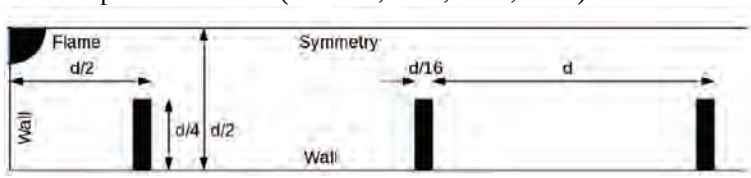


Figure 1. Computational domain used for simulation of DDT in a channel with obstacles. The obstacles and the initial flame location are shown along with the boundary conditions used. The simulation domain is same as that used by Gamezo, et al. (2007). Here, the blockage ratio (the ratio of obstacle-height to the half-channel width) is 0.5 and the initial flame radius is d/8. The length of the channel in each case is long enough to initiate detonation and the obstacles are equally-spaced.

Table 1. Grid and domain lengths for channels of different widths used in the current study

Case	*d*/2 (cm)	*L* (cm)	Number of obstacles	L_f (cm)	Grid size for $L_f \times d/2$
G1	1.0	60.0	30	23.0	3312×144
G2	2.0	72.0	18	34.0	4896×288
G3	2.0	72.0	18	32.0	9216×576
G4	4.0	96.0	12	44.0	6336×576
G5	8.0	176.0	11	56.5	8136×1152

Table 2. Input parameters for deflagration and detonation in stoichiometric H_2–*Air* mixture and inert particles used in the current study (Gamezo, et al., 2007; 2008)

Gas Phase		
$P_{g,0}$	1.0 atm	Initial gas pressure
$T_{g,0}$	293 K	Initial gas temperature
R	385.93 J/(mol K)	Gas constant
γ	1.17	Ratio of specific heats
MW	21 g/mol	Molecular weight
A	$6.85 \times 10^{12} cm^3/(gs)$	Pre-exponential factor
Ea	$46.37RT_0$	Activation energy
Q	$43.28RT_0$	Chemical energy release
μ_0	$2.9 \times 10^{-5} g/(scmK^{0.7})$	Transport constant
Le	1.0	Lewis number
Pr	0.72	Prandtl number
Solid Phase		
ρ_p	7874 kg/m^3	Particle density
$T_{p,0}$	293 K	Initial particle temperature
$u_{p,i}$	0.0 m/s	Initial particle velocity

Table 3. Number of particles tracked in each case (i.e., for a particular α_{p0}) in the current study. The radius of particles r_p=16.0 μm and are distributed over length=80.0 cm and over the entire half-width (8 cm), and are located at 80.0 cm from the closed end of the channel. The gas phase setup for all two-phase cases is same as Case G5.

Case	α_{p0}	Number of particles
With obstacles		
P1	1.0×10^{-5}	2517
P2	5.0×10^{-5}	12560
P3	1.0×10^{-4}	25107
P4	5.0×10^{-4}	125515
Without obstacles		
P5	1.0×10^{-5}	2565
P6	5.0×10^{-5}	12800
P7	1.0×10^{-4}	25587
P8	5.0×10^{-4}	127915

2.3 Code Scaling

The performance of the code for the representative case with d=16.0 cm on Raptor (Cray XE6) is shown in Figure 2. The speed-up and the efficiency are defined as

$$\text{Speed-Up} = \frac{\text{Time for execution using } N_{ref} \text{ processors}}{\text{Time for execution using } N \text{ processors}} \tag{1}$$

$$\text{Efficiency} = \frac{\text{Time for execution using } N_{ref} \text{ processors} \times N_{ref}}{\text{Time for execution using } N \text{ processors} \times N} \tag{1}$$

where N_{ref} for in the present study is 128. The scaling of the code is very good and the efficiency remained above 90% over the wide-range of processors used, i.e., from 128 to 4,000. Most of the cases presented in this paper have been run using 1,000 processors where the efficiency is nearly 95%. The scaling analysis is also performed with particle tracking included. For this, 72,000 particles are tracked in a shock tube (6×0.13×0.13 m) resolved using 3,000×65×65 grid-points. The cases with particles also show good scaling similar to the gas-phase cases.

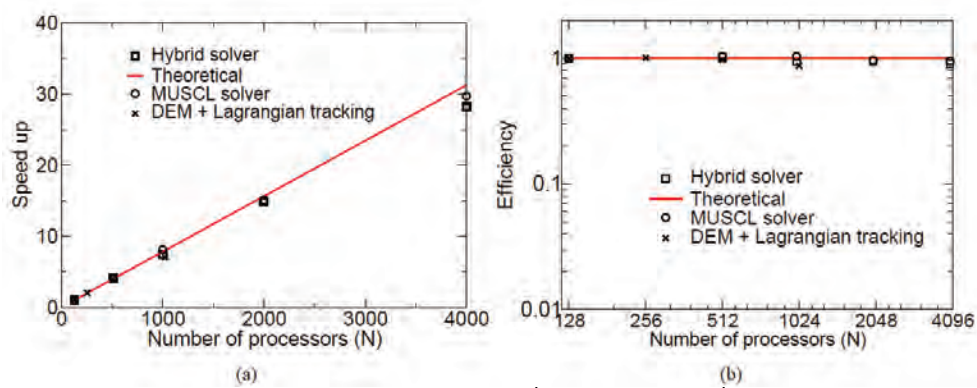


Figure 2. (a) Speed-up and (b) efficiency with MUSCL and hybrid (2nd-order centered/3rd-order MUSCL scheme) solvers for Case with d=16 cm (9,342×2,304) on Raptor (Cray XE6). N_{ref} is 128.

3. Results and Discussion

The process of flame acceleration and subsequent initiation of detonation in pure gaseous mixture is simulated, initially, in order to investigate the effect of inert particles on DDT in channels with obstacles. Half-channel widths of 1, 2, 4, and 8 cm with blockage ratio 0.5 are considered where flame-acceleration regimes are examined. Validation of the results obtained from pure gaseous simulations is performed by comparing with the results available in the literature. The accelerated flame obtained from the gaseous simulation is used to study DDT in channels with and without obstacles in single and two-phase reactive mixtures.

3.1 DDT in Pure Gaseous Mixtures

The vital steps involved in formation of a detonation starting from an ignited flame kernel in a channel are: i) formation of an accelerating flame and flow, ii) shock-wave formation ahead of the flame, and iii) detonation initiation and propagation (Gamezo, et al., 2008). These steps, as observed in Case G5, are shown in Figure 3.

Initially, the ignited flame kernel introduced at the closed-end of the channel propagates slowly. As the flame kernel propagates into the reactants, the flame surface area increases resulting in greater energy release and an accelerating flame is formed. In addition, as the accelerating flame propagates, the flame and flow instabilities, such as Kelvin-Helmholtz (KH), Rayleigh-Taylor (RT) and Richtmyer-Meshkov (RM) instabilities (Gamezo, et al., 2008; Gottiparthi and Menon, 2011), result in further increase in flame surface, as shown in Figure 3(a). The flow ahead of the accelerating flame is compressed and a leading-shock-wave is formed. The leading-shock of sufficient strength can reflect from an obstacle and form a strong reflected-shock. This reflected-shock travels downstream and couples with the main-flame front, initiating a detonation (see Figure 3(b)). The detonation generated between obstacles 7 and 8 spreads into the reactant mixture and overtakes the flame and the leading-shock-wave.

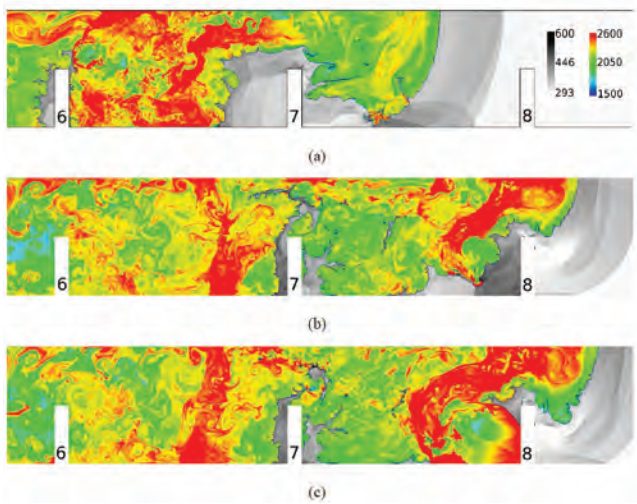


Figure 3. Flame acceleration, and initiation and propagation of detonation in channel with obstacles. The half-channel width is 8 cm and the blockage ratio is 0.5 (Case G5). Notice that in (b) detonation is initiated due to reflected shock-flame interaction. The temperature range of the unburnt mixture is shown in gray-scale and the burnt mixture is shown in color spectrum.

3.1.1 Effect of Channel Width

In order to simulate the main regimes of DDT in channels with obstacles, channels of different widths are considered. For the channel with $d/2$=1 cm. Case G1, the ignited flame accelerated and produced detonations starting from obstacle 16. But these detonations did not spread past the leading-shock and failed after single- diffraction from the obstacle. In channel with $d/2$=2 cm, Case G2, the detonation is produced initially at obstacle 10. But this detonation fails to overtake the leading-shock. This failure of an initiated detonation due to diffraction at the obstacle in the channel with $d/2$=2 cm is shown in Figure 4. Similarly failed detonation appears at obstacle 11. However, at obstacle 12, the detonation produced overtakes the leading-shock and propagates surviving the diffraction at obstacle 13. But after diffraction from obstacle 14, the leading shock decouples from the reaction zone and the detonation is quenched. The process of this detonation formation and quenching is again observed from obstacle 16 to 18. This mode of propagation is called quasi-detonation regime and is observed in earlier numerical and experimental investigations. Similar quasi-detonation mode is observed in Case G4 where the channel half-width ($d/2$) is 4 cm.

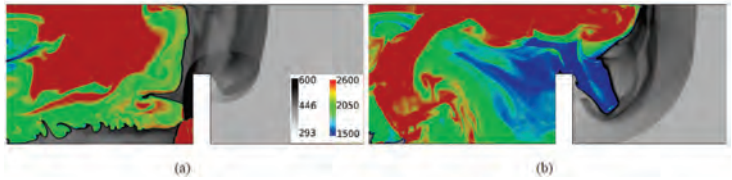


Figure 4. Detonation (a) initiation and (b) failure at the obstacle in channel with *d*/2=2 cm (Case G2). Here, the detonation fails due to expansion after diffraction from obstacle.

The flame position variation with time in channels of different *d* is shown in Figure 5. Here, the flame position is the maximum position along the length of the channel, where mass fraction of the reactant mixture is less than 0.5. The flame position increases rapidly as the flame accelerates and in the cases where the transition to detonation occurs, the flame position varies almost linearly with the time as the detonation propagates with velocity nearly equal to Chapman-Jouget detonation velocity, D_{CJ}, i.e., 2 km/s for the reactive mixture considered in the current study.

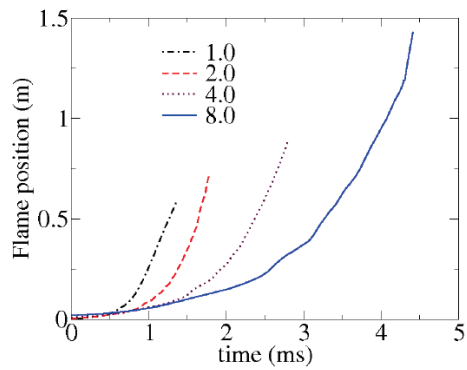


Figure 5. Variation of the position of the flame-tip with time for channel with obstacles for different half-channel widths (Cases G1, G2, G4 and G5)

3.1.2 Effect of Obstacle on Detonation Initiation

In cases where channels had regularly-spaced obstacles, as discussed above, the detonation was initiated at the obstacle due to the interaction of the leading-shock and the obstacle. In case with *d*/2=8.0 cm, the detonation was initiated between the 7th and the 8th obstacles at a distance of 116.6 cm from the closed-end of the channel. Thus, in order to examine the process of detonation initiation in channel without obstacles, the channel was made obstacle-free after the 7th obstacle. This allowed for the flame to accelerate at the same rate as in an obstacle-filled channel, but with the possibility of detonation initiation at an obstacle removed. So the accelerated flame and leading-shock formed after obstacle 7, as shown in Figure 6(a), propagates into the obstacle-free section of the channel. The accelerating flame in the obstacle-free section generated a hot -pot at a distance of 121.7 cm from the closed-end of the channel and a detonation was initiated. Thus, in comparison to a channel with obstacles, the distance to DDT is increased by 5.1 cm in a channel without obstacles. Also, the detonation resulted from a hot-spot in obstacle-free case, whereas a reflected shock initiated the detonation in the case with the obstacle.

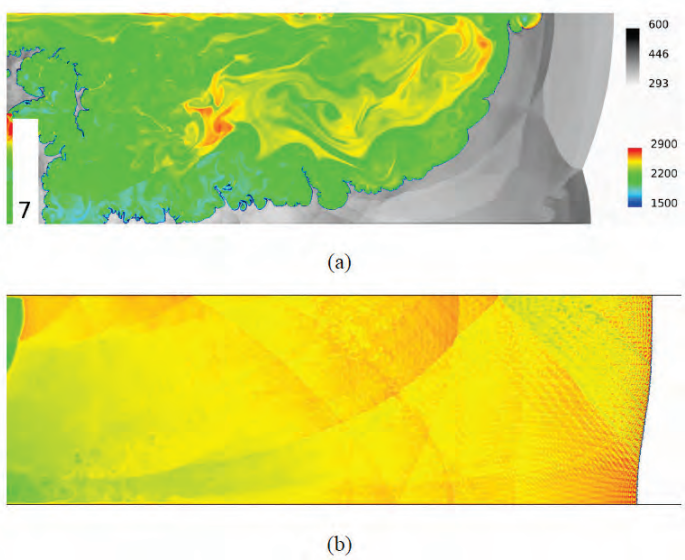


Figure 6. Flame acceleration and detonation initiation. The half-channel width is 8 cm and the blockage ratio is 0.5. The temperature range of the unburnt mixture is shown in gray-scale and the burnt mixture is shown in color spectrum. Here, all the obstacles after obstacle 7 are removed and the detonation initiates and propagates through obstacle-free channel. Note in (b) the characteristic detonation front structure is observed.

3.2 DDT in Gas-Particle Mixtures

Inert particles, which are uniformly distributed over the length and the width of the channel, are introduced into the accelerating flame developed in the channel with obstacles and $d/2$=8 cm, to study the process of DDT in gas-particle mixture. As shown in Figure 7(a), the flame accelerates in the pure gaseous mixtures and encounters gas-particle mixture at a location 80 cm from the closed-end of the channel (see Figure 9(a)). The propagation characteristics of the accelerating flame and the subsequent transition to detonation in the gas-particle mixture is studied by varying α_{p0}.

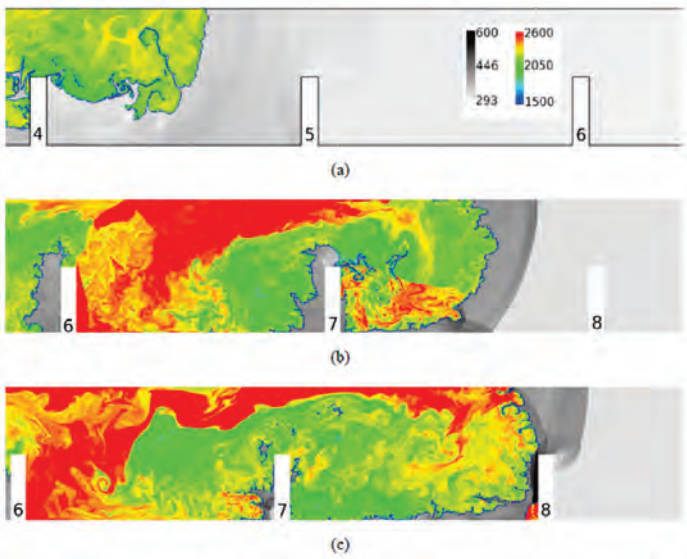


Figure 7. Flame acceleration and detonation initiation in a channel with inert particles. The half-channel width is 8 cm, the blockage ratio is 0.5, $\alpha_{p0}=10^{-4}$ and r_p=16 μm (Case P3). Notice the detonation is initiated at the obstacle in (c). The temperature range of the unburnt mixture is shown in gray-scale and the burnt mixture is shown in color spectrum.

3.2.1 Effect of Initial Volume Fraction of Solid Phase

The effect of particle volume fraction on DDT is studied by varying α_{p0} from 10^{-5} to 5.0×10^{-4} for a fixed-particle radius 16μm. When inert-particles interact with the accelerating flame, due to the instabilities generated by the particles, fine flamelets are generated along the flame-front as shown in Figure 7(b). These flamelets enhance the reaction and increase the flame-speed. Also, the flame spreads evenly, in comparison to pure gaseous case, along the width of the channel due to the spreading of the flamelets. This causes the flame to accelerate further. However, the particles also absorb momentum and energy from the flow, and can lead to deceleration of the flame and the leading shock wave. These two effects are counteracting and, as discussed below, would reduce or increase the distance to DDT based on the volume fraction of the particles in the flow.

In the case of $\alpha_{p0}=10^{-4}$ (Case P3), the energy and momentum absorbed by the particles causes the leading-shock and the flame to decelerate in comparison to the pure gaseous case (Case G5) as shown in Figure 8. The deceleration is marginal and the flame does not reach the leading-shock to initiate the detonation. But, unlike the pure gaseous case discussed earlier (Case G5), where the detonation is initiated because of flame-reflected shock couple, in this case the detonation is initiated at distance 1.19 m by the collision of the leading shock with the obstacle 8 as shown in Figure 7(c). Thus, the distance to DDT is increased by 2.4 cm. Even in Cases P1 and P2, the detonation is initiated at the obstacle as in Case P3. However, the flame in these cases marginally accelerates (see Figure 8) due to flamelet-generation by the particles. In the case with $\alpha_{p0}=5.0\times10^{-4}$, although flamelets are generated, the energy and momentum absorption from the flow leads to net flame deceleration. However, the leading shock also decelerates such that the flame reaches the leading-shock and couples to initiate the detonation. For this case, the detonation is initiated at a distance nearly 116 cm from the closed-end of the channel. The particle dispersion, for $\alpha_{p0}=10^{-4}$, in the channel at different times is shown in Figure 9. The flame and flow interact with the particles and generate particle clusters, especially near obstacles, where the volume fraction of the solid-phase, α_p, is 10–16%. Thus considering the blockage effect of the solid-phase is important when simulating the effect of particles on DDT. Finally, in Cases P5 to P8, where the obstacles after Obstacle 7 are removed, the detonation is not initiated within 176 cm, i.e., the location of the open-end of the channel. Thus, unlike the corresponding gaseous case, channel lengths greater than 176 cm, considered here, are required to observe the transition in these cases.

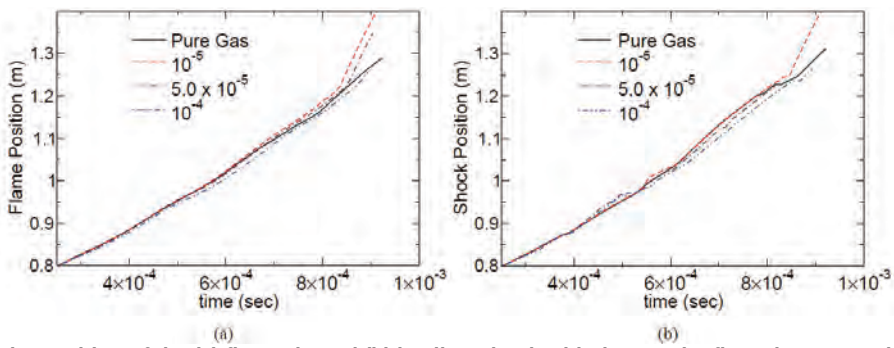


Figure 8. Variation of the position of the (a) flame-tip and (b) leading-shock with time as the flame interacts with the inert particles. α_{p0} in each case is indicated. The half-channel width is 8 cm and the blockage ratio is 0.5. Notice that the shock is slowed for α_{p0}=5.0×10^{-5} and 10^{-4} and flame is accelerated for α_{p0}=5.0×10^{-5} and 10^{-5} in comparison to pure gaseous mixture case.

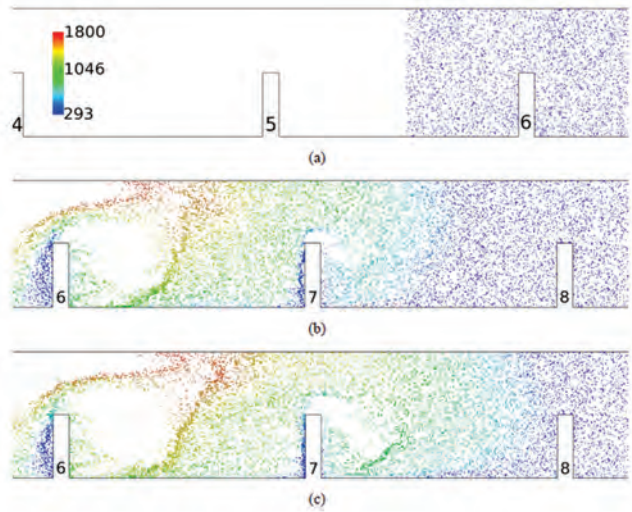


Figure 9. Particle clustering due to flame and flow acceleration in a channel with inert particles. The half-channel width is 8 cm, the blockage ratio is 0.5, α_{p0}=10^{-4} and r_p=16 μm (Case P3). Here (a), (b) and (c) correspond to (a), (b), and (c) in Figure 7, respectively. The particle color indicates the particle temperature.

In summary, detonation initiation in gas-particle mixtures in channels with obstacles can occur in at least three distinct ways. In the first scenario, the detonation is initiated by the leading-shock (LS) collision with an obstacle which generates a hot-spot at the obstacle corner, as shown in Figure 10(a) (in Cases P1, P2, and P3). In this case, accelerating flame (*F*) is not coupled with any shock, and detonation is formed away from the flame. In the second scenario shown in Figure 10(b) (Case P4), the leading-shock couples with the flame to initiate the detonation. This scenario has been observed only in two-phase cases where the particles help to decrease the strength of leading-shock which enables the flame to reach the shock. Finally, the interaction of the reflected leading-shock from an obstacle with the flame initiates the detonation in the third scenario (see Figure 10(c) (Case G5)).

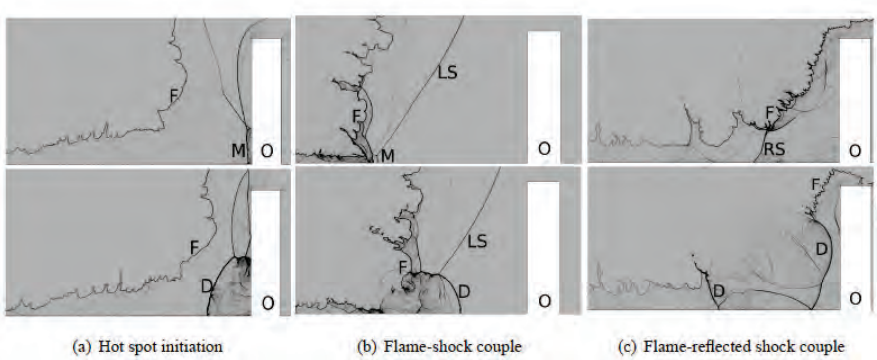


Figure 10. Detonation initiation due to the hot-spot, the flame-leading shock couple and the flame-reflected shock couple. Here, density gradient is plotted. *O*-Obstacle; *F*-Flame; *LS*-Leading Shock; *M*-Mach stem; *RS*-Reflected Shock; *D*-Detonation.

4. Conclusion

The process of deflagration-to-detonation transition in gas-particle mixtures is studied using inert (steel) particles and H_2-Air mixture. The reaction is modeled using single-step Arrhenius kinetics and Lagrangian tracking is used to solve for the particle phase. Further, DEM is used account for the volume occupied by the dispersed phase. Simulations in channels of half-width 1, 2, 4, and 8 cm are performed to simulate the quasi-detonation and detonation regimes of DDT in channels with obstacles, and to compare with the numerical and experimental results from literature. The results from the pure gaseous simulations are in agreement, qualitatively, with the past results (Gamezo, et al., 2007; 2008). The process of DDT in gas-particle mixtures in channels with obstacles is studied by varying initial volume fraction of solid phase, α_{p0}, for fixed-particle radius, r_p=16 cm. The studies indicate that the inert-particles control DDT in gas-particle mixtures by absorbing energy and momentum and generating flamelets. The particles, for $\alpha_{p0}=5.0\times10^{-4}$, facilitated the flame acceleration and leading-shock deceleration and the detonation was initiated at the wall, unlike pure gaseous cases where the detonation is initiated due to the reflected leading-shock and flame interaction. For $\alpha_{p0}=10^{-5}$, 5×10^{-5}, and 10^{-4}, the flame could not couple with the leading-shock wave and detonation initiation occurred at the obstacle corner due to hot-spot formation. Thus, three scenarios i.e., hot-spot formation, flame-leading-shock couple or flame-reflected-shock couple, have been observed to result in detonation initiation. Identification of the change in these initiation scenarios based on α_{p0} which will enable to predict the distance to DDT *a priori* is necessary, and will be addressed in future.

5. Significance to DoD

One of the missions of DTRA is to develop strategies to destroy chemical and biological agents that may be deployed in both combat and urban environments. In order to do so there is a critical need for a predictive tool to analyze various possible scenarios and develop successful deployment of mitigating strategies. One of the fundamental scenarios encountered in multiple-agent defeat strategies is DDT, and the current work focuses to address vital phenomena in flame acceleration and detonation initiation in multiphase mixtures. The computational approach developed and applied here will address some of the goals under a DTRA-funded basic research effort which aims for maximum bio-agent neutralization in post-blast wave environment.

Acknowledgments

This work is supported by the Defense Threat Reduction Agency (Dr. Suhithi Peiris, Program Manager). The computational resources were provided under DoD HPC Challenge Project by DoD Supercomputing Resource Centers at US Air Force Research Laboratory and US Army Engineer Research and Development Center.

References

Abgrall, R., and R. Saurel, "Discrete equations for physical and numerical compressible multiphase mixtures." *J. Comput. Phys.*, 186(2), pp. 361–396, 2003.

Baer, M.R., and J.W. Nunziato, "A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials." *Int. J. of Multiphase Flow*, 12(6), pp. 861–889, 1986.

Balakrishnan, K., D.V. Nance, and S. Menon, "Simulation of impulse effects from explosive charges containing metal particles." *Shock Waves*, 20, pp. 217–239, 2010.

Borissov, A.A., *Dynamic Structure of Detonation in Gaseous and Dispersed Media*, Kluwer Academic Publishers, 1991.

Bradley, D., M. Lawes, and K. Liu, "Turbulent flame speeds in ducts and the deflagration/detonation transition." *Combust. Flame*, 154, pp. 96–108, 2008.

Ciccarelli, G., and S. Dorofeev, "Flame acceleration and transition to detonation in ducts." *Progress in Energy and Combustion Science*, 34, pp. 499–550, 2008.

Einfeldt, B., "On godunov-type methods for gas dynamics." *SIAM J. Numer. Anal.*, 25(2), pp. 294–318, 1988.

Gamezo, V.N., T. Ogawa, and E.S. Oran, "Numerical simulation of flame propagation and DDT in obstructed channels filled with hydrogen-air mixture." *Proc. Combust. Inst.*, 31, pp. 2463–2471, 2007.

Gamezo, V.N., T. Ogawa, and E.S. Oran, "Flame acceleration and DDT in channels with obstacles: Effect of obstacle spacing." *Combust. Flame*, 155, pp. 302–315, 2008.

Génin, F., and S. Menon, “Studies of shock/turbulent shear layer interaction using large-eddy simulation.” *Comput. Fluids*, 39, pp. 800–819, 2010.

Gottiparthi, K.C., and S. Menon, “Study of deflagration-to-detonation transition in gas-particle mixtures.” AIAA Paper 2010-0801, 2011.

Gottiparthi, K.C., and S. Menon, “A study of interaction of clouds of inert particles with detonation in gases.” *Combust. Sci. Technol.*, 184, pp. 406–433, 2012.

Kapila, A.K., R. Menikoff, J.B. Bdzil, S.F. Son, and D.S. Stewart, “Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations.” *Phys. Fluids*, 13(10), pp. 3002–3024, 2001.

Lee, J.H.S., *The Detonation Phenomenon*. Cambridge University Press, New York, 2008.

Silvestrini, M., B. Genova, G. Parisi, and F.J.L. Trujillo, “Flame acceleration and DDT run-up distance for smooth and obstacles filled tubes.” *J. Loss Prev. Process Ind.*, 21, pp. 555–562, 2008.

Wolanski, P., “Deflagration and detonation combustion of dust mixtures.” *Prog. in Astronautics and Aeronautics*, 132, pp. 3–31, 1990.

Zhang, F., P.A. Thibault, and S.B. Murray, “Transition from deflagration to detonation in an end multiphase slug.” *Combust. Flame*, 114, pp. 13–25, 1998.

Strong Coupling Expansion for the Nonequilibrium Properties of Ultracold Atoms on an Optical Lattice

J.K. Freericks and K. Mikelsons
Department of Physics, Georgetown University, Washington, DC
{freericks, karlis}@physics.georgetown.edu

H.R. Krishnamurthy
Centre for Condensed Matter Theory, Department of Physics, Indian Institute of Science, Bangalore, India
hrkrish@physics.iisc.ernet.in

Abstract

We describe a new method, based on a strong coupling expansion, to model ultra-cold atoms on an optical lattice in non-equilibrium. The method is ideally-suited for the systems with strong interactions and high initial temperatures, which is often the case for cold atom experiments. We discuss the computational aspects of this method, including precision, accuracy, scaling and numerical implementation. We present results for fermionic atoms in a strong homogeneous electric field, showing the Bloch oscillations of the current and the evolution of the momentum distribution in time.

1. Introduction

The field of ultra-cold atoms has been growing extensively over the past few years. One of the reasons why this is so is that these systems are clean and tunable and, hence, have the possibility for providing direct testing of theoretical predictions. Richard Feynman proposed using controllable quantum mechanical systems for analog quantum computation, allowing the system to evolve, and then reading out the results of the quantum mechanical system (Feynman, 1982). While simple in principle, carrying out such a project is a daunting task full of many technical challenges. Defense Advanced Research Projects Agency (DARPA) is sponsoring a program on creating just such an optical lattice emulator, employing ultra-cold atoms within an egg-carton-like potential created by retro-reflected laser beams, and simulating the presence of a periodic potential of real materials. Nearly all experiments currently carried out work in non-equilibrium, where the system is driven by some kind of external disturbance and then has its properties measured to learn more about the system. The goal of this work is to properly simulate this non-equilibrium aspect by using strong-coupling perturbation theory. While much progress has been made on equilibrium problems, there is limited work on the non-equilibrium dynamics of such strongly correlated systems and, hence, little is known about their behavior. We show in this work that one can solve these problems with an approximate method well-suited to the cold-atom experiments.

2. Numerical Method

Our theoretical approach is based on the so-called strong-coupling expansion (SCE). The SCE has been used for a variety of problems with good success, and can be considered as a fairly established technique, especially in the equilibrium. Early efforts of the strong-coupling expansion for cold-atoms focused on the equilibrium properties of the Bose-Hubbard model (Freericks and Monien, 1996; see also Freericks, et al., 2009). Applications to the Fermi-Hubbard model have showed that they also work very well in the high-temperature regime (Jördens, et al., 2008 and 2010).

The Fermi-Hubbard model (Hubbard, 1963), is a simplified model of fermions moving on a lattice via hopping, and interacting via an on-site repulsion when two fermions of opposite spin occupy the same lattice site (two fermions of the same spin cannot occupy the same lattice site due to Pauli blocking from the Pauli exclusion principle). The Hubbard model can be described by a simple Hamiltonian as follows:

$$H(t) = -J(t)\sum_{<ij>,\sigma}\left(e^{i\vec{A}(t)\left(\vec{r}_i-\vec{r}_j\right)}c^{\dagger}_{i\sigma}c_{\sigma j} + h.c.\right) + U(t)\sum_i n_{i\uparrow}n_{i\downarrow},$$

where

- $J(t)$ is the hopping amplitude between the nearest-neighbor sites on the optical lattice,
- $U(t)$ is the on-site repulsion between two atoms in different hyperfine spin-states on the same optical lattice site,
- $\vec{A}(t)$ is the vector potential determined by the electric field: $\vec{A}(t)=\int_0^t \vec{E}(t')\,dt'$,
- $c_{i\sigma}^{\dagger}$ is the creation operator for a particle at site i and in the hyperfine spin-state σ,
- $n_{i\sigma}$ is the number operator given by $n_{i\sigma}=c_{i\sigma}^{\dagger}c_{i\sigma}$,
- $\vec{r}_i$ is the position of the site i.

The first term represents the kinetic energy due to the hopping of the particles, while the second term is the potential energy due to interactions between the particles. In general, the hopping magnitude, the on-site repulsion and the electric field can be time-dependent within this model. It turns out that this model can be created in a straightforward way in experiments on ultra-cold fermionic atoms with two hyperfine states, such as Li^6 or K^{40}.

For systems where the interactions between atoms are strong, they tend to impede the movement, or hopping of the particles, and the atoms are nearly localized. This occurs because there is only one atom occupying a given lattice site at a given moment in time, and if we have on average one atom per site, the motion is frozen out due to the fact that doubly occupying a lattice site is forbidden by the strong repulsive interaction. These are precisely the conditions in many experiments of ultra-cold atoms on optical lattices. The basic idea of the SCE is to start with the local (or atomic) problem (with zero hopping), which can always be easily solved, and proceed to calculate the corrections as increasing orders in expansion in the hopping. Our focus in this work is on fermionic atoms on an optical lattice, which are described by the Fermi-Hubbard model and can be used to model a number of other physical systems.

The basic quantity that is calculated in many-body physics is the Green's function, which measures the quantum-mechanical amplitude and phase for creating a particle on one lattice site at time t_2 and destroying it at a different lattice site at time t_1:

$$G_{ij}(t_1,t_2)=-i\left\langle T_c c_i(t_1)c_j^{\dagger}(t_2)\right\rangle=-i\frac{\mathrm{Tr}\left[T_c e^{-\beta H(t=0)}c_i(t_1)c_j^{\dagger}(t_2)\right]}{\mathrm{Tr}\left[e^{-\beta H(t=0)}\right]}$$

The thermal average in this formula is performed with respect to the original Hamiltonian at time t=0 and T_c denotes contour ordering of the creation and annihilation operators (see below). All operators are expressed in the Heisenberg picture. The non-trivial quantum-mechanical effects in the Green's function are recorded in the so-called self-energy (which determines how the interactions shift the energies of the quantum states, and how long those states live as coherent excitations). In a non-equilibrium theory, the self-energy becomes of critical importance if one wants to study relaxation or damping effects, because one needs to have a theory that can accurately determine it in order to properly determine the damping. The self-energy is defined as:

$$\mathbf{\Sigma}=\mathbf{G}_0^{-1}-\mathbf{G}^{-1}$$

This definition of the self-energy in the SCE is similar to the conventional (weak-coupling) definition, except that here $\mathbf{G_0}$ is the atomic (not the non-interacting) Green's function. In the simplest picture, this quantity describes a correction to propagation of the atom due to hopping to a neighboring site on the optical lattice and then returning back.

The calculation of non-equilibrium properties is complicated (as compared to an equilibrium calculation), since the absence of the time translation invariance implies that the Green's functions and self-energy describing the system are now continuous matrix operators in two time-variables, defined on the so-called Kadanoff-Baym-Keldysh contour. This contour starts at an early time, runs to a maximal simulated real-time, then back to the early time, and finally includes an imaginary time-branch that is necessary for the initial thermalized boundary condition of the system before the field is turned on (and whose length is given by the inverse of the initial temperature of the system). The time-ordering is with respect to the ordering along the contour, implying that later times on the contour occur on the lower branch rather than the upper branch, which is an earlier time. The Kadanoff-Baym-Keldysh contour is shown in Figure 1 in a discretized form with N_t time-steps, because we need to discretize the continuous matrix operators into discrete matrices in order to perform numerical calculations. Thus, the matrix dimension grows linearly with the maximal simulated real-time. The equations of motion become matrix equations and their solution requires operations with large matrices (such as inversion, contraction with a higher rank tensor, etc.), drastically increasing the computational complexity of the problem.

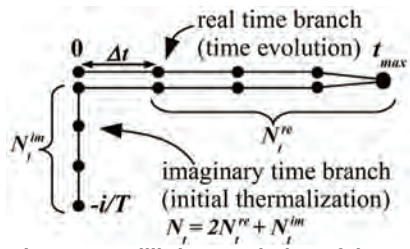


Figure 1. Kadanoff-Baym-Keldysh contour. The non-equilibrium evolution of the system is described by the real-time branches. The imaginary-time branch accounts for the initial thermalized boundary condition of the system. For computational purposes, the contour gets discretized into $N_t = 2N_t^{re} + N_t^{im}$ time-steps.

We now sketch the algorithm used to solve this problem. The atomic problem can always be directly solved to obtain the atomic Green's function. Once that is done, we formulate a series expansion in powers of the hopping (Freericks, et al., 2009). We include terms up to second-order in the hopping, which corresponds to atoms i) hopping once to a neighboring site; ii) atoms hopping twice and ending up two sites away; and iii) atoms hopping to a neighboring site and returning to the original site (see Figure 2). Truncation of terms beyond the second-order is required due to the rapidly growing computational complexity, and results in an often unsatisfactory approximation, as it does not allow for atoms to propagate further than two sites away from the original one. Fortunately, it can be significantly improved by a partial but infinite resummation of terms, which results in a better treatment of the long-range propagation. This level of resummation of terms results in a matrix inversion to solve the linear matrix equation of motion for every point in momentum space (Dyson's equation). The second-order contributions to the propagation can be grouped and expressed through the so-called second-order local cumulant self-energy (further denoted as simply the self-energy). This second-order contribution is especially important in non-equilibrium, as it is required to model relaxation processes, whereby an initially perturbed system relaxes to a stationary long-time solution. Furthermore, we require that in a homogeneous system, the propagation of atoms on the neighboring site is governed by the same Green's function as on the original site. This approximation results in a self-consistency requirement whereby the self-energy depends on the local Green's function, which, in turn, depends on the self-energy. We solve the resultant system of non-linear matrix equations iteratively by choosing a starting approximation (such as taking the local Green's function equal to the atomic Green's function), and proceeding with the self-consistency loop, where sequential iterations produce a more accurate estimate for the unknown quantities. Usually, about ten self-consistency iterations are necessary to reach convergence. The total energy and the potential energy can be obtained from the equal time derivative of the converged local Green's function. Finally, the calculation needs to be repeated for different degrees of time discretization to estimate the systematic error associated with the finite number of time-steps.

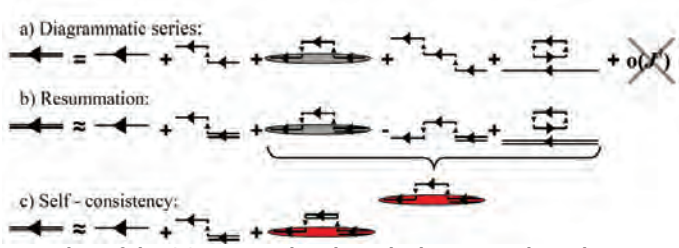


Figure 2. Diagrammatic representation of the SCE approximation: a) Diagrammatic series up to second-order in hopping with higher-order terms truncated. A single line denotes the atomic Green's function, a double line denotes the full Green's function and a dashed line indicates hopping to a neighboring site. The gray shaded oval denotes the two particle local atomic Green's function (Equation 13 in Freericks, et al., 2009). b) Resummation of the diagrams giving rise to Dyson's equation. The second-order term that represents hopping two sites away is generated by the resummation of the first-order term, yet this resummation introduces another local second- order term. The three second-order terms (Equation 23 in Freericks, et al., 2009) can be grouped together into a single-term that involves the two particle cumulant Green's function (red shaded oval; Equation 24 in Freericks, et al., 2009). c) The self-consistency requirement implies that the propagation on the adjacent site in the second-order term is the same as the full Green's function.

The basic algorithm is depicted in Figure 3: i) we start with calculating the atomic Green's function and setting the initial guess for the local Green's function equal to that; ii) we compute the self-energy by evaluating the terms through second-order of the SCE; and iii) we find the momentum dependent Green's function by solving Dyson's equation [this is accomplished by finding the inverse of an $N_t \times N_t$ matrix for each momentum value] and sum them to obtain the local Green's function. Quantities such as the current and average kinetic energy are measured during this step. This loop

[steps (ii)–(iii)] is iterated until the quantities converge. To facilitate the convergence of this iterative process towards a stable physical solution, we introduce a linear mixing parameter for the self-energy, which controls the fraction of the updated self-energy used in the following iteration. Decreasing this parameter stabilizes the self-consistency cycle at the expense of slower convergence. We find that value of 0.5 for this parameter to be adequate for most of our calculations. This results in an exponential convergence towards the stable solution with approximately 10–20 iterations needed to decrease the relative change in self-energy between two successive iterations to less than 10^{-3}.

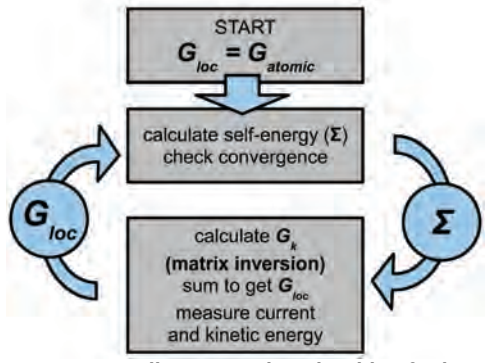


Figure 3. Schematic of the self-consistent, strong-coupling expansion algorithm for homogeneous problems. The calculation of the local Green's function from the local self-energy is a large matrix problem that does not simplify via Fourier transformation. Calculation of the local self-energy is a more expensive operation, but it is easy to parallelize.

The SCE in non-equilibrium has the same limitations as its equilibrium counterpart: it only works if the interaction to hopping ratio is sufficiently large and if the initial temperature is not too low. When these conditions are violated, we see a decrease in the accuracy of the SCE results that is accompanied by a slower (or ultimately failing) convergence of the self-consistency cycle. Normally, the SCE converges down to temperatures smaller than the hopping. This implies that the SCE works very well within the parameter regime of current experiments with ultra-cold atoms on optical lattices.

3. Scaling of the Numerical Algorithm

The algorithm has two natural parallelizable pieces. The first is the solution of the Dyson's equation to solve for the momentum-dependent Green's function. This needs to be done for each value of momentum in the discretized reciprocal (or momentum) space. We use the complex matrix inversion routines ZGETRF and ZGETRI from LAPACK software libraries (Anderson, 1999) for the matrix inversions. Each core performs one matrix inversion at a time, and the contributions to the local Green's function and other measurements are then added together. This part of the algorithm scales as N_t^3 and takes about 5–20% of the total computation time. The number of points in reciprocal space can be adjusted (by choosing an appropriate discretization of reciprocal space) to be a multiple of the number of cores for better scaling.

The other parallel part is finding the self-energy. This is the most expensive part of the algorithm, as it scales as N_t^4. Calculating each element of the self-energy matrix requires $o(N_t^2)$ operations and is computed on a single processor. Usually, we choose both N_t and the number of processors to be powers of two, so that the total number of matrix entries (N_t^2) is always a multiple of the number of processors (also a power of two). We use the BLAS software library (Blackford, 2002) for this part of algorithm. Because the contribution to the self-energy matrix involves calculation of the two-particle Green's function, which is a rank-four tensor and depends on four time-indices and their relative sequential order (and thus there are 24 possible time-orderings), it is not possible to formulate the calculation of the self-energy as a BLAS Level 3 call (matrix–matrix operation). Instead, calculation of a single element of the self-energy matrix uses the BLAS Level 2 routine ZDOTU for a dot product of a range of values of the Green's function, and appropriate lookup arrays for the calculation of the two-particle Green's function.

Aside from these two computationally-expensive steps of the algorithm, the remaining operations add little overhead. Initialization of the data, as well as output of the results usually takes less than 1% of the execution time. Due to the extremely-parallel nature of the calculation, the code runs quite efficiently on parallel machines, with only up to around 1% of the total time spent for communications (see Figure 4).

We illustrate the efficiency of the code by examining the scaling with the number of processors and scaling with the problem size. We start with the strong scaling results in Figure 4. We run the same code on different numbers of processors, chosen to be a fraction of the maximal number of processors needed for a given run. The code shows good linear scaling for small numbers of processors, and fitting the lowest four data points to a line produces the theoretical maximal scaling (red dashed) line shown in the figure. The performance of the actual code lies at about 94% of linear scaling on 4,096 processors, which is excellent performance. The code does not have a significant serial part in the main loop, which is why it scales so well.

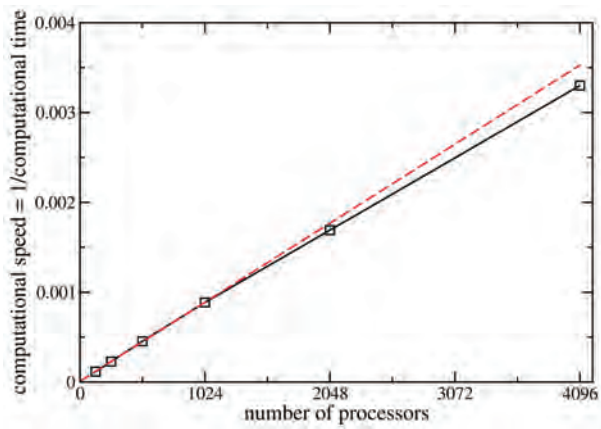


Figure 4. Strong-scaling plot for the SCE algorithm. We achieve 94% of linear-scaling on 4,096 processors. This is an excellent strong-scaling result. These results were obtained for a sample calculation for one iteration of the self-consistency loop with N_t=2,048 time-slices and N_k=32,768 points in momentum space on the AFRL Cray XE6 (raptor).

In Figure 5, we show the results for scaling with the problem size. The code is run for different numbers of time-slices corresponding to simulating different intervals of physical time, as well as different levels of time discretization for a fixed-time interval. Results with different levels of time discretization can be used to assess the error due to the discretization. As expected, most of the computational time is spent calculating the self-energy, which shows a consistent N_t^4 scaling with the number of time-slices. The other computationally-extensive part of algorithm, calculation of the local Green's function, scales as N_t^3 for a fixed-number of momentum points. This part of code scales strictly linearly with increasing the number of momentum points.

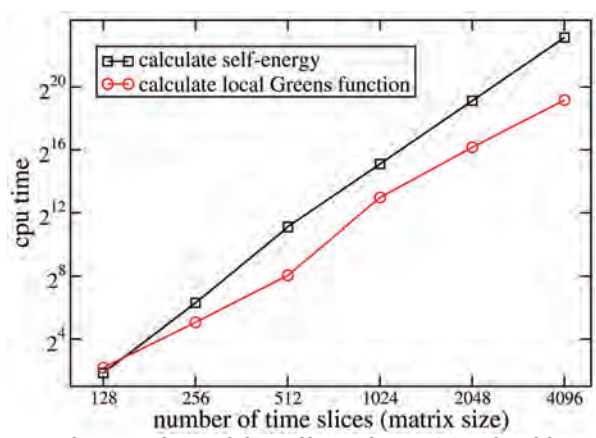


Figure 5. Scaling comparison for two main operations of the self-consistent SCE algorithm on the Cray XE6 (chugach) (at DoD HPCMP Open Research Systems). The dominant part of the calculation is taken to calculate the self-energy, which scales as N_t^4. Calculation of the local Green's function involves inversion of matrices, and thus scales as N_t^3 (A kink in this scaling is due to a better cache utilization for smaller matrices). Using this scaling, we can reliably predict the computation time for even longer times (more time-slices). The number of points in momentum space is N_k=4,096. The number of processors range from 4 (N_t=128) to 2,048 (N_t=4,096), so these results represent good weak-scaling behavior. (Note these results are not plotted in the conventional weak-scaling method, since we show the number of time slices, not the number of processors on the horizontal axis, and the plot is a log-log plot.)

The algorithm is highly-efficient for two reasons: i) the communications is kept to a minimum, as we only have two Message Passing Interface (MPI) matrix reduction and distribution calls per self-consistency iteration, and ii) the matrix operations are handled with LAPACK and BLAS codes that are highly optimized for each processor.

4. Results and Discussion

As an example application, we study fermionic cold-atoms on a three-dimensional optical lattice with the simulated electric field aligned along the diagonal direction (1,1,1) of the lattice (we examine a periodic system, so we do not include the trap potential for the atoms, which makes the system inhomogeneous). In general, our method is not restricted to any number of dimensions or lattice geometries, but we expect it to work better in higher-dimensions, and the three-dimensional problem is the largest computational problem in terms of lattice size and number of particles, so we focus on that case. We study a homogeneous system with number of atoms equal to the number of optical lattice sites ("half-filling"). The "electric field" within the context of the cold-atom experiments can be created in two different ways: i) a linear potential can be added, whose slope is the magnitude of the electric field, or ii) the lattice potential can have the two retro-reflected beams shifted in frequency, so the system has a standing wave in a moving frame, and the lattice is pulled through the cloud of atoms (with the rate of pulling being proportional to the electric field). In equilibrium, a strongly interacting system at half-filling is characterized by a "Mott-insulating" state (Mott, 1949), where the hopping of particles is strongly suppressed due to the energy penalty for two particles to occupy the same lattice site. In this study, we keep the interaction U and the hopping J fixed, so that the time-dependence of the problem comes due to the electric field being switched on.

As a first result, we show the effect of a time-independent field to produce Bloch oscillations in the current and the corresponding density distributions. Bloch oscillations arise as a response of a periodic system to a time-independent field, due to the electric field shifting the momentum of atoms at a constant rate and momentum being reflected off the Brillouin zone boundaries when the momentum reaches the edge of the Brillouin zone (Bloch, 1929). The frequency of the Bloch oscillations is determined by the strength of the electric field, while the amplitude is determined by the initial density distribution (itself a function of temperature) and various other factors (such as the interactions) that determine the relaxation of the system to a stationary state.

In Figure 6, we show the resultant Bloch oscillations in the current after an application of a relatively strong time-independent field E=4.0 (in the units of the on-site repulsive interaction U between the atoms) for three values of the initial temperature. The current shows rapid oscillations, with period $2\pi/E$. As expected, the amplitude is higher for the lower-temperature cases. The amplitude modulation has a period determined by the interaction U.

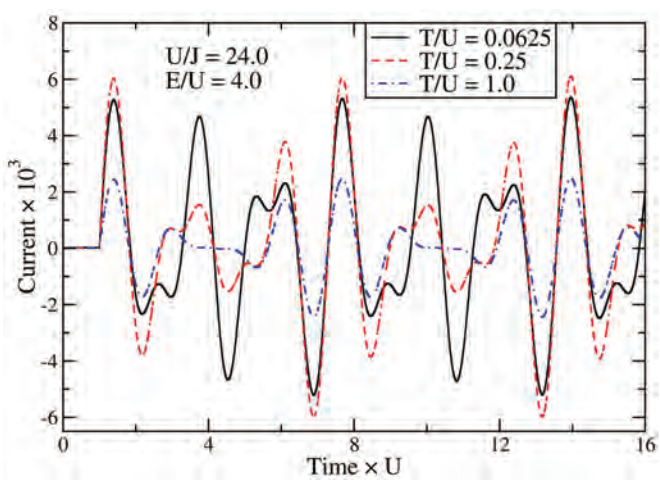


Figure 6. Plot of the current after a constant time-independent electric field E=4.0 is switched on at time *t*=1.0. The current shows Bloch oscillations with the frequency determined by the strength of the field and increasing amplitude with lower- temperature. The oscillations do not show a significant damping, but do have an overall amplitude modulation. The system is a 3D optical lattice half-filled with fermion cold-atoms, so that on average there is one atom per site. The electric field is aligned along the diagonal (1,1,1) direction of the lattice.

The Bloch oscillations also manifest themselves in the expectation value of the total energy. However, since the total energy is obtained from the equal time-derivative of the Green's function, it has a comparatively large dependence on the time-step used in the simulation. More precise data can be obtained by repeating the calculation for at least two different

values of the time-step, and by performing a scaling of the data towards the limit of zero time-step (see Figure 7). For strong-fields, the change in the total energy at short-times is almost entirely due to the changing kinetic energy, whereas the potential energy (due to doubly occupied sites) increases much more slowly.

The change of the total energy is due to Joule heating caused by the finite current after the field is turned on. This dissipated energy can be calculated by integrating the current times the field. Since we have obtained the data for the total energy and the current independently, we can verify how well our results satisfy total energy conservation (see Figure 8). Here, we show data for the case when field is equal to the interaction. In this case, the total energy grows rapidly and oscillates around the infinite temperature equilibrium value of $U/4$. The current shows a single peak responsible for the heating of the system. The total energy obtained by integrating the current times the field agrees well with the data obtained from the derivative of the Green's function, yet there is a constant offset at long-times. The agreement between the measured current and the current obtained from the time-derivative of the total energy is better, but not perfect. Since extrapolated data are used in Figure 8, it is likely that the slight disagreement in energy conservation is due to the nature of the approximation, which does not include all of the terms in the strong-coupling expansion, or it might be that the step-size was not small enough to be in the scaling regime and allow us to accurately go to the zero step-size limit. The energy conservation for larger values for field is considerably better than the $E=U$ case.

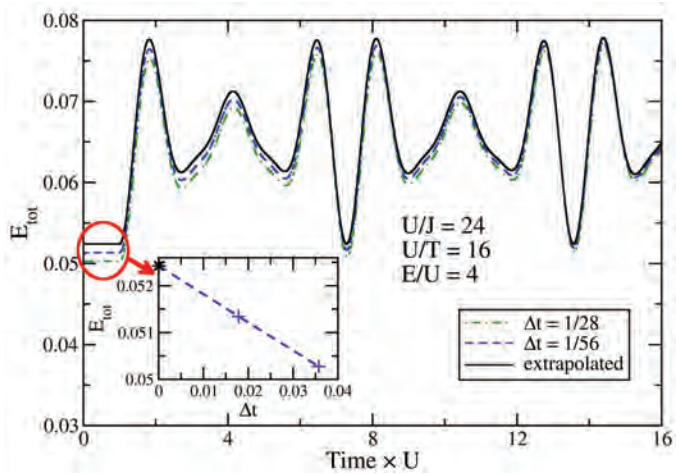


Figure 7. The total energy for hopping *U/J*=24, temperature *U/T*=16 and field *E/U*=4. Data for two different values of time-steps are used to extrapolate towards the $\Delta t \rightarrow 0$ limit. The inset shows scaling with the time-step size of the total energy at an early time before the field is turned on. The extrapolated data in this case coincide with the equilibrium value of the total energy obtained by simulating a very-short time-period with zero-field (black star in the inset).

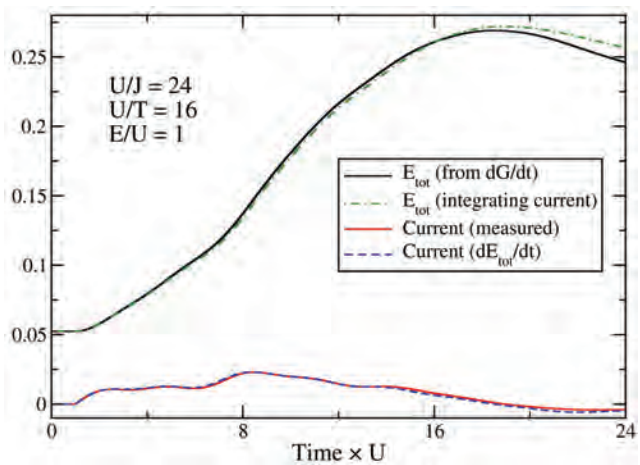


Figure 8. Energy conservation: The energies obtained from the derivative of the Green's function (solid black line) and by integrating the current time-field (green dash-dotted line) show a very good agreement; however, at long-times there appears to be a constant energy shift. Alternatively, one can compare the measured current (solid red line) with the one obtained by differentiating the total energy with time (dashed blue line) – these data show a better agreement, with small differences at some regions where the current changes most rapidly.

Another way to test to accuracy of the calculation is to compare the data for the spectral moments to the exact theoretical values. The spectral moments are linked to the derivatives of the retarded Green's function, and this relation also holds in the non-equilibrium. Furthermore, the exact values for these moments can be found (Turkowski and Freericks, 2006). The zeroth spectral moment is simply the integral of the spectral function, which is equal to one. The first moment is zero for a half-filled system due to the particle-hole symmetry, which means that the spectrum is an even function. The expression for the second moment is $U^2/4+6J^2$ for a three-dimensional (3D) system at half-filling. In Figure 9, we compare our results for the zeroth and second spectral moments (obtained from the retarded Green's function) with the exact values. In both cases, the small deviations from the exact values vanish as the results are scaled towards the limit of zero time-step size. The scaling towards the exact values is quadratic in the time-step size.

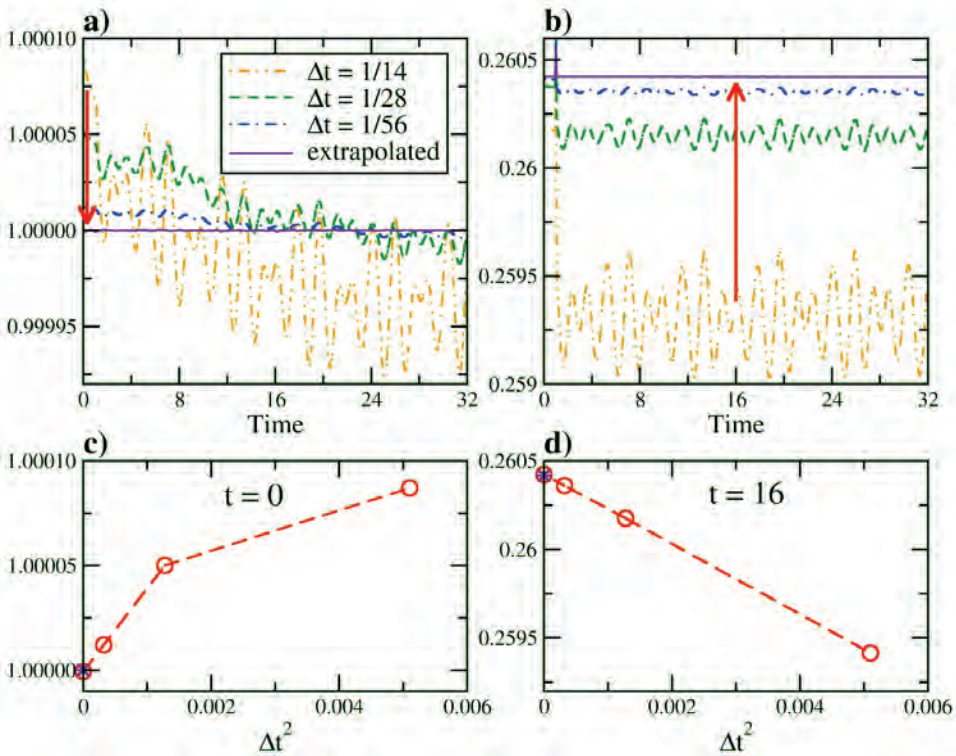


Figure 9. Scaling of the spectral moments for *U/J*=24, *U/T*=16 and *E/U*=4. a) and b): zeroth and second spectral moments for three values of the time-step. For a finite-time-step, the moments slightly deviate from the exact value and slightly vary with time. The extrapolated values match perfectly with the exact value for the whole time interval. c) and d): scaling of the zeroth and second moments towards the Δ *t* → 0 limit. Both cases show a perfect scaling with Δ t^2 for small Δ *t* and a perfect agreement between the extrapolated and the exact values.

Finally, we show the gauge-invariant momentum distribution of the atoms in Figure 10 (Bertoncini and Jauho, 1991). In the case when the field is along the diagonal of the lattice, the momentum distribution of a 3D system (normally dependent on three momentum projections) can be parametrized by only two variables: $\varepsilon=-2[\cos(k_x)+\cos(k_y)+\cos(k_z)]$ (related to band-energy) and $\eta=-2[\sin(k_x)+\sin(k_y)+\sin(k_z)]$ (related to the component of the velocity along the direction of the field), and in the (ε,η) plane all momentum points fall within a circle with radius 6. Initially, before the field is switched on (at time t=1.0), the system is in equilibrium, and the momentum distribution only depends on the "band-energy" variable ε. After the field is switched on, the distribution starts to rotate with a frequency determined by the Bloch oscillations. With time, the range of variation in the momentum distribution changes following the modulation of the Bloch oscillations in the current.

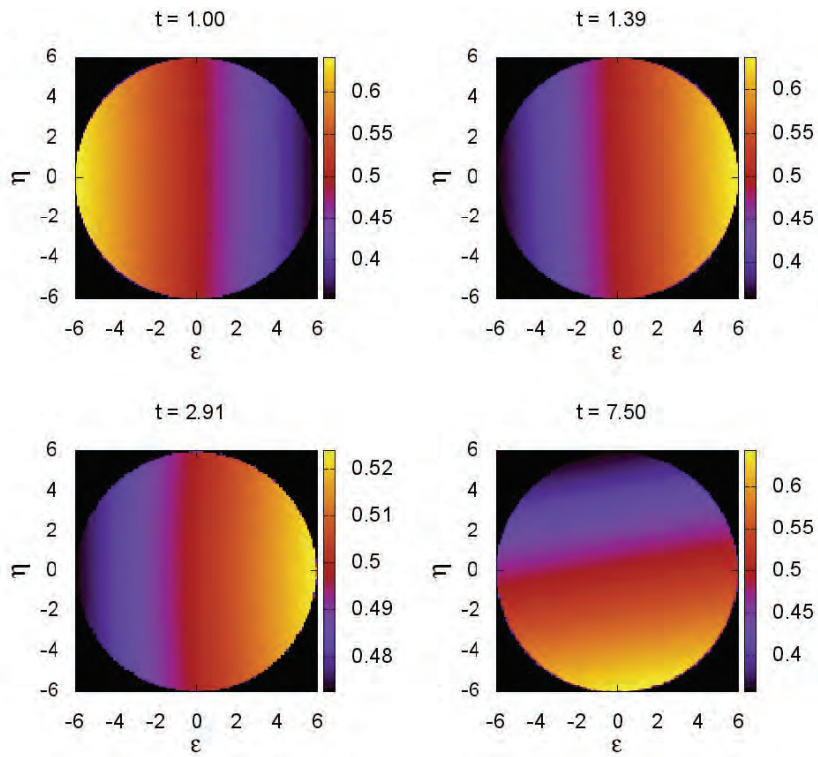


Figure 10. Momentum distribution of the atoms after a time-independent electric field (E=8.0) is switched on at time *t*=1.0. The momentum distribution is parametrized by two variables ε ("band-energy") and η ("velocity along the direction of the field"). The top left picture shows the distribution in equilibrium, just before the field is switched on. The rest of the pictures show the distribution at later times, when it is significantly perturbed away from the thermal equilibrium. These changes occur primarily in the rotation of the line dividing high- and low-occupancy and in the fact that range of the occupation changes in time, as can be seen by the changing scales in the panels at long-times.

5. Significance to the DoD

The DARPA-sponsored program for creating an optical lattice emulator has had significant success with equilibrium simulators, showing how such analog quantum computers can work in a wide-variety of different platforms. The next steps beyond this work can go in a number of different directions. One can try to add more realistic materials properties to the models, and use the simulators as a virtual materials design laboratory for discovering smart materials with specified properties; or one can take these systems into the nonequilibrium domain and use them to understand the behavior of these systems under the presence of large driving fields, which is likely to occur in military applications. It is in this latter direction that the current work is moving, and we hope that our techniques will allow us to have more to say in the future on this interesting novel frontier in physics.

6. Conclusion

We have introduced a new method for studying non-equilibrium quantum lattice systems in the strong-coupling regime. This method incorporates the effects of damping that are essential to describe the non-equilibrium behavior. However, this improved approximation has a computational complexity that scales as the fourth-power of the number of points in simulated-time interval and, hence, must be limited, in the end, to rather short times. We survey the essential parts of the algorithm and show how an efficient parallel implementation can be achieved. We have investigated the Bloch oscillations in the case of a strong field and large interactions. We also showed the evolution of the momentum distribution due to the presence of a driving field. Since the time-dependence of parameters can be arbitrarily varied, this method can readily be used to simulate a number of different experiments with cold-atoms. Future work will involve extending this method to treat inhomogeneous systems to include the trap, and also to model bosonic particles.

Acknowledgments

JKF acknowledges support from the Army Research Office Grant Number W911NF0710576, with funds from the DARPA OLE program. JKF and KM were also supported by the AFOSR under the MURI program from grant number FA9559-09-1-0617. JKF also acknowledges the McDevitt bequest at Georgetown. HRK acknowledges support from the DST (India). The collaboration between the US and India was supported by grant number JC-18-2009 of the Indo-US Science and Technology Forum. DoD HPC computer time was provided on Cray XE6 machines located at the US Air Force Engineering and Research and Development Center (AFRL) and the US Army Engineering Research and Development Center (ERDC). More specifically, we used the following machines: chugach, garnet, and raptor. The computational work was supported primarily by the Challenge Project DARPA-C4J.

References

Anderson, E., Z. Bai, C. Bischof, S.Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, LAPACK Users' Guide, 3rd Ed., *Society for Industrial and Applied Mathematics*, Philadelphia, PA, 1999; see also http://www.netlib.org/lapack/.

Bertoncini, R. and A.P. Jauho, "Gauge-invariant formulation of the intracollisional field effect including collisional broadening." *Physical Review B*, 44, pp. 3655–3664, 1991.

Blackford, L.S., J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R.C. Whaley, "An Updated Set of Basic Linear Algebra Subprograms (BLAS)." *ACM Transactions on Mathematical Software*, 28-2, pp. 135–151, 2002; see also http://www.netlib.org/blas/.

Bloch, F., "Über die Quantenmechanik der Elektronen in Kristallgittern." *Zeitschrift für Physik*, 52, pp. 555–600, 1929.

Feynman, R.P., "Simulating physics with computers." *International Journal of Theoretical Physics*, 21, pp. 467–488, 1982.

Freericks, J.K. and H. Monien, "Strong-coupling expansions for the pure and disordered Bose Hubbard model." *Physical Review B*, 53, pp. 2691–2700, 1996.

Freericks, J. K., H.R. Krishnamurthy, Y. Kato, N. Kawashima, and N. Trivedi, "Strong-coupling expansion for the momentum distribution of the Bose-Hubbard model with benchmarking against the exact results." *Physical Review A*, 79, pp. 053631-1–22, 2009.

Hubbard, J., "Electron correlations in narrow energy bands." *Proceedings of the Royal Society of London, Series A, Mathematical and Physical Sciences*, 276, pp. 238–257, 1963.

Jördens, R., N. Strohmaier, K. Günter, H. Moritz, and T. Esslinger, "A Mott insulator of fermionic atoms in an optical lattice." *Nature*, 455, pp. 204–207, 2008.

Jördens, R., L. Taruell, D. Greif, T. Uehlinger, N. Strohmaier, H. Moritz, T. Esslinger, L. De Leo, C. Kollath, A. Georges, V. Scarola, L. Pollet, E. Burovski, E. Kozik, and M. Troyer, "Quantitative determination of temperature in the approach to magnetic order of ultra-cold fermions in an optical lattice." *Physical Review Letters*, 104, pp. 180401-1–4, 2010.

Mott, N F., "The Basis of the Electron Theory of Metals, with Special Reference to the Transition Metals." *Proceedings of the Physical Society, Section A*, 62, pp. 416–422, 1949.

Turkowski, V.M. and J.K. Freericks, "Spectral moment sum rules for strongly correlated electrons in time-dependent electric fields." *Physical Review B*, 73, pp. 075108-1–15, 2006; Erratum, *Physical Review B*, 73, pp. 209902(E)-1, 2006.

6.2 Applied Research

Air Force

Can Legacy Codes Scale on Tens of Thousands of PEs or Do We Need to Reinvent the Wheel?

Robert F. Tomaro, William Z. Strang, and Kenneth E. Wurtzler
Cobalt Solutions, LLC, Springfield, OH
{tomarorf, wurtzler, strang}@cobaltcfd.com

Abstract

Over the last several years, there have been many discussions involving the potential scalability of legacy codes/numerics concerning their ability to fully-utilize large core-count machines. The prevailing thought and conclusion of those involved in the discussions were that legacy codes could not scale well on thousands of processors, let alone tens of thousands of processors. Therefore, new codes using new numerics had to be developed. In fact, a large Department of Defense (DoD) effort was established 5 or 6 years ago tasked with developing the new numerics required to fully-utilize hundreds of thousands of processor machines. In light of the recently published scalability results of the DoD's effort at computational fluid dynamics code development, we decided to retest the scalability of Cobalt to see if the above conclusion and results held true for that legacy code.

1. Introduction

In the early 2000s, *Cobalt*[1,2] was used for benchmarking new high performance computing (HPC) machines. This included scalability tests on *KRAKEN*, *jvn*, *eagle*, and *sapphire*. NAVO received the 2,048 processor IBM, *KRAKEN*, as a TI-04 machine[3]. The first Capability Application Project (CAP) was completed on *KRAKEN* including an F/A-18E free-to-roll simulation using *Cobalt*. The US Army Research Laboratory (ARL) received the 2,048 processor Linux-based machine, *jvn*, in TI-04, while the US Air Force Research Laboratory (AFRL) received the 2,048 PE SGI, *eagle*[4], as a TI-05 machine. Finally, *sapphire* was installed at the US Army Engineer Research and Development Center (ERDC) from TI-05[5]. The machine was a Cray XT3 containing 4,176 processors. At that time, *sapphire* was the most powerful supercomputer in the Department of Defense (DoD). The last time the scalability of *Cobalt* was performed for HPC was on *sapphire* in 2005.

Figure 1 shows the 2005 scalability results for *Cobalt* for *KRAKEN*, *jvn*, *eagle*, and *sapphire*. The chart shows the super-linear speed-up of *Cobalt* on those four machines. Super-linear speed-up is defined as scalability efficiency greater than 100%. Of particular interest are the results on *sapphire*. These results are often quoted as the best computational fluid dynamics (CFD) scalability obtained by a production-level code. However, it is often said that *Cobalt* scales up to 5,000 processors. There are two problems with this statement. First, the results on *sapphire* were on 4,000 processors. Second, and more importantly, the phrase "scales-up to" implies that *Cobalt* did not scale beyond 4,000 processors. In actuality, we simply ran out of machine. The results on *sapphire* clearly show that scalability efficiency was still increasing at the processor limit. Additionally, the grid-size is never specified when these results are mentioned. We will show later that the number of cells per processor is a critical parameter when measuring scalability efficiency. We state here that the grid used in the 2005 scalability study contained 6.5 million cells.

Figure 2 shows the scalability of CREATE's *Kestrel* v2.1b as presented by Morton[6] at the 2011 HPC Users Group Conference. These results are for a 47 million cell grid on the Cray XE6, *raptor*, located at AFRL. The CFD solver in *Kestrel* v2.1b is kAVUS, which is based on the legacy code AVUS from AFRL[7]. The efficiency of *Kestrel*'s scalability is under 100% on 2,048 PEs. On 4,096 PEs, *Kestrel*'s scalability is 65%, which was deemed acceptable by Morton, et al.

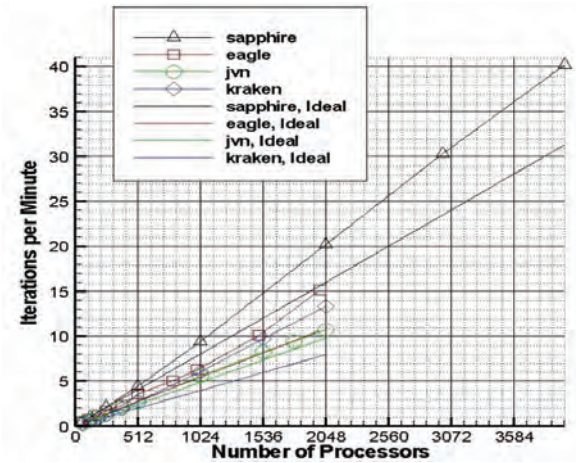


Figure 1. Scalability results for Cobalt circa 2005

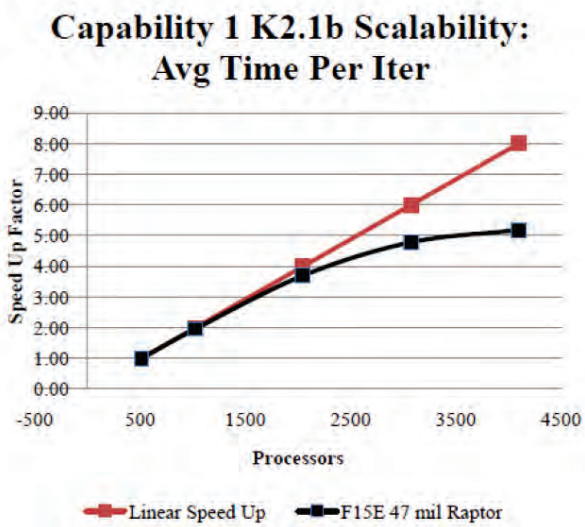


Figure 2. Scalability of *Kestrel* v2.1b in 2011

2. Current Scalability Results for *Cobalt*

We decided to update the *Cobalt* scalability charts for a few reasons. First, it has been seven years since the benchmarking was performed on *sapphire*. Second, there have been many modifications to the internal workings of *Cobalt* in the time since. Third, machine architecture has changed from single-core machines to multi-core machines, which could easily affect scalability results. Lastly, the current scalability results of *Kestrel* v2.1b on *raptor* show, in our opinion, poor scalability. This could imply that *Cobalt*, another so-called legacy code, will not scale well on the current architectures at HPC, which must be investigated. We will present scalability results on three different single-grid meshes, as well as one overset case. The largest single-grid case will be compared with *Kestrel* v2.1b.

A. 3.15 Million Cell Grid

The 3.15 million cell grid is a hybrid grid around a JDAM, which is similar to the store from the F-18C store separation challenge of 1999[8]. The grid has roughly 2.7 million tetrahedrons and 430K prisms. The baseline number of PEs, or the minimum number of PEs, used on this case was 32. This baseline is used to determine the scalability on greater numbers of PEs and, typically, one tries to make the baseline as small as possible. The scale factor at the baseline will always be one. The 3.15 million cell grid was run on 32, 64, 128, 258, 512, 1,024, 2,048, and 4,096 PEs on *raptor* using *Cobalt* V5.2. Figure 3 provides the scalability curve for this grid.

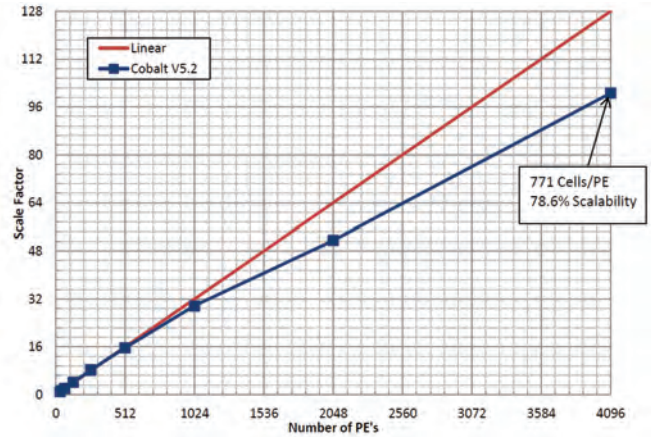


Figure 3. *Cobalt* scalability on *Raptor* for a 3.15 million cell grid

Table 1. Results for 3.15 million cell grid

Number of PE's	Cells per PE	Scalability Efficiency
32	98663	100.0%
64	49332	102.9%
128	24666	103.2%
256	12333	103.5%
512	6166	98.1%
1024	3083	93.0%
2048	1542	80.5%
4096	771	78.6%

Table 1 presents scalability efficiency as a function of processor count and number of cells per processor. For this grid, *Cobalt* has super-linear speed-up for 64, 128, and 256 PEs. This is due to enhanced cache performance as the number of cells per PE decreases. Even at 771 cells per PE, *Cobalt* is achieving nearly 80% scalability efficiency. Below this number of cells per PE, communication is starting to cost more than the computation, thereby decreasing scalability efficiency.

B. 10 Million Cell Grid

The 10 million cell grid is a hybrid grid around an F-18C. This grid has roughly 8 million tetrahedrons and 2.1 million prisms. The baseline number of PEs used on this case was 32. This grid contains several types of boundary conditions including no-slip solid walls, far-field, engine exhaust and inlets. The 10 million cell grid was run on 32, 64, 128, 258, 512, 1,024, 2,048, 4,096, 8,192, and 15,360 PEs on *raptor* using *Cobalt* V5.2. Figure 4 provides the scalability curve for this grid.

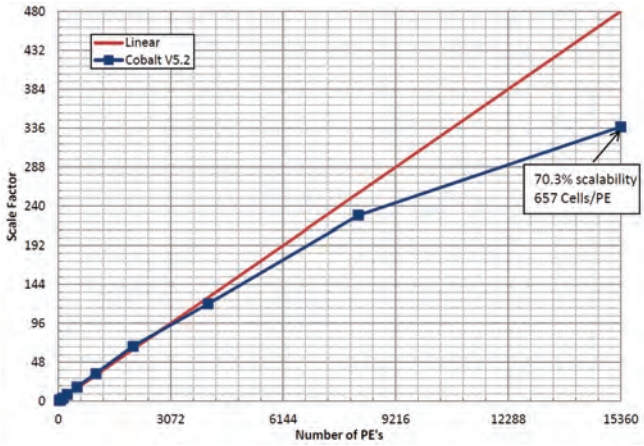


Figure 4. *Cobalt* scalability on Raptor for a 10 million cell grid

Table 2 shows scalability as a function of processor count and number of cells per processor. For this grid, *Cobalt* has super-linear speed-up from 64 to 2,048 PEs. Even at 657 cells per PE, *Cobalt* is achieving 70% scalability efficiency.

Table 2. Results for the 10 million cell grid

Number of PE's	Cells per PE	Scalability Efficiency
32	315319	100.0%
64	157659	104.4%
128	78830	106.4%
256	39415	108.3%
512	19707	111.4%
1024	9854	106.8%
2048	4927	106.3%
4096	2463	93.8%
8192	1232	89.3%
15360	657	70.3%

C. 49 Million Cell Grid

The 49 million cell grid is a hybrid grid around a C-17. This grid has roughly 44 million tetrahedrons and 5.3 million prisms. The baseline number of PEs used on this case was 96. This grid contains several types of boundary conditions including no-slip solid walls, far-field, engine exhaust and inlets. We selected this size of grid to be comparable to the *Kestrel* results in Figure 2. This grid was run on 96 through 21,856 PEs on *raptor* using *Cobalt* B5.11. Twenty-one thousand, eight-hundred fifty-six PEs is half the number of PEs of *raptor* and can be accessed through the regular queue. Figure 5 provides the scalability curve for this grid.

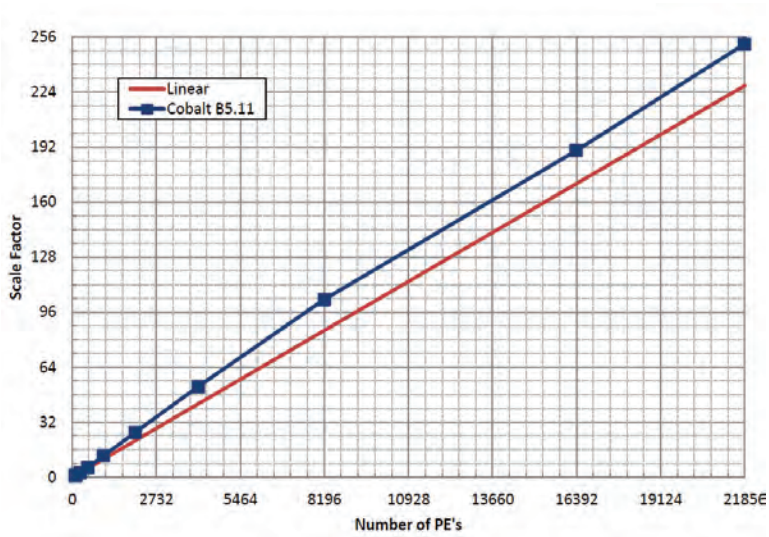


Figure 5. *Cobalt* scalability on Raptor for a 49 million cell grid

During initial scalability testing of *Cobalt* V5.2, we discovered that the grid partitioning phase began to require excessive CPU time once the number of PEs exceeded around 6,000. In fact, the grid partitioning step required 50 minutes of wall-clock time on 8,192 PEs. This was deemed unacceptable. Diagnosis revealed that the problem was in the integration of *Cobalt* with the ParMETIS graph partitioning software[9]. After a couple hours of rewriting code in our current beta release, we were able to reduce the grid partitioning phase to 50 seconds on 8,192 PEs and 124 seconds on 21,856 PEs. We therefore decided to use our current beta, *Cobalt* B5.11, for the scalability tests for the 49 million cell grid. This version also uses the latest release of ParMETIS, V4.0.

Table 3 shows scalability as a function of processor count and number of cells per processor. For this grid with this baseline, *Cobalt* has super-linear speed-up from 256 to 21,856 PEs.

Table 3. Results for the 49 million cell grid

Number of PE's	Cells per PE	Scalability Efficiency
96	514662	100.0%
128	385996	100.0%
256	192998	104.9%
512	96499	109.3%
1024	48250	116.3%
2048	24125	122.4%
4096	12062	123.8%
8192	6031	121.1%
16384	3016	111.3%
21856	2261	110.7%

D. 13 Million Overset Grid

Lastly, we wanted to test the scalability of an overset case. The 13 million cell grid is a combination of the F-18C and JDAM grids using the Overset module available in *Cobalt*. The baseline number of PEs used on this case was 32. Unfortunately, each overset case will scale differently because of the hole-cuts being different for each case. The Overset module does not scale as well as a single-grid case. This case was run on 32, 64, 128, 258, 512, 1,024, 2,048, and 4,096 PEs on *raptor* using *Cobalt* V5.2. Figure 6 provides the scalability curve for this Overset grid.

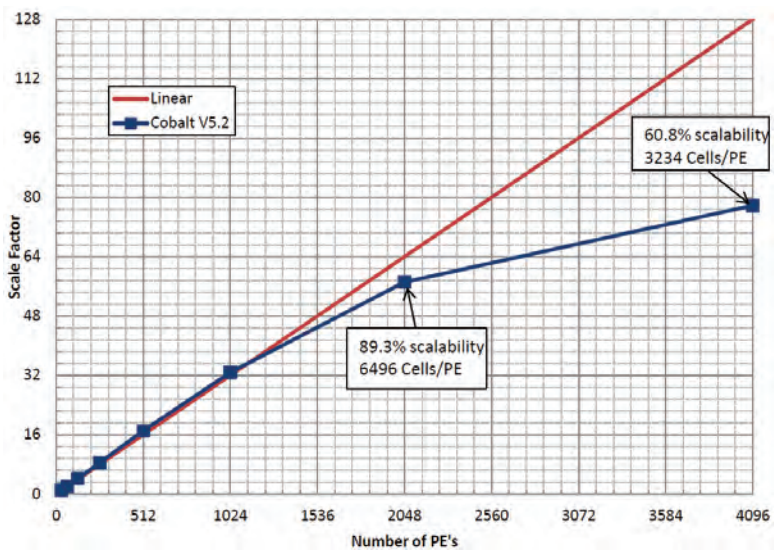


Figure 6. *Cobalt* scalability on Raptor for a 13 million cell grid

3. Conclusion

The goals of this study were: 1) To update the scalability charts of *Cobalt*; 2) To see if *Cobalt* still scaled well on modern machine architectures, with tens of thousands of processors; and 3) To investigate the validity of the general conclusion that no legacy code can achieve this level of scalability, requiring the development of new numerics.

Figures 3, 4, and 5 show the scalability of *Cobalt* for various size grids. All three figures prove that *Cobalt* is highly-scalable even on tens of thousands of processors. To compare the scalability of the three different grid sizes, scalability efficiency based on the number of cells per PE is plotted in Figure 7. We set the baseline for each grid to the data point that was closest to 100,000 cells per PE. For the 3.15 million cell grid and the 10 million cell grid, the curves are very similar. With this baseline, super-linear speed-up is obtained when the number of cells per processor is greater than 8,000. A scalability efficiency of around 90% is achieved for 1,500 to 2,000 cells per PE or larger and 80% scalability efficiency is obtained for roughly 1,000 cells per PE. The 49 million cell grid shows even better scalability performance. This is due in part to changes in *Cobalt* from V5.2 to B5.11 and to the use of ParMETIS V4.0. Figure 7 provides some guidance for users to estimate the number of PEs they should request to obtain the best scalability for their grid.

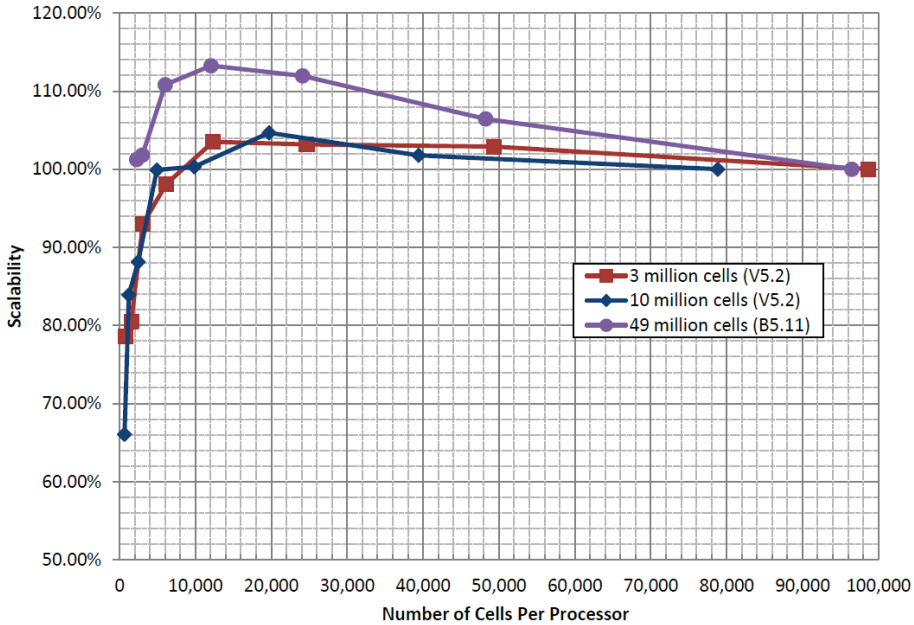


Figure 7. Scalability efficiency versus number of cells per PE

In order to compare the scalability of *Cobalt* and the current state-of-the-art CFD effort of the CREATE program, whose supposed task is to develop software that scales well on hundreds of thousands of PEs, we used a grid similar in size to their grid (49 million versus 47 million) and ran on the same machine they ran (*raptor*). To be fair, the following comparison also uses a baseline of 512 PEs, which is the value Morton, et al., used for *Kestrel* v2.1b, and not the baseline value of 96 PEs used in the *Cobalt* results shown in Figure 5. Figure 8 compares the scalability of *Cobalt* and *Kestrel*. At the maximum number of PEs that Morton, et al., presented, *Kestrel* v2.1b has 65% scalability efficiency on 4,096 PEs whereas *Cobalt* achieves 113% scalability efficiency on 4,096 PEs. In fact, *Cobalt* continues to be super-linear all the way up to 21,856 PEs which is the maximum number we used. Further, *Kestrel* has a scalability efficiency of 65% with 11,475 cells/PE. The lowest scalability efficiency of *Cobalt* found in this study is 70.3% with 657 cells/PE. Using Tables 1–3 and Figure 7, we estimate that *Cobalt* would have scalability efficiency greater than 80% on 60,000 PEs with this 49 million cell grid.

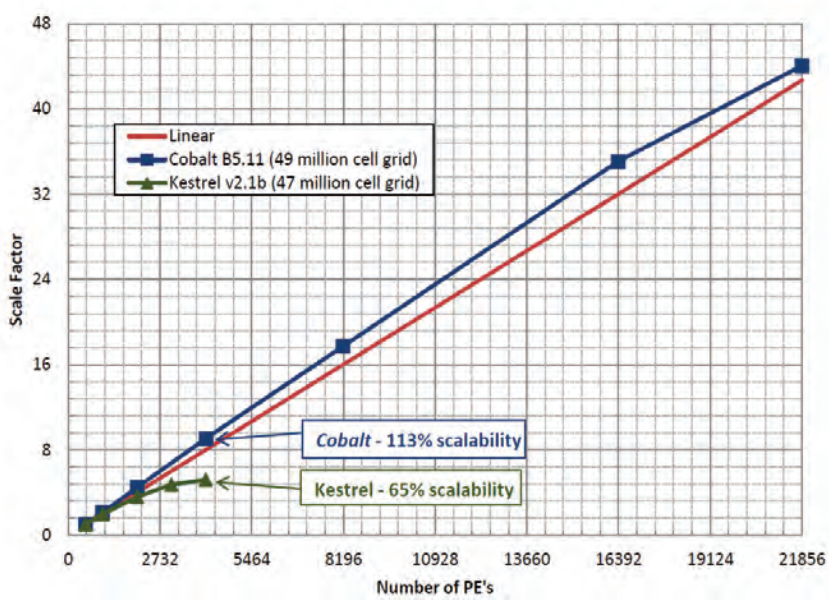


Figure 8. Comparison of *Cobalt* and *Kestrel* scalability on Raptor

The current versions of *Cobalt*, V5.2 and B5.11, both available on most HPC machines, have proven to be highly-scalable, scaling well on tens of thousands of processors. We have also shown that *Cobalt* clearly has the potential to scale well on hundreds of thousands of processors, given a grid size of roughly 85 million cells or larger. We have shown that scalability efficiency for *Cobalt* can be roughly predicted based on a cells-per-PE basis. Finally, the general conclusion that so-called legacy codes cannot scale well on tens to hundreds of thousands of processors has been shown to be erroneous.

References

1. Strang, W.Z., Tomaro, R.F., and Grismer, M.J., "The Defining Methods of Cobalt: A Parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver." *AIAA Paper 1999-0786*, 1999.

2. Tomaro, R.F., Strang, W.Z., and Sankar, L.N., "An Implicit Algorithm For Solving Time Dependent Flows on Unstructured Grids." *AIAA Paper 1997-0333*, 1997.

3. *Navigator*, NAVO MSRC, Spring 2005.

4. *Wright Cycles*, ASR, Spring 2006.

5. ERDC MSRC, *Resource*, Fall 2005.

6. Morton, S.A., Eymann, T.A., Lamberson, S.R., McDaniel, D.R., Sears, D.R., Tuckey, T.R. and Utrilla, J., "Rigid and Maneuvering Aircraft Results for Kestrel v2 with Kestrel v3 Design Attributes." *DoD HPCMP User's Group Conference*, Portland OR, 21–23 June 2011.

7. Douglas, C.C. and Zornes, A.F., "Computational Fluid Dynamics (CFD) Modeling and Analysis, Delivery Order 0006: Cache-Aware Air Vehicles Unstructured Solver (AVUS)." *AFRL-VA-WP-TM-2006-3009*, August, 2005.

8. Tomaro, R.F, Witzeman, F.C., and Strang, W.Z., "A Solution on the F-18C for Store Separation Simulation Using Cobalt_{60}." *AIAA Paper 1999-0122*.

9. Karypis, G. and Kumar, V., "Multilevel Algorithms for Multi-Constraint Graph Partitioning." *Supercomputing*, May 5, 1998.

CFD Calculation of Aerodynamic Indicial Functions for a Generic Fighter Configuration

Martiqua L. Post, Mehdi Ghoreyshi, and Russell M. Cummings
Department of Aeronautics, US Air Force Academy (USAFA), Colorado Springs, CO
{martiqua.post, mehdi.ghoreyshi, russ.cummings}@usafa.edu

Abstract

A reduced-order modeling of nonlinear and unsteady aerodynamics based on indicial (step) response functions and Duhamel's superposition integral is presented. These time-domain models could predict the unsteady aerodynamic responses of an aircraft performing any arbitrary motion over a wide flight regime, but require calculating a large number of response functions. A method to efficiently reduce the number of indicial response calculations is tested. This method uses a time-dependent surrogate model (input/output mapping) to fit the relationship between flight conditions and response functions from a limited number of response simulations (samples). Each sample itself is directly calculated from unsteady computational fluid dynamic simulations and a grid-motion tool. An important feature of this approach is uncoupling the effects of angle-of-attack and pitch-rate from pitching motions. The aerodynamic models are then created with predicted indicial functions at each time-instant using the surrogate model. This model is then applied for aerodynamics modeling of a generic fighter configuration performing arbitrary pitching and plunging motions at various Mach numbers. Due to the mesh size and a large number of time-steps required to simulate the aircraft response functions, all simulations were performed on Department of Defense high performance computing machines. Results presented show that reduced-order models can accurately predict time-marching solutions of aircraft for a wide range of motions, but with an advantage that reduced-order model predictions require on the order of a few seconds once the model is created. The results also demonstrate that the surrogate model being tested aids in reducing the overall computational efforts to develop reduced-order models.

1. Introduction

The unsteady aerodynamic forces and moments acting on a fast maneuvering fighter aircraft can have a significant effect on the aircraft calculated stability and control characteristics.[1] Some observed unsteady aerodynamic phenomena include aircraft buffeting, wing rock, roll reversal, and directional instability.[1] The aero-elastic instabilities of flutter or limit cycle oscillations are also associated with unsteady aerodynamic loads.[2] Despite great efforts using the best available predictive capabilities, nearly every major fighter program since 1960 has had costly nonlinear aerodynamic or fluid-structure interaction issues that were not discovered until flight testing.[3] Some recent aircraft experiencing unexpected characteristics are the F/A-18, F-18E, and F-22.[3,4] The lack of a full understanding of the unsteady aerodynamics typically leads to ''cut-and-try'' efforts, which result in very expensive and time-consuming solutions.[5] The current computational fluid dynamics (CFD) tools have recently become credible for modeling unsteady nonlinear physics and, hence, would help to reduce the amount of wind tunnel and flight testing required.[6] With advanced computing techniques, one straightforward way to calculate unsteady aerodynamics of a maneuvering aircraft is to develop a full-order mathematical model based on direct solution of the discretized Navier-Stokes equations coupled with the dynamic equations governing the aircraft motion.[7] A full-order model for stability and control analysis is a computationally very expensive approach, since such a model requires a large number of coupled computations for different values of motion frequency and amplitude. An alternative approach to solving the full-order model is to develop a Reduced Order Model (ROM) that seeks to approximate CFD results by extracting information from a limited number of full-order simulations. Ideally, the specified ROM can predict aircraft responses over a wide range of amplitudes and frequencies in a fraction of a few seconds of computational time, without the need of running CFD again. This paper considers the development of ROMs based on indicial functions

that allow prediction of pitching and plunging responses of a fighter aircraft within the frequency/amplitude/Mach number space.

The transient aerodynamic response to a step-change in a forcing parameter, such as angle-of-attack or pitch-rate is a so-called indicial function. Assuming that the indicial functions are known, the aerodynamic forces and moments induced in any arbitrary maneuver can be estimated in the time-domain by means of the well-known Duhamel superposition integral.[8] The indicial functions can be derived from analytical, CFD, or experimental methods.[9] Limited analytical expressions of indicial functions exist for two-dimensional airfoils. For incompressible flows, Wagner[10] was the first who detailed the analytical unsteady-lift of a thin airfoil undergoing a plunging motion using a single indicial function (the so-called Wagner's function) with its exact values known in terms of Bessel functions. For unsteady, compressible flow past two-dimensional airfoils, Bisplinghoff, et al.[11] also described an exponential approximation to the exact solutions of indicial functions at different Mach numbers. However, these analytical expressions are not valid for aircraft configurations.

The efforts to estimate the indicial functions for aircraft configurations can be classified into two groups: the direct and the indirect methods. Leishman[12] has presented an indirect technique for identifying indicial functions from aerodynamic responses due to harmonic motions. However, the derived indicial functions using indirect methods depend largely on the quality of motion, e.g., amplitude, Mach number, and frequency. Experimental tests are limited for high frequencies and Mach numbers and practically nonexistent for direct indicial function measurements. An alternative is to use CFD, but special considerations are required to simulate step-responses in CFD. Singh and Baeder[13] used a surface transpiration approach to directly calculate the indicial response due to angle-of-attack using CFD. Ghoreyshi, et al.[14] also proposed an approach based on grid-motion for CFD-type calculation of inidical functions. In this paper, the indicial functions of aircraft are calculated using the CFD and grid-motion approach.

For motions at low angles-of-attack and assuming incompressible flow, only a single indicial function with respect to each forcing parameter needs to be generated.[15] For compressible and high angle-of-attack flows, many indicial functions need to be generated for different Mach numbers and angles-of-attack. The generation of all these functions using CFD is expensive and makes the creation of ROMs time-consuming. Note that these models are still much cheaper than a brute force approach because the ROMs based on indicial functions eliminate the need to repeat calculations for each frequency. A cost-effective, unsteady aerodynamic model needs a mathematical description of indicial functions as a function of angle-of-attack and Mach number. However, this model is often unavailable for three-dimensional configurations. It is more common to use surrogate models which are mathematical approximations of the true response of the system, built using some observed responses. By building surrogate models using a few observed responses, the total cost of modeling is reduced. In this paper, a surrogate model is used based on the Kriging technique to model indicial functions as a function of angle-of-attack and Mach number.

In this paper, the creation of reduced-order unsteady aerodynamic models using indical functions is reviewed. Next, the flow solver and an approach for CFD calculation of indicial functions are described. A surrogate model, built using some observed responses, is then described to approximate indicial function at new Mach numbers and angles-of-attack. The created ROM and the surrogate model are then used for aerodynamic predictions of a generic fighter configuration. The aircraft geometry and validation of CFD predictions are presented. Finally, the validity of ROMs is assessed by comparison of the model output with time-accurate CFD simulations.

2. Formulation

2.1 Reduced-Order Aerodynamics Modeling

The problem of predicting unsteady lift and pitch moment responses of a generic fighter to pitching and plunging motions is considered. Assuming these motions could be started from different Mach numbers, the Mach number is held constant during each motion, i.e., $\dot{V}=0$. The unsteady and nonlinear aerodynamic models used in this work are based on aerodynamic indicial functions by using superposition integrals. Tobak, et al.[16,17] and Reisenthel, et al.[18,19] detailed the superposition process for the modeling of unsteady lift and pitch moment from angle-of-attack and pitch rate indicial functions. Following their work, the time responses in lift due to the step-changes in angle-of-attack, α, and pitch angular velocity, q, are denoted as $C_{L\alpha}$ and C_{Lq}, respectively. A more general approach is used in this paper, such that these responses depend on angle-of-attack and Mach number. The unsteady-lift coefficient at time t is then obtained as[16]:

$$C_L(\mathrm{t}) = C_{L0} + \frac{d}{dt}\left[\int_0^t C_{L\alpha}(t-\tau,\alpha,M)\,\alpha(\tau)\,d\tau\right] + \frac{d}{dt}\left[\int_0^t C_{Lq}(t-\tau,\alpha,M)\,q(\tau)\,d\tau\right] \tag{1}$$

where, C_{L0} denotes the zero angle-of-attack lift coefficient and is found from static calculations. If $\dot{V} \neq 0$ an indicial response function with respect to the rate of change of velocity also needs to be included. Likewise the time-responses in pitch moment due to the step changes in angle-of-attack, α, and pitch angular velocity, q, is denoted as $C_{m\propto}$ and C_{mq} and then the pitch moment is estimated as follows:

$$C_m(\mathrm{t}) = C_{m0} + \frac{d}{dt}\left[\int_0^t C_{m\propto}(t-\tau,\alpha,M)\alpha(\tau)d\tau\right] + \frac{d}{dt}\left[\int_0^t C_{mq}(t-\tau,\alpha,M)q(\tau)d\tau\right] \tag{2}$$

Note that the unsteady effects in drag force are assumed to be small and, therefore, are not discussed here. The response functions due to pitch-rate, i.e., $C_{jq}(\alpha, M)$, for j=L, m can be estimated using a time-dependent interpolation scheme from the observed responses. These values are next used to estimate the second-terms in Equations 1–2, however the estimation of first-terms needs more explanation. Assuming a set of angle-of-attack samples of $\alpha=[\alpha_1,\alpha_2,\ldots,\alpha_n]$ at free-stream Mach numbers of $M=[M_1,M_2,\ldots,M_m]$, the aerodynamic responses to each angle of α_i, $i=1,\ldots n$ at Mach number of M_j, $j=1,\ldots,m$ is denoted as $H_{j\alpha}(t, \alpha_i, M_j)$ for j=L, m. In these response simulations, $\alpha(t)=0$ at $t=0$ and is held constant at α_i for all $t>0$. For a new angle of $\alpha^*>0$ at a new free-stream Mach number of M^*, the responses of $H_{j\alpha}(t, \alpha_k, M^*)$ are being interpolated at $\alpha_k=[\alpha_1, \alpha_2,\ldots, \alpha_s]$, such that $0<\alpha_1<\alpha_2<,\ldots\leq \alpha_s$ and $\alpha_s=\alpha^*$. These angles can have a uniform or non-uniform spacing. The indicial functions of $C_{j\alpha k}(t)$ for $k=1,\ldots,s$ at each interval of $[\alpha_{k-1}, \alpha_k]$ are defined as

$$C_{j\alpha 1}(t) = \frac{H_{j\alpha}(t,\alpha_1,M^*) - C_{j0}}{\alpha_1} \tag{3}$$

$$C_{j\alpha k}(t) = \frac{H_{j\alpha}(t,\alpha_k,M^*) - H_{j\alpha}(t,\alpha_{k-1},M^*)}{\alpha_k - \alpha_{k-1}} \qquad for\ k = 2,\ldots,s \tag{4}$$

where, C_{j0} denotes the zero angle-of-attack lift or pitch moment coefficient. The first integration operations in Equations 1–2 can now be estimated as

$$\begin{aligned}\int_0^t C_{j\propto}(t-\tau,\alpha^*,M^*)\alpha(\tau)d\tau = \int_0^t C_{j\propto 1}(t-\tau)\alpha_1 d\tau + \int_0^t C_{j\propto 2}(t-\tau)(\alpha_2-\alpha_1)d\tau + \cdots \\ + \int_0^t C_{j\propto s}(t-\tau)(\alpha_s-\alpha_{s-1})d\tau\end{aligned} \tag{5}$$

These steps can easily be followed for a negative angle-of-attack, i.e., $\alpha^*<0$. Once these models have been created, the aerodynamic response to a wide range of motions can be predicted on the order of a few seconds; however, the response functions still need to be determined. In this paper, the indicial functions of aircraft are calculated using the CFD and grid motion approach.

2.2 CFD Calculation of Indicial Functions

The flow solver used for this study is the commercially-available flow solver Cobalt[20] that solves the unsteady, three-dimensional and compressible Navier-Stokes equations in an inertial reference frame. Cobalt uses an arbitrary Lagrangian-Eulerian formulation and, hence, allows all translational and rotational degrees-of-freedom. The code can simulate both free and specified six degrees-of-freedom (6DOF) motions. The rigid-motion is specified from a motion input file that has the locations of a reference point on the aircraft at each time-step. In addition, the rotation of the aircraft about this reference point is also defined using the rotation angles of yaw, pitch, and bank. The aircraft reference point velocity, $v_\alpha(t)$, in an inertial frame is first calculated from instantaneous values of angles-of-attack, side-slip, Euler angles, and forward speed at each time. The initial aircraft velocity, v_0, is also specified in terms of Mach number, angle-of-attack, and side-slip angle from the steady-state $v_\alpha(t)-v_0$. This approach is used to calculate the indicial responses. As an example for a step change in angle-of-attack, the grid immediately starts to move at $t=0$ to the right and downward as shown in Figure 1. The translation continues over time with a constant velocity vector.[14] Since there is no rotation, all the effects in aerodynamic loads are from changes in the angle-of-attack. For a unit step change in pitch rate, the grid moves and rotates simultaneously.[14] The grid starts to rotate with a unit pitch-rate at $t=0^+$. To hold the angle-of-attack zero during the rotation, the grid moves right and upward, shown in Figure 1.

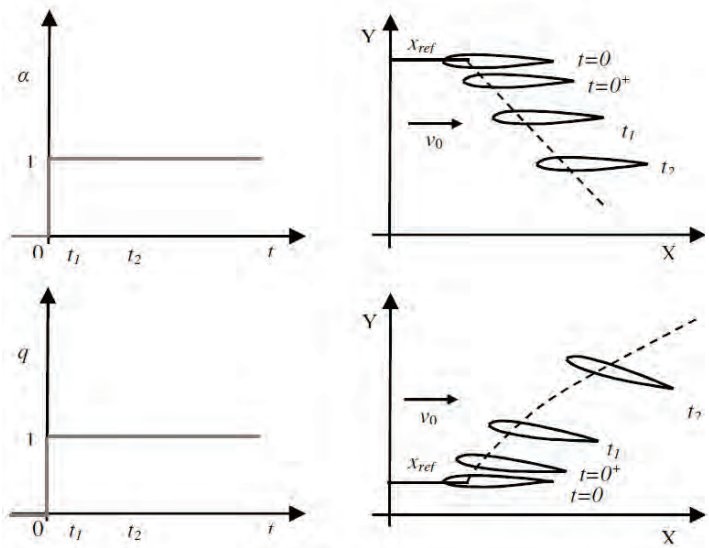


Figure 1. Grid motion for modeling a step change in angle-of-attack and pitch-rate[14]

2.3 Surrogate-Based Modeling of Indicial Functions

Having a ROM to predict aerodynamic responses to any arbitrary motion over a wide flight regime could become a very expensive approach because a large number of indicial functions need to be computed. In order to achieve a reasonable computational cost, a special time-dependent surrogate-based modeling approach proposed by Ghoreyshi, et al.[21] is used to predict indicial responses for a new point from some available (observed) responses. In this approach, the observed responses are viewed as a set of time-correlated spatial processes where the output is considered a time-dependent function. Ghoreyshi, et al.[21] have tested this approach for reduced order modeling of nonlinear and unsteady aerodynamic loads of a UCAV configuration. Assume an input vector of $\vec{x} = (x_1, \ldots, x_n)$, where n represents the dimensionality of the design space. To construct a surrogate model for fitting the input-output relationships, the unsteady aerodynamic responses corresponding to a limited number of input parameters (training parameters or samples) need to be generated. Design of Experiment (DoE) methods,[21] for example, can be used to select m samples. From time-marching CFD solutions of these samples, we obtain the data set represented in Equations 6–7.

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{6}$$

$$Z_D = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & \cdots & z_{mp} \end{bmatrix} \tag{7}$$

where matrix $D(m \times n)$ is the input matrix, whose rows correspond to different combinations of the design parameters. For each row in the input matrix, a time-dependent response was calculated at p discrete values of time, and this information is summarized in the output matrix of $Z_D(m \times p)$. The objective of surrogate modeling is to develop a model that allows predicting the aerodynamic response of $\vec{Z} = (\vec{x}_0) = (z_{01}, \ldots, z_{0p})$ at a new combination of design parameters $\vec{x}_0$.

3. Test Case

The Standard Dynamics Model (SDM) is a generic fighter configuration based on the F-16 planform. The model includes a slender strake-delta wing, horizontal and vertical stabilizers, ventral fin, and a closed inlet section. The three-view drawing is shown in Figure 2(a). This geometry has been tested extensively at various wind tunnel facilities to collect wind tunnel data.[23–25] Note that slightly different geometries were used in previous studies.

The lifting surfaces (strake, wing, and tail) have a thin airfoil with sharp leading edges. This enforces a fixed separation point on the leading edge and the formation of vortices over the surface. The vortex-induced suction pressure accounts for the additional lift, named vortex lift, which aids in the delay of stall. The complex interaction of strake and wing vortices creates very nonlinear aerodynamic characteristics. Also, at high angles-of-attack, these vortices break down, resulting in a sudden decrease in lift. The forward movement of vortex breakdown has significant effects on the pitch moment. Validation of CFD codes for predicting these effects can be a very challenging task.

In the current paper, only unsteady Reynolds-average Navier-Stokes (URANS) calculations are used. The full-span geometry mesh is 0.0529 scale of full size geometry and is shown in Figure 2(b). The pitch axis and moment reference point are located at 35% of mean aerodynamic center (MAC). The mesh was generated in two steps. In the first step, the inviscid tetrahedral mesh was generated using the ICEMCFD code. This mesh was then used as a background mesh by TRITET[26] which builds prism layers using a frontal technique. TRITET rebuilds the inviscid mesh while respecting the size of the original inviscid mesh from ICEMCFD. The full-span geometry mesh consists of a 9 million point mesh and 19.5 million cells. The grid resolution was chosen based on previous experience.[7]

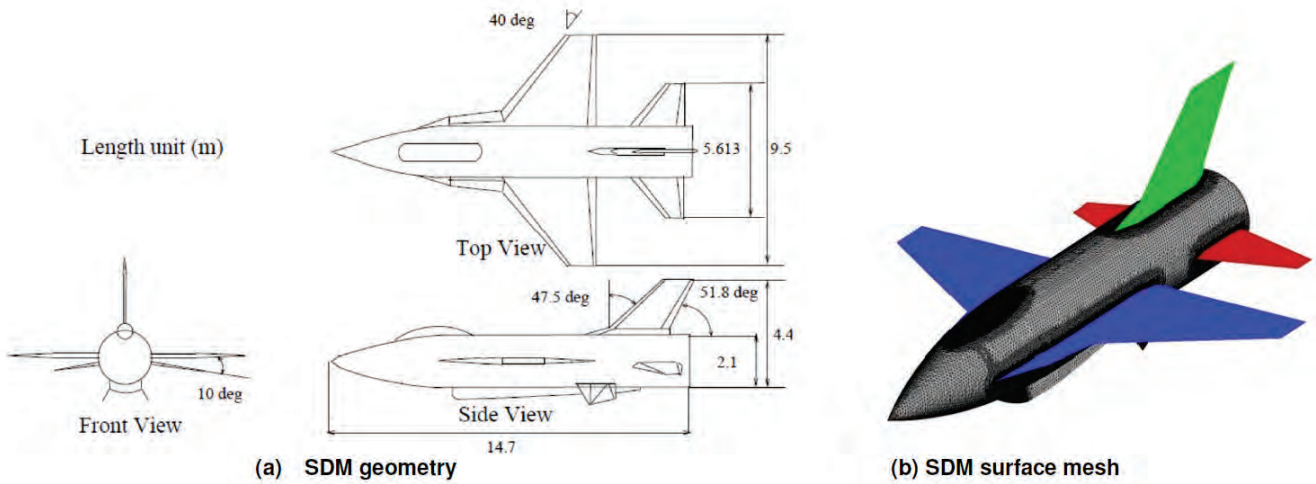
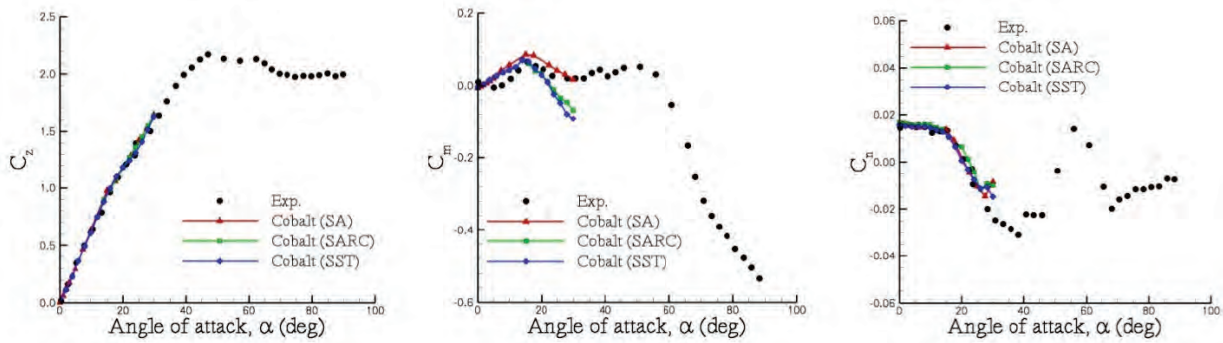


Figure 2. SDM geometry and surface mesh

4. Results

4.1 Static Predictions

The wind tunnel experiments were first used to validate the CFD predictions at low-speed. The conditions of the tests were V=110 m/s, Re=0.57 million and β=5° for α=0-90°. All CFD simulations were run at free-stream conditions consistent with flow conditions in the wind tunnel tests. For flow solution, RANS equations are discretized by second-order spatial and temporal operators. The turbulence models used are: Spalart-Allmaras (SA),[27] SA with Rotation/curvature Correction (SARC),[28] and Menter's Shear Stress Transport (SST).[29] All simulations were computed on an unstructured mesh with prisms in the boundary-layer and tetrahedral elsewhere on the full-span geometry. The cases were run on the Cray XE6 and Cray XE6 (open system) machines at the US Army Engineering Research Development Center (ERDC) [Garnet with 2.7GHz core speed and Chugach with 2.3GHz core speed] which have approximately 20,000 and 11,000 cores, respectively. The total run times of 1,000 iterations using 128 processors for the SARC and SST turbulence models were 5 and 6 hours, respectively. The static force and moment coefficients are compared with experimental data in Figure 3. The comparisons show that there is a good agreement between RANS predictions and the measurements for angles-of-attack below 25°. However, all turbulence models predict a positive pitch moment slope at zero degrees, while experiments show a falling trend at this angle. This is likely due to different inlet geometries in wind-tunnel and the used SDM geometry. At five degrees angle-of-attack, a pair of vortices emanating from the strake and wing leading-edge have formed as shown in Figure 4(a). These vortices do not exhibit breakdown and do not interact over the airframe. The vortex formation causes an additional increase of the normal force and pitch moment coefficients. In experimental tests, the sign of the pitch moment is reversed at this angle, while CFD shows a jump in the moment rate of increase. No significant changes in lateral force and moment coefficients were observed at this angle.



(a) Normal force coefficient (b) pitch moment coefficient (c) yaw moment coefficient

Figure 3. Static aerodynamic predictions; the moment reference point is located at 35% MAC

The vortices grow in size and strength with increasing angle-of-attack. At ten degrees, the center of the wing vortex is shifted laterally, while the shedding point is moved forward as shown in Figure 4(b). There are still no signs of vortex breakdown. The yaw moment slightly changes due to the wing vortex movement. Around 14 degrees, the two vortices wind around each other towards the trailing-edge of the wing as shown in Figure 4(c). With a slight increase in angle-of-attack, the wing vortex appears to breakdown quickly as shown in Figure 4(d). The vortex breakdown leads to a smaller normal force rate of increase and a negative pitch moment slope. The vortex breakdown phenomenon is asymmetric, and hence the yaw moment suddenly starts to change very fast. At 22 degrees, the strake vortex is also burst as shown in Figure 4(e). Finally, at 25 degrees there is no sign of a wing vortex as shown in Figure 4(f).

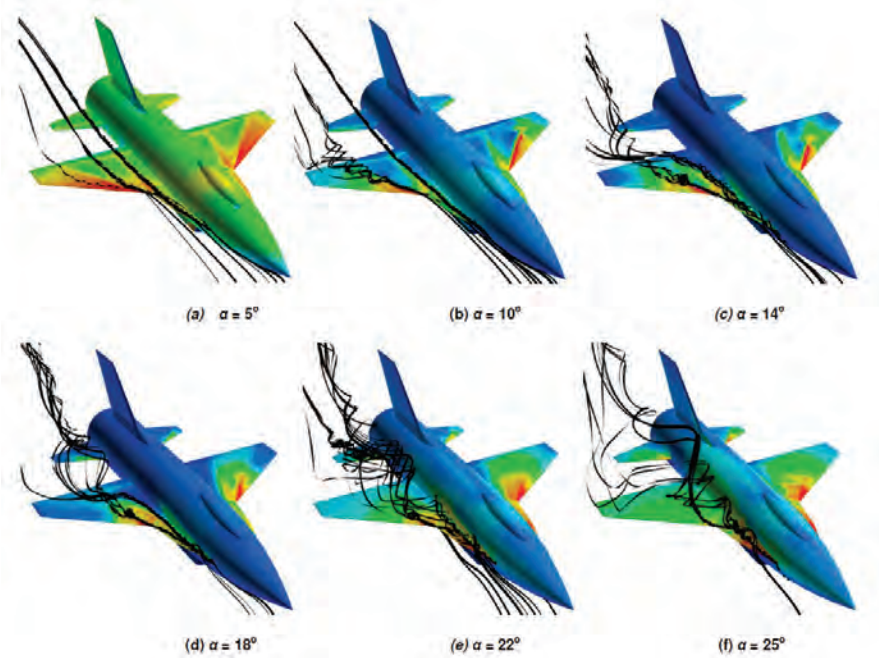


Figure 4. SDM flow-field visualization

4.2 Calculation of Indicial Functions

The indicial response functions in this paper are interpolated from some available samples in the angle-of-attack and free-stream Mach number space. Note that these functions only need to have dependency on angle-of-attack and Mach number, and once they have been calculated they could be used to predict the aerodynamic response to any frequency of interest. The samples could be generated using methods of factorial designs, Latin hypercube sampling, low-discrepancy sequences, and designs based on statistical optimality criteria (A-, D-, and G-optimal designs).[30] Factorial designs are extremely simple to construct and have been used in this work.

The considered SDM motions encompass α and *M* values in the range of [-10°, 10°] and [0.3, 0.7], respectively. Assuming symmetrical flow solutions with respect to the angle-of-attack, the indicial functions are only calculated for positive angles-of-attack. A set of samples including 50 points is defined on the α and *M* space using factorial design;[30] these points are shown in Figure 5. The indicial functions are calculated using the CFD and grid motion approach for each sample conditions. All these calculations started from a steady-state solution such that the Mach number in the steady-state simulations corresponds to each sample Mach number. For the indicial functions due to angle-of-attack, the steady-state angle-of-attack is set to zero degrees, but for the indicial functions due to pitch-rate, the steady angle-of-attack corresponds to each sample α. The step function calculations are second-order in time with a non-dimensional time-step of $\Delta t^* = \Delta t.\frac{V}{c} = 0.01$. For more details on time-step selection, the reader is referred to the work of Cummings, et al.[31]

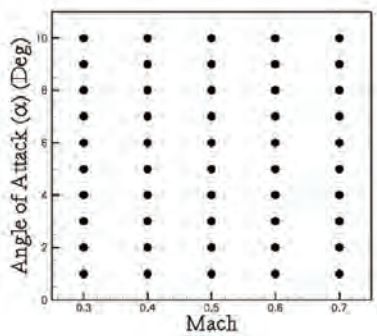


Figure 5. Design sample

The calculated indicial functions due to angle-of-attack are shown in Figure 6 for M=0.3 and M=0.6. Figures 6(a) and 6(b) show that the indicial lift has a peak at *s=0* followed by a rapidly falling trend. The lift again builds up and asymptotically reaches the steady-state value. The pitch moment has also a negative peak at *s=0* as shown in Figures 6(c) and 6(d). The initial peak can be explained based on the energy of acoustic wave systems created by the initial perturbation.[32] The most obvious difference between responses at low- and high-speed is that the initial peak becomes smaller for higher speeds. An explanation is given by Leishman[12]: this is due to the propagation of pressure disturbances at the speed of sound, compared with the incompressible case, where the disturbances propagate at infinite speed. Figure 6 also shows that the initial values of indicial functions are invariant with angle-of-attack, but the intermediate trend and steady-state values change, depending on the angle-of-attack. Although, the final values of indicial lift are nearly unchanged for angles-of-attack below five degrees, but the pitch moment final values are different even at small angles-of-attack due to vortices on the wing.

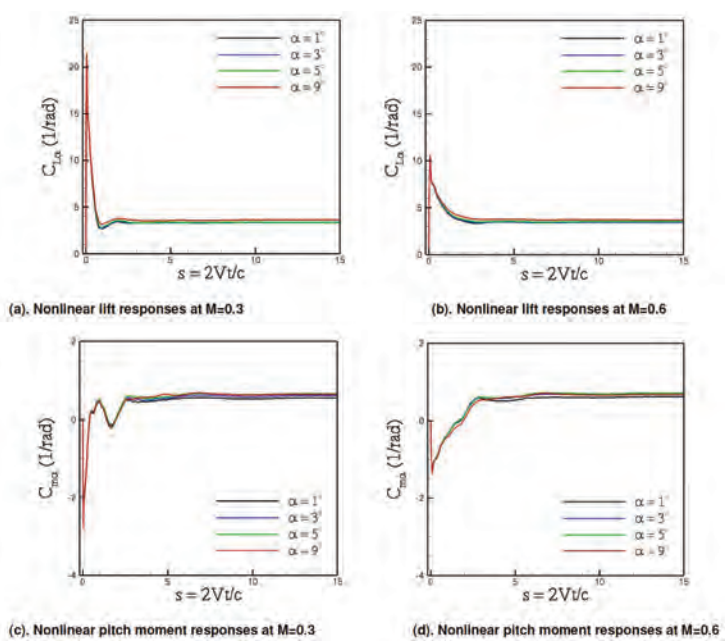
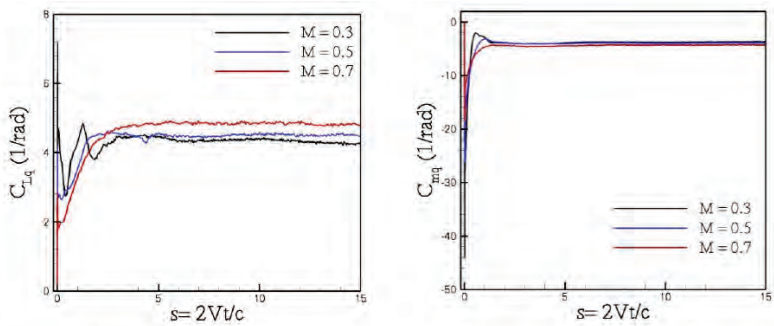


Figure 6. Nonlinear lift and pitch moment indicial solutions due to angle-of-attack for M=0.3 and 0.6

The effects of Mach number on the lift and pitch moment indicial functions are shown in Figure 7. Likewise angle-of-attack response, the lift and pitch moment have initial peaks due to the initial perturbations and asymptotically reach the steady-state value. Figure 7 shows that increasing Mach number results in the increase of steady-state C_{Lq} and the decrease of steady-state C_{mq} (the so-called pitch damping derivative) as shown in Figure 8. Figure 8 compares the steady-state C_{mq} values calculated from the CFD code with the out-of-phase components of pitch moment derivative, i.e., $C_{mq} + C_{m\dot{\alpha}}$ measured at different Mach numbers and zero degrees angle-of-attack; these experimental data are detailed by Da Ronch, et al.[33] Like the static predictions, the CFD values slightly underestimate the experimental pitch moment data, although the CFD data do not include the effects of rate of change of angle-of-attack, i.e., $C_{m\dot{\alpha}}$. Typically, C_{mq} is the largest factor in the sum $C_{mq} + C_{m\dot{\alpha}}$, typically accounting for 90% of the sum. Again, the underestimation of experimental data is likely due to different inlet geometries in wind-tunnel and the used SDM geometry. Note that the indicial function approach allows the direct calculation of pitch damping derivatives but extraction of dynamic derivatives from harmonic motions results in the in-phase and out-of-phase components[33] and additional work required to separate each derivative from these components.



(a). Pitch rate lift indicial functions (b). Pitch rate pitch moment indicial functions

Figure 7. Lift and pitch moment indicial solutions due to pitch-rate for α=0°. The pitch axis and moment reference point are located at 35% MAC.

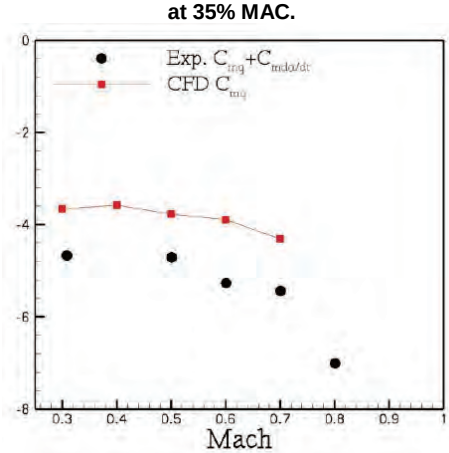
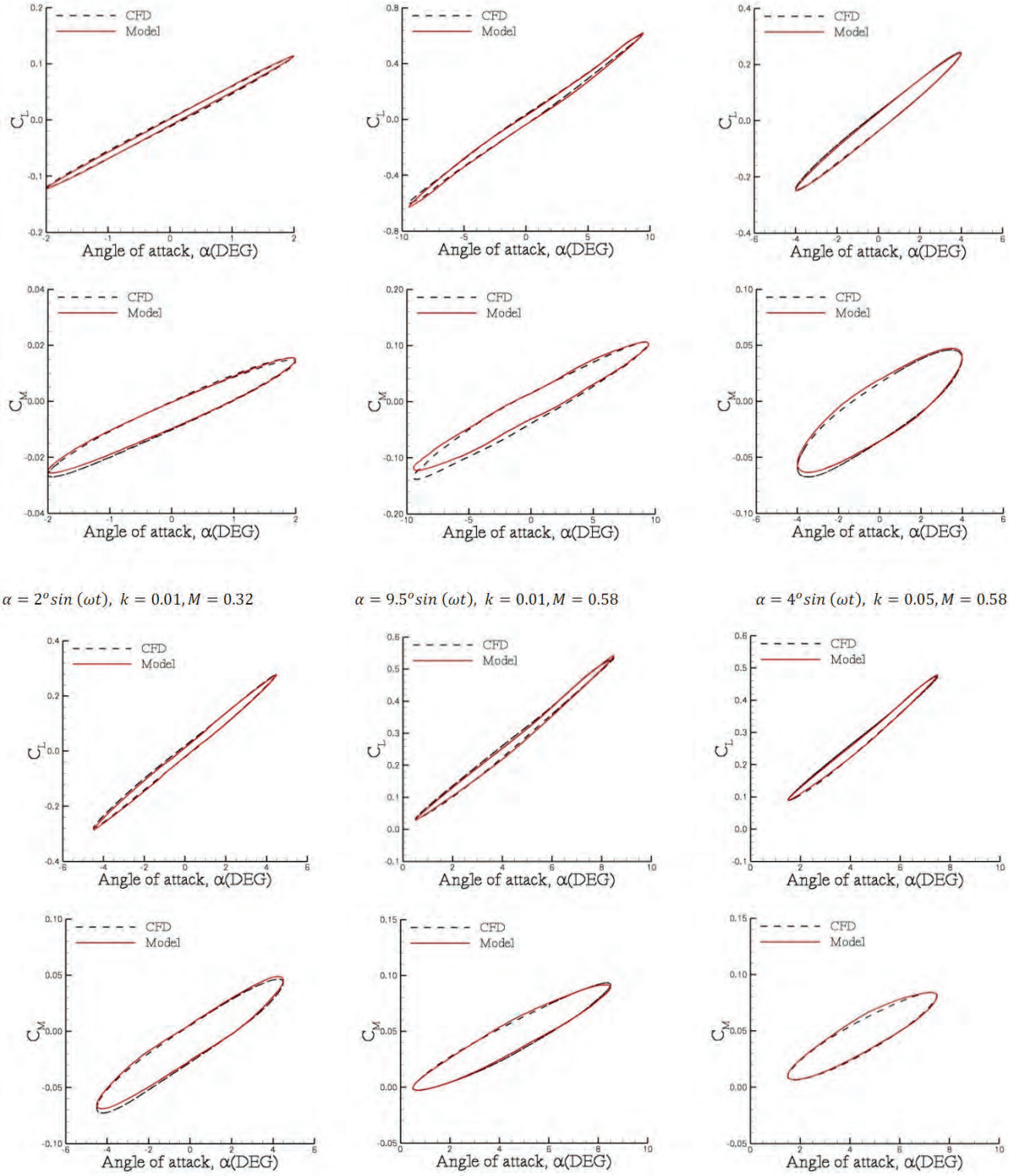


Figure 8. Validation of C_{mq} values calculated from pitch-rate indicial functions. Experimental data are from Da Ronch, et al.[33]

4.3 Reduced-Order Modeling of Dynamic Motions

A ROM is now created along with a time-dependent surrogate model to determine the terms in Equations 1–2 at each time-step. The validity of ROMs are tested for several motions and compared with time-accurate CFD simulations in Figure 9. These motions start from different steady-state conditions (not being used at sample design) and run for different amplitudes and frequencies. The time-accurate solutions are labeled as "CFD" in the plots. Figure 9 shows that the ROM lift and pitch moment predictions agree well with the full-order CFD simulation values. Small discrepancies are found in the pitch moment predictions at negative angles-of-attack. This is likely due to the fact that SDM pitch moment is not symmetric with angle-of-attack and, hence, the response functions generated at positive angles cannot predict the slope changes correctly. Note that the average cost of generating each full-order simulation is around 1,280 CPU hours, while the ROM predictions require on the order of a few seconds.



$\alpha = 2^o sin\,(\omega t),\ k = 0.01, M = 0.32$ $\alpha = 9.5^o sin\,(\omega t),\ k = 0.01, M = 0.58$ $\alpha = 4^o sin\,(\omega t),\ k = 0.05, M = 0.58$

$\alpha = 4.5^o sin\,(\omega t),\ k = 0.015, M = 0.66$ $\alpha = 4.5^o + 4.5^o sin\,(\omega t), k = 0.02, M = 0.37$ $\alpha = 4.5^o + 3.5^o sin\,(\omega t), k = 0.03, M = 0.58$

Figure 9. ROM prediction of lift and pitch-moment responses to arbitrary pitching motions. In above, *ω* denotes angular velocity and *k=ωc/2V* is reduced frequency.

5. Conclusion

The use of indicial function for unsteady and nonlinear aerodynamics modeling of a generic fighter configuration is investigated in this work. The longitudinal aerodynamic forces and moments are only considered; thus, the aircraft responses corresponding to a step change in angle-of-attack and pitch-rate are found. The step functions were calculated using a CFD and a grid motion approach for a set of samples defined in angle-of-attack and Mach number space. The results show that indicial functions have a peak response at initial time-steps. This is related to a traveling acoustic wave formed by the flow disturbance. At higher Mach numbers, the peak values are diminished due to compressibility. A time-dependent surrogate model was used to interpolate these functions for the new conditions. The ROMs are tested by comparison of the model output with time-accurate CFD simulations for several motions. The results show that predictions agree well with full-order CFD simulation values. Future work will extend this study to include samples generated by Latin hypercube sampling and generate ROMs for six-degree-of-freedom maneuver.

Acknowledgments

The authors appreciate the support provided by the Modeling and Simulation Research Center at the US Air Force Academy. Mehdi Ghoreyshi also would like to thank the financial support by the National Research Council/US Air Force Office of Scientific Research. The computational mesh was generated by Adam Jirasek. Appreciation is expressed to the DoD High Performance Computing for providing computer time.

References

1. Pamadi, B.N., *Performance, Stability, Dynamics, and Control of Airplanes*, AIAA Education Series, Reston, Virginia, 2004.

2. Wright, J.R. and Cooper, J E., *Introduction to Aircraft Aeroelasticity and Loads*, John Wiley, Chichester, UK, 2008.

3. Chambers, J.R. and Hall, R.M., "Historical Review of Uncommanded Lateral-Directional Motions at Transonic Conditions."*Journal of Aircraft*, Vol. 4, No. 3, pp. 436–447, 2004.

4. Hall, R.M., Woodson, S.H., and Chambers, J.R., "Accomplishments of the Abrupt-Wing-Stall Program." *Journal of Aircraft*, Vol. 42, No. 3, pp. 653–660, 2005.

5. Hall, R.M., "Overview of the Abrupt Wing Stall Program." *Progress in Aerospace Sciences*, Vol. 40, No. 7, pp. 417–452, 2004.

6. Silva, W.A., "Application of Nonlinear Systems Theory to Transonic Unsteady Aerodynamics Responses." *Journal of Aircraft*, Vol. 30, No. 5, pp. 660–668, 1993.

7. Ghoreyshi, M., Jirasek, A., and Cummings, R.M., "CFD Modleing for Trajectory Predictions of a Generic Fighter Configuration." *AIAA Paper 2011-6523*, August 2011.

8. Leishman, J.G. and Nguyen, K.Q., "State-Space Representation of Unsteady Airfoil Behavior." *AIAA Journal*, Vol. 28, No. 5, pp. 836–844, 1989.

9. Librescu, L. and Song, O., *Thin-walled Composite Beams: Theory and Application*, Springer, Dordrecht, Netherland, 2006.

10. Wagner, H., "Uber die entstehung des dynamischen auftriebes von trag." *Zeitschrift fur angewandte Mathematik und Mechanik*, Vol. 1, pp. 17–35, 1925.

11. Bisplinghoff, R.L., Ashley, H., and Halfman, R.L, *Aeroelasticity*, Dover Publications, Mineola, NY, 1955.

12. Leishman, J.G., "Indicial Lift Approximations for Two-Dimensional Subsonic Flow as Obtained from Oscillatory Measurements." *Journal of Aircraft*, Vol. 30, No. 3, pp. 340–351, 1993.

13. Singh, R. and Baeder, J., "Direct Calculation of Three-Dimensional Indicial Lift Response Using Computational Fluid Dynamics." *Journal of Aircraft*, Vol. 34, No. 4, pp. 465-471, 1997.

14. Ghoreyshi, M., Jirasek, A., and Cummings, R. M., "A Computational Investigation into the Use of Response Functions for Aerodynamic Loads Modeling." for *AIAA Journal*, 2012 (in press); also available as *AIAA Paper 2011-3518*.

15. Leishman, J.G. and Crouse, G.L., "State-Space Model for Unsteady Airfoil Behavior and Dynamic Stall." *AIAA Paper 89-1319*, April 1989.

16. Tobak, M., Chapman, G.T., and Schiff, L.B., "Mathematical Modeling of the Aerodynamic Characteristics in Flight Dynamics." *NASA-TM-85880*, 1984.

17. Tobak, M. and Chapman, G.T., "Nonlinear Problems in Flight Dynamics Involving Aaerodynamic Bifurcations." *NASA-TM-86706*, 1985.

18. Reisenthel, P.H., "Development of a Nonlinear Indicial Model Using Response Functions Generated by a Neural Network." *AIAA Paper 97-0337*, January 1997.

19. Reisenthel, P.H. and Bettencourt, M.T., "Data-based Aerodynamic Modeling Using Nonlinear Indicial Theory." *AIAA Paper 99-0763*, January 1999.

20. Strang, W.Z., Tomaro, R.F., and Grismer, M.J., "The Defining Methods of Cobalt: A Parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver." *AIAA Paper 99-0786*, January 1999.

21. Ghoreyshi, M. and Cummings, R.M., "Unsteady Aerodynamics Modeling for Aircraft Maneuvers: A New Approach Using Time-Dependent Surrogate Modeling." *AIAA Paper 2012-3327*, June 2012.

22. Simpson, T.W., Poplinski, D., Koch, P.N., and Allen, J.K. "Metamodels for Computer-Based Engineering Design: Survey and Recommendations." *Engineering with Computers*, Vol. 17, No. 2, pp. 129–150, 2001.

23. Beyers, M.E., "Subsonic Roll Oscillation Experiments on the Standard Dynamics Model." *AIAA Paper 83-2134*, August 1983.

24. Jermey, C. and Schiff, L.B., "Wind Tunnel Investigation of the Aerodynamic Characterestics of the Standard Dynamic Model in Conning Motion at Mach 0.6." *AIAA Paper 85-1828*, August 1985.

25. Balakrishna, S. and Niranjana, T., "Wind Tunnel Dynamic Flying Study of the Pitching Moment Derivatives of the Standard Dynamics Model in Active Control." *AIAA Paper 87-2626*, August 1987.

26. Tyssel, L., "Hybrid Grid Generation for complex 3D Geometries." *Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulation*, pp. 337–346, 2000.

27. Spalart, P.A. and Allmaras, S.R., "A One-Equation Turbulence Model for Aerodynamic Flows." *AIAA Paper 92-0439*, January 1992.

28. Spalart, P.A. and Schur, M., "On the Sensitisation of Turbulence Models to Rotation and Curvature." *Aerospace Science and Technology*, Vol. 1, pp. 297–302, 1997.

29. Menter, F., "Two-Equation Eddy-Viscosity Turbulence Model for Engineering Applications." *AIAA Jounral*, Vol. 32, No. 8, pp.1598–1605, 1994.

30. Mackman, T.J., Allen, C.B., Ghoreyshi, M., and Badcock, K.J., "Comparison of Adaptive Sampling Methods for Generation of Aerodynamic Models for Flight Simulation." *AIAA Paper 2011-1171*, January 2011.

31. Cummings, R.M., Morton, S.A., and McDaniel, D.R., "Experiences in Accurately Predicting Time-Dependent Flows." *Progress in Aerospace Sciences*, Vol. 44, pp. 241–257, 2008.

32. Ghoreyshi, M. and Cummings, R.M., "Challenges in the Aerodynamics Modeling of an Oscillating and Translating Airfoil at Large Incidence Angles." *AIAA Paper 2012-0690*, January 2012.

33. Da Ronch, A., Vallespin, D., Ghoreyshi, M., and Badcock, K.J., "Evaluation of Dynamic Derivatives Using Computational Fluid Dynamics." *AIAA Journal*, Vol. 50, No. 2, pp.470–484, 2012.

Detecting Laminar-to-Turbulent Transition using TDLAS EXPLORER

Robert J. Yentsch, Nathan J. Mullenix, and Datta V. Gaitonde
Mechanical and Aerospace Engineering, The Ohio State University, Columbus, OH
{yentsch.1, mullenix.10, gaitonde.3}@osu.edu

Michael S. Brown
US Air Force Research Laboratory, Advanced Propulsion Division (AFRL/RZAS), Wright-Patterson AFB, OH
michael.brown@wpafb.af.mil

Abstract

Boundary-layer transition is an important parameter for flight vehicles, and the use of non-invasive optical measurement techniques to detect transition is still in its infancy. This paper simulates detection of laminar-to-turbulent transition in a flow-field using TDLAS (Tunable Diode Laser Absorption Spectroscopy). TDLAS EXPLORER, a software suite developed for use on the Department of Defense High Performance Computing Modernization Program systems in support of AFRL's (US Air Force Research Laboratory's) non-invasive in-stream optical measurements programs, is used to post-process a large, transient boundary-layer computational fluid dynamics (CFD) dataset generated with ILES (implicit large-eddy simulation). Several technical and theoretical challenges must be addressed. The size of the CFD dataset is extremely large (several thousand time-steps on a mesh having tens of millions of points), and the effects of scaling of the Reynolds number must be evaluated. Initial results indicate that laminar to turbulent boundary-layer transition is detectable through TDLAS, and that significant performance improvements in the evaluation may be had through tighter integration with the CFD code during run-time.

1. Introduction

Non-invasive measurements have been an ongoing research topic for many years, as surface measurement systems can only measure flow properties through the filter of the boundary-layer, and physically-invasive measurement systems disturb the flow they are attempting to measure. Particularly, in high-speed flows and inside of combustion systems, survivability alone can make invasive measurement systems unacceptable. Tunable Diode Laser Absorption Spectroscopy (TDLAS) addresses both of these issues; however, the full range of phenomena that can be measured with this promising technique is an active research area. Boundary-layer transition itself is an important topic, particularly in high-speed flows, where changes in drag or heat transfer can drastically affect aerodynamics or survivability. It is also hoped that TDLAS systems can be used for optical mass-capture measurements (Barhorst, et al., 2009), and some initial flight tests have been conducted to investigate suitability for this (Sappey, et al., 2009) and other applications (Lindstrom, et al., 2009).

1.1 TDLAS Measurement

TDLAS measurement involves sweeping a tunable diode lasers emission spectrum (a by-product of changing the current) through a frequency range where one or more characteristic absorption lines of a particular target molecule (i.e., H_2O or O_2) exist. After passing through the media, the laser beam strikes a photodiode. Careful post-processing can exploit optical-frequency and mass-fraction-dependent absorption characteristics of the flow to determine various state parameters such as temperature, pressure, velocity and target species mole fraction at a single axial location in the flow-path; although use of multiple lines-of-sight does permit two-dimensional (2D) optical tomography of the state variable fields. One difficulty of utilizing this measurement technique is that the absorptivity of the media is at each point a non-linear function of thermodynamic state, molecular composition, and directional velocity. Furthermore, the degradation of the signal exhibits hysteresis. This leads to a complicated and computationally-expensive post-processing stage, as well as introducing uncertainty and indeterminability.

For this tool to be useful as part of the control system in a flight vehicle, there must be *a-priori* knowledge of expected readings, thus TDLAS EXPLORER was developed to allow the designer or researcher to test the theoretical response of an arbitrary TDLAS system, so that better informed decisions can be made about the position, processing, and usefulness of the system very early in the design process. The tool can use existing computational fluid dynamics (CFD) solutions as flow-fields, and enables parametric investigation of placement parameters such as number and location of optical sight lines, photo-detector response, and data acquisition speeds.

Figure 1 shows an example of a previous application of TDLAS EXPLORER to post-processing of simulations of the HIFiRE Flight 1 (Yentsch and Gaitonde, 2011). As can be seen from the intensity curves, hysteresis effects are demonstrated in that the symmetric CFD solution gives different results where both lasers are directionally-oriented right-to-left in the figure. This mirrors the orientation of the system in the actual flight test, where that hysteresis effect, i.e., the change in intensity at any point is a function of the input intensity, is being explored for use in optical mass capture measurement.

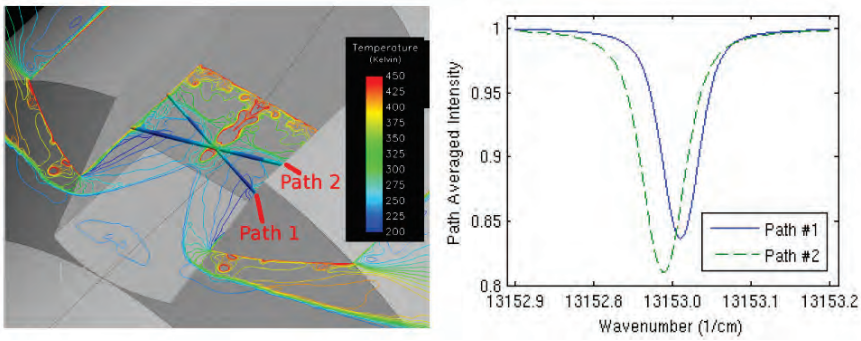


Figure 1. Example application of TDLAS EXPLORER to CFD simulations of HIFiRE Flight 1

1.2 Generation of an Equilibrium Turbulent Boundary Layer

To investigate a TDLAS system's ability to identify and resolve laminar-to-turbulent transition in a boundary-layer, we use an existing high-fidelity, time-accurate ILES (implicit large-eddy simulation) simulation of turbulent transition in supersonic wall-bounded flows, performed at the US Air Force Research Laboratory (AFRL) and the US Engineer Research and Development Center, Department of Defense (DoD) Supercomputing Resource Center (ERDC DSRC), under an US Air Force Office of Scientific Research (AFOSR)-sponsored task (Mullenix, et al., 2012). The CFD dataset is used as an input to produce a realistic estimate of optical system response and resolution.

The turbulent boundary-layer flow-field was initially created to simulate the interaction of localized-arc filament plasma actuators (LAFPAs) with the complex flow-field associated with a STBLI (shock-train boundary-layer interaction) in conjunction with experiments being performed at the OSU Gas Dynamics and Turbulence Lab (GDTL). Earlier works (Mullenix, et al., 2012; Mullenix, et al., 2011) have demonstrated the spatial development of an equilibrium turbulent boundary-layer shown in Figure 2 resulting from the tripping of a laminar boundary-layer using a steady counter-flow forcing modeled on a dielectric barrier discharge (DBD) plasma actuator. It was shown that the obtained boundary-layer satisfies mean and statistical properties expected from the literature, and that there is no detectable artificial length or time-scales imparted by this method. Initial calculations were conducted on an expensive (65 million) point grid, but further work has yielded similar results on coarser grids of 35 and 20 million points. The dataset used here was computed on the 20 million point grid, but has been output on an isotropically-sequenced 2.5 million point grid which still retains more than sufficient resolution for this purpose.

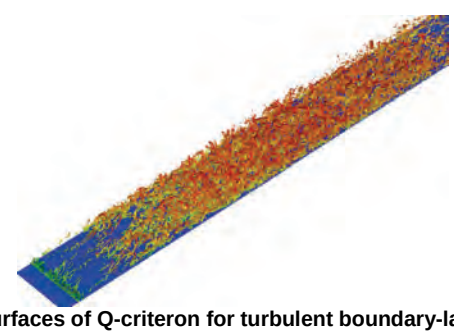


Figure 2. Iso-surfaces of Q-criteron for turbulent boundary-layer simulations

2. Numerical Issues and Methodology

2.1 Time-step Resolution and Domain Decomposition

The size of the computational domain required to generate the turbulent flow-field on the smaller 20 million point mesh, as previously described, necessitates parallelization of the computation, both from time and memory standpoints. 192 cores on 24 nodes of ERDC's Diamond supercomputer were used in the simulation of the boundary-layer. Although for the actual simulation of the TDLAS response presented later in this paper, the recombined full datasets were used, it is constructive to consider efficiency gains that may be had in future simulations by more tightly integrating TDLAS EXPLORER with the flow-field solver.

Currently, simulations in TDLAS EXPLORER are performed by interpolating the full flow-field output from the solver onto an automatically-generated laser grid (shown later) which simulates the path of the beam. This step is oftentimes more computationally-expensive than the actual TDLAS simulation itself, particularly in cases such as this, where the size of the donor grid is large and many points have to be searched during the interpolation process. In the case here, the interpolation process on six cores took approximately half an hour, whereas the actual TDLAS simulation on a single core took only 30 seconds. The size comparison between the flow-field and laser grids as shown in Figure 3 indicates that only a small portion of the flow-field grid overlaps the laser grid and, therefore, the computational expense of the interpolation process can be significantly decreased.

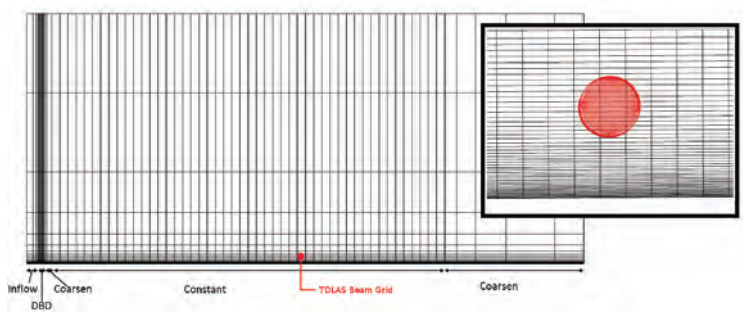


Figure 3. Representative size comparison between flow-field and laser beam grids (cells shown for clustering demonstration only) and insert showing realistic comparison

FDL3DI's output subsystem automatically recombines the decomposed grid for output during the course of the run. Accordingly, the simplest possible solution to decreasing the computation expense of the TDLAS simulation is to output only a small subset of grid points which are known *a priori* to overlap with the laser grid. This can then be post-processed in the typical fashion, albeit at a much faster rate due to the decreased time for file input/output (I/O) and flow-field interpolation. It would also allow the full-resolution grid to be output efficiently, rather than the scaled-grid used in these simulations.

More relevant in transient simulations is the time-dependent nature of the TDLAS system. As previously mentioned, TDLAS systems sweep through a frequency range over a finite period of time to measure absorption, as shown by a realistic signal in Figure 4. Note that the blue line represents the theoretical unabsorbed signal; the ramp in intensity is a by-product of the current ramp that causes the desired sweep in frequency. The amount of time required for one frequency sweep varies depending on the system; however, if the time-scales of the flow-field are on the order of the time-scale for the frequency sweep, certain phenomena could be completely missed by the detection system. In applications to control systems in particular, if a phenomenon displays periodic behavior having a frequency close to the sweep frequency of the measurement system, the signal related to those phenomena could be artificially amplified, giving false input to the control system.

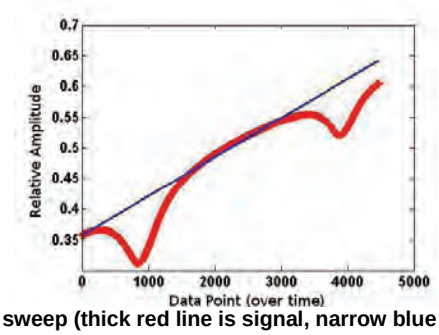


Figure 4. Realistic TDLAS dataset for one sweep (thick red line is signal, narrow blue line is curve fit to zero-absorptivity media)

To most accurately simulate the response of the system, it would be desirable to resolve an equal number of data points in a single sweep as does the data acquisition system of the TDLAS. Given that a representative system could take 1,000 data points in a single, millisecond long sweep that gives a sample period of 1.0×10^{-6} seconds. The time-step size of our boundary-layer simulations is 9.5×10^{-9} seconds with flow-field solutions written every 2,500 time-steps, or every 2.375×10^{-5} seconds. Therefore, approximately 24,000 data points would be sampled in a realistic TDLAS system between every available transient output file. Assuming that structures in the flow convect at the local velocity (approximately 445 m/s, or 80% of the free stream, at the center of the beam path), then during the interval between flow-field solutions, structures have convected nearly an order-of-magnitude more length than the diameter of the laser beam.

Because each sample point in the sweep represents a different frequency which may be attenuated to a different degree, the period between samples must be compared to the width of the attenuated region. For example, the attenuated peaks in oxygen-sensing TDLAS system may have half-widths of 10% of the frequency sweep range; whereas, the attenuated peaks in water-sensing TDLAS systems may have half-widths of only 1% of the frequency sweep range. The oxygen sensing TDLAS system being modeled for use with the boundary-layer simulations has a very wide half-width, and testing has shown that the absorptivity curve can be very well-reproduced with approximately 100 wave-number points per sweep, or 1×10^{-5} seconds between samples giving approximately three sample points between each time-step. To accommodate this, TDLAS EXPLORER interpolates between available time-steps for each sample point, but it is yet unknown exactly what effect this has on the accuracy of the simulation. Although it is clearly desirable to output the flow-field more often, it should be noted that outputting the entire flow-field at approximately the same time-rate as the desired sample-time would add an approximately 8% wall-clock overhead, a not insignificant amount of CPU-time on a simulation this size.

At this point, FDL3DI performs all recombination and file IO operations on CPU 0. This imparts a communication overhead, as well as a delay where the slave CPU's wait for CPU 0 to finish its operations before proceeding. One possible solution to increasing efficiency in both the output and interpolation steps involves having each node which has an overlap with the laser grid write out its decomposed domain itself. This would remove the communication overhead, as well as decrease the time the slave nodes spend waiting, as the collection of smaller concurrent writes to disk should take significantly less time than one large write. Attempts have been made in the past to perform parallel IO on all nodes in FDL3DI; however, portability and file system stability issues involving parallel IO to Lustre file systems have led to the more conservative approach currently used.

Instead of attempting to write the grid from each node, performing interpolation from the decomposed domains on each local CPU has been considered. This could be performed every time-step with a minimum of communication overhead on a very small decomposed domain which would greatly decrease the search time required for the interpolation, as well as spread that effort among the CPU's. Figure 5 illustrates the complexities posed by the potential for the laser grid to cross decomposed, overlapping domains (a nine-point overlap was used in the boundary-layer simulations), where the colored blocks represent the overlapping domains and the black circular block represents the laser-beam grid.

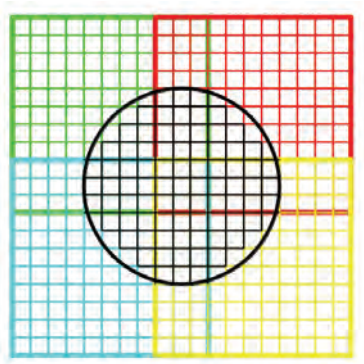


Figure 5. Representation of grid and domain decomposition overlap

2.2 Non-Dimensionalization and Reynolds Number Scaling Effects

TDLAS EXPLORER expects its' input to be dimensional; however, FDL3DI and its output are non-dimensionalized. Since limits of machine accuracy can cause unintended effects when numbers with high-magnitudes (e.g., pressure in Pascals) interact with numbers of low-magnitudes (e.g., density in kg/m^3), use of non-dimensional variables is relatively common in numerical simulation. Conversion of primitive variables back to dimensional numbers is fairly trivial, and a pre-processing module is available in TDLAS EXPLORER to perform this action.

A more difficult issue is scaling of the Reynolds number. In large-eddy simulation (LES), mesh resolution requirements scale to a high-power (~4) of the Reynolds number. The Reynolds number in the experiments would have required a factor of almost $(10^4)^3 \sim 10^7$ number of grid-points as was used in the simulation. In order to keep the length, viscosity, Mach number, and velocity scales the same, the pressure (and correspondingly, the density) was scaled to 10% of the experimental values while holding all other parameters constant as shown in Table 1.

Table 1. Reynolds number scaling and free-stream conditions for boundary-layer simulation

Property	Experimental	LES
M_∞	2.33	2.33
U_∞ *(m/s)*	556	556
P_∞ *(Pa)*	23511.12	2351.11
T_∞ *(K)*	141.7	141.7
T_{wall} *(K)*	269.75	269.75
Incoming BL Thickness	5.3×10^{-3}	5.3×10^{-3}
$Re_{\delta 99}$	**175202.06**	**17520.21**

For TDLAS modeling, the absorptivity of the media is highly-dependent on the thermodynamic properties. Changes in density most directly affect the overall absorptivity of the media through changes in the number density. This scaling of the overall absorptivity will cause a greater or lower magnitude integrated signal to be observed. This is an important issue given that one important question in this applications of TDLAS measurement systems is whether the fluctuations in the boundary-layer are of sufficient magnitude to be observable above the noise inherent in the measurement system.

The profile of the absorptivity as a function of wave-number is modeled through the Voigt function, a convolution of the Gaussian and Lorentzian functions. In general, relatively lower-pressures and higher-temperatures tend toward a predominately Gaussian shape, whereas higher-pressures and lower-temperatures tend toward a predominantly Lorentzian shape. Thus, scaling the pressure to a much lower value relative to the temperature would artificially shift the line-shape towards the Gaussian profile, which will cause a lower half-width-value.

The velocity of the flow can be calculated using the Doppler shift in the peak absorptivity relative to the expected value (a well known quantity). Fortunately, the scaling left the velocity the same as in the experiment, so any simulation results that use this quantity will be unaffected.

3. Results of Simulations

Simulations were performed on the available dataset consisting of full domain output files every 2,500 time-steps. Although it is not fully-known whether sufficient temporal resolution exists to sufficiently show the propagation of structures, nor has the number density scaling been corrected, as previously discussed, it should be sufficient to examine the effects of transition.

TDLAS EXPLORER is not explicitly parallel through use of Message Passing Interface (MPI) or OpenMP; however, it can take advantage of multi-core machines through the use of forking shell scripts to more rapidly perform the pre-and post-processing stages. This is a reasonable compromise for full-parallelization given that these stages (particularly the interpolation) often take more time than the simulation itself.

Using utilities contained as a part of the TDLAS EXPLORER software package, the solution (available in PLOT3D format) was made dimensional, and the temperature was calculated from the pressure and the density available as part of the primitive variable set using ideal gas law relationships. A grid representing a 2 mm diameter laser beam was created such that the path of the beam had a component parallel to the flow, and that it was contained fully within the boundary-layer far enough downstream that statistical analysis indicated the boundary-layer was fully-turbulent. Next, the variables from the full flow-field were interpolated on to the laser-beam grid as shown below in Figure 6.

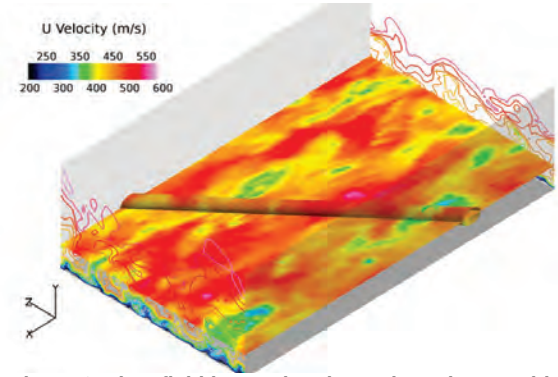


Figure 6. Flow-field interpolated onto laser-beam grid

The simulation of the laser degradation with O_2 as the target molecule was then performed with the parameters in Table 2 below which are representative of realistic equipment where parameters were taken from several sources (Verdyen, 1981; Brown, et al., 2011).

Table 2. TDLAS simulation parameters

Property	Value
O_2 Mole Fraction	0.23
Optical Frequency Peak (cm^{-1})	13153
Line Strength Constant (cm^2)	2.1×10^{-22}
Sweep Time (S)	1.0×10^{-3}
Samples per Sweep	100
Min/Max Sweep Frequency (cm^{-1})	13152.8/13153.2

Given the total simulation time of 2.66×10^{-3} seconds, two complete sweeps were performed. Figure 7 shows the results of the TDLAS simulation in terms of intensity drop vs. wave-number. These outputs represent what would be expected as output of an actual TDLAS system after the component has been removed (as previously described in Figure 4). Slight variations are visible between the two sweeps in the figure on the left as a result of changes in velocity and thermodynamic quantities as described below. It should be noted that the magnitudes of the intensity drop are very low due to the scaling of the density of the media. Realistic signal magnitudes can be found by scaling the computed flow-field. The figure on the right compares the computed dataset to a single time-point where the pressure and density have been scaled back to more realistic values to correct for the previously-described Reynolds number scaling. This serves to highlight the large difference in magnitude in the absorptivity caused by scaling of the density (note the overall changes in magnitude), as well as changes in the line-shape caused by scaling of the pressure (note the greater half-width corrected figure).

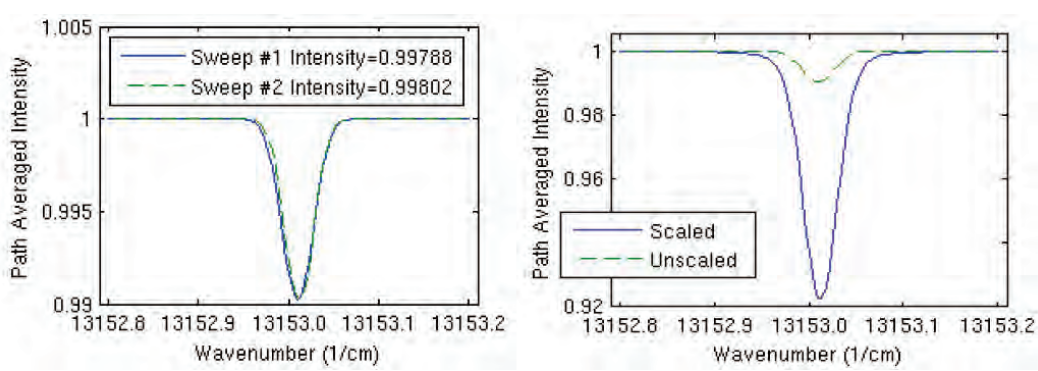


Figure 7. Sweep simulation intensities for transient (left) and scaled vs. unscaled comparison (right)

The transition from laminar to turbulent boundary-layer profiles cause the changes in velocity previously noted and shown in Figure 8. During the first sweep, the span-wise averaged velocity is observed to change from the smooth fuller laminar profile used as initial conditions to the fluctuating turbulent profile indicative of transition. Interestingly, the overall magnitude of the velocity increases a significant amount even though the free-stream velocity remains the same. The

transition from the laminar initial conditions to turbulent flow causes the velocity profile in the boundary-layer to become more full, presenting a larger velocity at the height of the laser (center at y=3 mm).

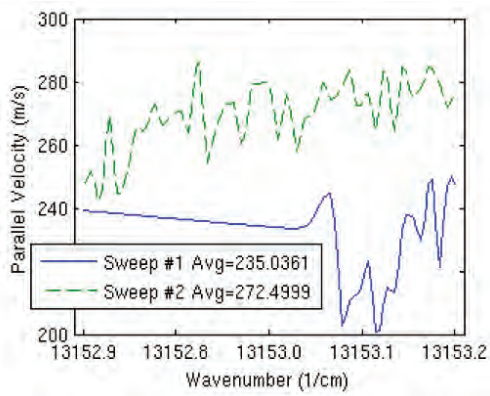


Figure 8. Parallel velocity vs. wave number for two sweeps

In realistic measurement systems the span-wise average velocity parallel to the laser path for the entire sweep is calculated based on the Doppler shift from the known-peak absorptivity to the measured-peak absorptivity as shown in the equation below, where ν_0 is the optical frequency peak, $c=3\times10^8$ is the speed of light, and $\theta=33°$ is the angle between the free-stream and the laser path. Table 3 below compares the actual velocity calculated directly from the simulation flow-field and the velocity calculated through the Doppler shift of the simulation. The high-error is likely a combination of the transient velocity fluctuations (i.e., higher- or lower-velocity structures may be present during the small period of time when the frequency is near the peak) and the lower-frequency resolution (100 points vs. 1,000 per sweep).

$$V = \frac{\Delta v_0}{v_0}\frac{c}{\cos(\theta)}$$

Table 3. Velocity measurement through Doppler shift

Sweep	Actual Velocity (m/s)	Doppler Shift (cm^{-1})	Simulation Velocity (m/s)	Error (%)
1	235	1.017×10^{-2}	278	18
2	273	1.202×10^{-2}	328	20

The changes in the temperature and density are shown in Figure 9. The changes to pressure were small and are omitted. Although the free-stream temperature was unchanged during the simulation, the increase in mixing caused by the transition from laminar to turbulent boundary-layer flow causes the colder free-stream air to mix and lower the bulk temperature in the boundary-layer. This would cause decreases in the half-width parameter of the lineshape, as observed above in Figure 7. The density increased in response to the decrease in temperature, leading to an increase in the absorptivity of the media and thus a decrease in the overall magnitude of the transmitted intensity as was also shown in Figure 7**.**

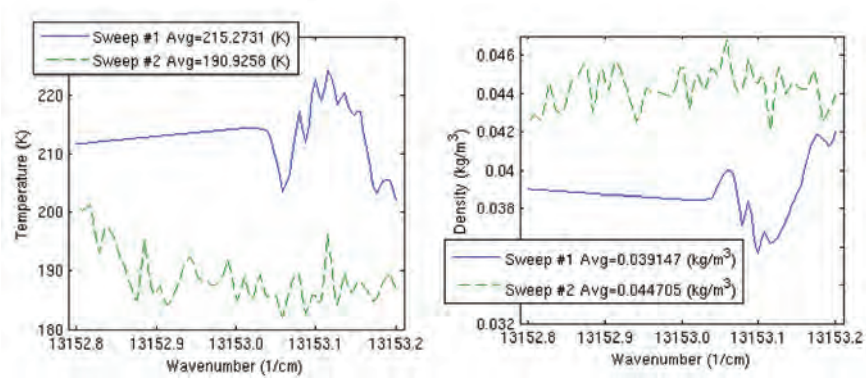


Figure 9. Sweep temperature (left) and density (right)

4. Conclusion

In this paper, laminar-to-turbulent transition has been detected through multi-physics TDLAS simulations on a LES dataset, indicating that laminar-to-turbulent transition should be detectable through use of an actual TDLAS measurement system. Several changes are observable in the output signal as a result of the changes in thermodynamic quantities and in the velocity magnitude caused by the transition. Although the overall magnitude of the changes are small, due to scaling of the Reynolds number necessary to perform the CFD simulation, a rescaled steady-state simulation shows that realistic magnitudes should be well-within the detectable range of actual measurement systems. It is furthermore believed that significant improvements in simulation time can be achieved through tighter coupling of the TDLAS EXPLORER software with the ILES code.

Acknowledgments

This work has been supported by AFOSR (Monitor: John Schmisseur) and the Ohio Research Scholars Program. The authors are grateful for a grant of HPC time from the DoD Supercomputing Resource Centers (DSRCs) at AFRL and ERDC. Support for the development of TDLAS EXPLORER was provided by the AFRL/RZ.

References

Barhorst, T., Williams, S., Chen, Shin-Juh, Paige, M., Silver, J.A., Sappey, A., McCormick, P., Masterson, P., Zhao, Q., Sutherland, L., Smith, I., VanHoudt, P., Hannam, J., and Owenby, D., "Development of an In Flight Non-Intrusive Mass Capture System." *AIAA Paper 2009-5067*, 2009.

Brown, M.S. and Barhorst, T., "Post-Flight Analysis of the Diode-Laser-Based Mass Capture Experiment Onboard HIFiRE Flight 1." *AIAA Paper No. 2011-2359*, 17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, San Francisco, CA, April 11–14, 2011.

Lindstrom, C.D., Jackson, K.R., Williams, S., Givens, R., Bailey, F.W., Tam, C.-J., and Terry, W.F., "Shock-Train Structure Resolved with Absorption Spectroscopy Part 1: System Design and Validation." *AIAA J.*, 47(10), pp. 2368–2378, 2009.

Mullenix, N.J., Gaitonde, D.V., and Visbal, M.R., "A Plasma-Actuator-Based Method to Generate a Supersonic Turbulent Boundary Layer Inflow Condition for Numerical Simulations." *AIAA-2011-3556*, 20th AIAA Computational Fluid Dynamics Conference, Honolulu, HI, June 27–June 30 2011.

Mullenix, Nathan J., Gaitonde, Datta V., and Visbal, Miguel R., "Generation of an Equalibrium Turbulent Boundary Layer for Upstream Boundary Condition Specification." *AIAA Paper No. 2012-0439*, 50th Annual Aerospace Sciences Meeting, Nashville, TN, January 09–12, 2012.

Sappey, A., McCormick, P., Masterson, P., Zhao, Q., Sutherland, L., Smith, I., VanHoudt, P., Hannam, J., Owenby, D., Williams, S., and Barhorst, T., "Development of a Flight-Worthy TDLAS-Based Oxygen Sensor for HIFiRE-1." *AIAA Paper 2009-4971*, 2009.

Verdyen, Joseph, *Laser Electronics*, Chapter 7, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

Yentsch, Robert J. and Gaitonde, Datta V., "Numerical Investigation of Hypersonic Phenomena Encountered in HIFiRE Flight 1." *AIAA Paper No. 2012-0943*, 50th Annual Aerospace Sciences Meeting, Nashville, TN, January 09–12, 2012.

Engine-Level Simulation of Liquid Rocket Combustion Instabilities: Trans-critical Combustion Simulations in Single Injector Configurations

Nicolas Guézennec, Matthieu Masquelet, and Suresh Menon
Department of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA
{nicolas.guezennec, gtg540e, suresh.menon}@gatech.edu

Ramakanth Munipalli
HyPerComp, Inc., Westlake Village, CA
mrk@hypercomp.net

Edward D. Lynch and Brian Lariviere
Pratt & Whitney Rocketdyne, Inc., Canoga Park, CA
{edward.lynch2, brian.lariviere}@pwr.utc.com

Douglas G. Talley
US Air Force Research Laboratory, Space and Missile Propulsion Division, Aerophysics Branch (AFRL RZSA), Edwards AFB, CA
douglas.talley@edwards.af.mil

Abstract

Detailed understanding of turbulent combustion in liquid rocket engines (LRE) requires an ability to predict the coupling between the transient features, acoustics, vortex/shear-layer dynamics and the unsteady combustion heat release. Conventional and ad hoc models that mimic or match one set of conditions but fail in another test case cannot be used for reliable predictions. This paper presents a simulation strategy based on Large-Eddy Simulation (LES) that uses a finite-volume scheme on multi-block, structured grids and solves the full multi-species, compressible LES equations using a hybrid central-upwind scheme to capture both turbulence shear flow and large density gradients. The sensitivity of predictions to the real gas equation of state such as the Peng-Robinson one is addressed in this study. The main modeling challenges concern the simultaneous capture of the flame structure, the flame-turbulence interactions and the regions of compressibility. The current work focuses on turbulent combustion in three single-injector configurations for these objectives: (a) trans-critical liquid oxygen (LOX)/gaseous hydrogen (GH2) combustion, (b) trans-critical LOX/methane combustion and (c) high-pressure GOX/methane combustion with thermo-acoustic instabilities. Results will be reported on the flame structure, liquid core length, and spreading rate, and comparison with data where appropriate. Finally, for LES of such problems, a more fundamental challenge is to determine the implication of the LES sub-grid closures for real gas flame dynamics. As a preliminary effort, the Linear-Eddy sub-grid model (LEM) is being applied to some of these cases.

1. Introduction

Modeling turbulent combustion in liquid rocket engines (LRE) is one of the main research challenges for the aerospace community. These engines produce a colossal amount of energy in a confined volume (>10GW/m^3). This high-energy release density leads to high-likelihood of thermo-acoustic instabilities, which are organized, oscillatory motions sustained by combustion. They manifest themselves as large amplitude pressure oscillations and are the result of non-linear coupling between unsteady heat release and the acoustics of the device. Combustion instabilities can completely destroy a liquid rocket engine in less than one second, and are notoriously difficult to predict and solve. To avoid such a catastrophic event during the rocket flight, many validation ground tests have to be performed during the design and the production of the engine. For example, the European engine Vulcain required 285 test firings totalizing 85,000 s of operation (Brossel, et al., 1995), the equivalent of more than 120 rocket launches. This empirical approach is prohibitively expensive. Therefore, introducing predictive tools in the design process would significantly reduce the cost of liquid rocket engines. This requires

an understanding of combustion instabilities within the context of the high-pressures present in modern rockets. These pressures typically exceed the critical pressure of the propellants. However, some propellants, such as oxygen or methane, can be initially introduced in the chamber at a temperature below their critical temperature. They would then undergo a transition from a sub-critical temperature to a super-critical temperature as they are mixed and burned in the engine. The term "trans-critical" has been coined to refer to this process. Under this regime, surface tension and latent heat become null (Poling, et al., 2001). Atomization and vaporization are replaced by continuous mixing processes dominated by diffusion and turbulent convection (Bellan, 2000). Moreover, gas and liquid cannot be distinguished; trans-critical fluids may be very dense while keeping gas-like properties.

To respond to this challenge, the Propulsion Directorate of US Air Force Research Laboratory (AFRL) has initiated a program called Advanced Liquid Rocket Engine Stability Technology (ALREST). Part of the ALREST strategy rests on the Large-Eddy Simulation (LES) code LESLIE3D developed at the Georgia Institute of Technology. Previous studies have shown that LES is able to handle real gas thermodynamics and to predict the subtle details leading to combustion instability in complex geometries. Large-Eddy Simulations, mainly carried out on mixing layers and coaxial configurations for LOX/H2 flames, have extended the understanding of the processes which control trans-critical combustion. By computing a sector of a single-element experiment, Oefelein (2006) was able to provide a detailed characterization of the LOX/H2 flame near field. He also identified the primary holding mechanism leading to the flame attachment on the injector tip. Successful LES of combustion instabilities were also reported for laboratory scale combustors during the last decade: ramjets, dump combustors, gas turbines sector (Selle, et al., 2004), rockets engines (Smith, et al., 2011). But very few studies have been carried out in the case of a full industrial combustion chamber with multi-element injection. Recent LES of a helicopter annular combustor by Staffelbach, et al. (2009) revealed the growth of a self-excited azimuthal mode. Therefore, the objective of ALREST is to gather the state-of-the art on LES to perform simulations of combustion instabilities in full rocket combustors, including multi-element injection and models for super-critical reacting flows. Multiple simulations are planned under the ALREST effort, in a series of progressively more difficult cases; the end-point configuration targeted being the full-scale liquid oxygen (LOX) and methane combustor by Jensen, et al. (1989). The simulations are carried out on the Department of Defense (DoD) high performance computing (HPC) computers under the Challenge Project, *Engine-Level Simulation of Liquid Rocket Combustion Instabilities.*

The present paper reports the results obtained during the first seven months of the Challenge Project. The objective of this initial phase was to test LESLIE3D's numerical strategy by performing LES of laboratory scale rocket combustors. The first section presents the LES methodology applied in LESLIE3D. Benchmark scaling tests are also reported. Then three single-element simulations are described. Two selected cases are single shear injector elements under trans-critical conditions without any instabilities. Case 1 reproduces the LOX/GH2 reacting experiment in the Penn State University (PSU) cryogenic rig. Case 2 is the experimental LOX/CH4 Mascotte installation studied by Singla, et al., (Singla, et al., 2005). These two simulations have already been published (Guézennec, et al., 2012; Masquelet, et al., 2012). Results for these configurations will be summarized in this paper. Special attention will be carried out on Case 3, the Continuous Variable Resonance Chamber (CVRC) of Purdue University (Yu, 2009). This high-pressure gaseous oxygen/methane experiment presents longitudinal combustion instabilities.

2. The Code LESLIE3D

2.1 Numerical Methods

In the present study, the unsteady Favré-filtered compressible Navier-Stokes equations are solved with the code LESLIE3D, a structured multi-block parallel solver developed at the Georgia Institute of Technology. The LES sub-grid terms are closed by solving an additional transport equation for the sub-grid kinetic energy $k^{sgs} = \frac{1}{2}(\widetilde{u_i u_i}) - \tilde{u}_i \tilde{u}_i$ and by applying a sub-grid viscosity with gradient diffusion approach (Menon and Patel, 2006). The model for the sub-grid eddy viscosity is based on a characteristic length-scale (the filter size $\bar{\Delta}$) and a characteristic velocity (the turbulent fluctuations) and yields $v_t = C_v \bar{\Delta} \sqrt{k^{sgs}}$. In this study, C_v is a constant equal to 0.067. A dynamic option also exists (Chakravarthy and Menon, 2001) and will be considered in future studies.

To complete this LES approach, additional models are required to compute pressure from the conservative variables and to compute the transport properties from primitive variables. For the thermodynamics, real gas models are required to predict the properties of the liquid oxygen. Analytical cubic equations of state (EOS) are a good choice for computational fluid dynamics (CFD) solvers since they provide a good compromise between accuracy and cost. The Peng-Robinson

EOS is the one chosen for this study, since it predicts accurate energies and specific heats (Poling, et al., 2001), which makes it appropriate for reacting cases.

The code has a unique multi-scale approach to deal with reactive physics within the LES context. Since in LES, all scales smaller than the grid are modeled using sub-grid models, a straightforward extension of this approach to reacting flows is incorrect because, for combustion to occur, the species must mix molecularly. This mixing occurs at small scales that are not resolved in a conventional LES approach. Therefore, a new grid-within-grid approach was developed based on the Linear-Eddy Mixing (LEM) model whereby a finer grid is embedded inside the LES grid to simulate the scalar mixing and reaction processes (hence the name LESLIE3D). The ability of this approach to capture fine-scale flame-turbulence interactions, extinction-reignition, flame liftoff, etc., has been demonstrated in many publications. The LEM based LES is the baseline for the highest fidelity simulations that can be achieved using LESLIE3D.

The flow solver uses a finite-volume scheme with a second-order-accurate predictor-corrector temporal integration and a second-order-accurate hybrid solver for spatial integration (Genin and Menon, 2010). This finite-volume hybrid solver alternates between a second-order central scheme and a third-order-accurate MUSCL, an upwind-biased scheme. A dynamic and local switch, based on pressure and density gradients, determines (at each time-step and computational face) which scheme to use. The MUSCL reconstruction technique is used alongside an approximate Riemann solver (specifically, Harten-Lax-Van Leer (HLL)) with HLL contact/solver modifications, as well as the monotonized central limiter, to enforce the total variation diminishing condition. This hybrid scheme allows the capture of the large density gradients typically found near the injection plane, while keeping the required grid resolution reasonable and keeping the less dissipative central scheme in the far-field to accurately model the turbulence.

2.2 Code Scaling

Strong scaling results on Raptor (Cray XE6) are shown in Figure 1 for two cases that are representative of the computational conditions encountered during the project: a temporal mixing layer (TML) 256^3 (LEMLES, 2 species, 6 equations) and the Mascotte case presented in section 4.2 (reacting LES, 4 species, 8 equations). These configurations are also solved with the

$$\text{Speed-up} = \frac{\text{Execution time using } N_{ref} \text{ processors}}{\text{Execution time using } N \text{ processors}} \tag{1}$$

$$\text{Efficiency} = \frac{\text{Execution time using } N_{ref} \text{ processors } \times N_{ref}}{\text{Execution time using } N \text{ processors} \times N} \tag{2}$$

The code shows a very good scaling. Overall, efficiency greater than 80% is sustained from 32 to 2,000 processors. The typical number of processors used for the cases presented in this paper is around 1,000 processors, which corresponds to an efficiency of about 85%.

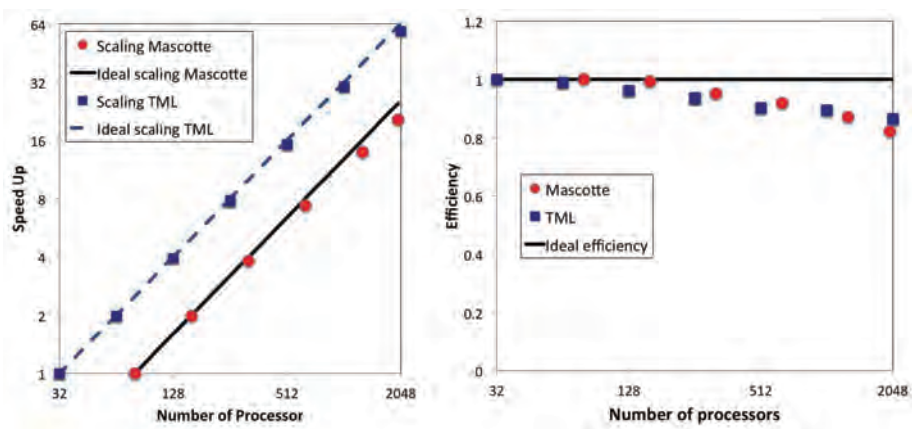


Figure 1. Strong scaling of the 256^3 TML with LEM and the Mascotte case on Raptor (Cray XE6): Hybrid 2nd-order centered/3rd-order MUSL scheme and Peng Robinson EOS. N_{ref}=32 for the TML and N_{ref}=80 for the Mascotte case. Left: Speed-up, Right: Efficiency.

3. Configurations

The first year of the Challenge effort focused on the simulation of three single-element configurations representative of the conditions encountered in the rocket engines. In particular, they share the same type of shear coaxial injection: the oxidizer is injected through a central pipe and mixes with the annular fuel stream.

3.1 Experimental Conditions

The LOX/GH2 PSU rig: The first set of simulations focused on the reacting single-element LOX-GH2 coaxial experiments conducted by Locke (2011). In order to prevent recirculation regions in the corners of the chamber and to mimic the co-flow in a multi-element configuration, the center coaxial flow is surrounded by a co-flow generated by a perforated plate with 78 holes (Masquelet, et al., 2012). High-speed backlit visualization of the flame is available, which allows for some instantaneous and statistical comparisons of the oxygen core length. The case of interest here includes a super-critical chamber pressure, with respect to the oxygen species (p_r=63 bar/55 bar>1). Since the oxygen injection temperature is sub-critical (T_r=125/155<1), the initial liquid jet is expected to undergo a trans-critical, even as it transitions from compressed liquid to super-critical fluid while mixing and reacting with the surrounding hydrogen and combustion products.

The MASCOTTE V04 experimental rig: This LOX/CH4 single-element shear coaxial injector was investigated by Singla, et al. (2005) on the Mascotte test rig. This small-scale thrust chamber is operated by ONERA. The combustor is composed of a square chamber equipped with visualization windows on its four sides. The mean flame structure was obtained from Abel transform of OH and CH LIF emission images. Therefore only qualitative comparison between experiment and LES will be possible. The pressure in the combustor is P=5.61 MPa which is above the critical pressure for both reactants. Methane is injected as a super-critical gas (T_{CH4}=288K and $\dot{m}_{CH4}$=143.1g/s) and the oxygen remains trans-critical (T_{O2}=85K and $\dot{m}_{O2}$=44.4g/s).

The CVRC experiment: The last case is the CVRC (Continuous Variable Resonance Chamber) experimental test rig (Yu, 2009). This system, based at Purdue University, is designed to study longitudinal combustion instabilities while varying the length of the chamber. It is composed of a fuel injector, an oxidizer injector whose length can vary, and a combustion chamber. For this study, the length of the oxidizer injector has been kept constant with L=16.5 cm. A choked slot and a sonic nozzle acoustically close the experiment at the inflow and the outflow, respectively. The oxidizer comes from the decomposition of a mixture containing 90% of hydrogen peroxide and 10% water. Therefore, the oxygen mass fraction in the inlet is $Y_{O_2} = 0.42$ and the water mass fraction is $Y_{H_2O} = 0.58$. The inlet temperature for the oxidizer is 1,030K and the mass flow rate is 0.32 kg/s. The fuel is pure methane injected with a mass flow rate of 0.027 kg/s and a temperature of 325K. The measured pressure in the system is 1.34 MPa.

3.2 Numerical Setup

For the three cases, the simulations are performed on a multi-block grid composed of an outer cylindrical domain of an inner butterfly Cartesian domain. The minimum grid-spacing is always located in the main shear-layer between the coaxial streams. The grid used for the CVRC simulation is shown in Figure 2 and is representative of all grids run for this work. The CRVC simulations model the upstream part of the injector as it is critical for the study of combustion instabilities but this is neglected as a first approximation for the trans-critical cases where the grid starts only a few LOX diameters upstream of the injection plane. Also, for acoustic reasons, a convergent-divergent nozzle is included in the CVRC simulations while the chamber pressure is kept constant in the trans-critical cases through the application of characteristic partially-reflecting boundary conditions. The trans-critical cases also include non-reflecting boundary conditions at the inflow. No-slip adiabatic conditions are imposed at walls for all the cases. More specific characteristics about the trans-critical cases are detailed in the papers of Masquelet, et al. (2012) and Guézennec, et al. (2012).

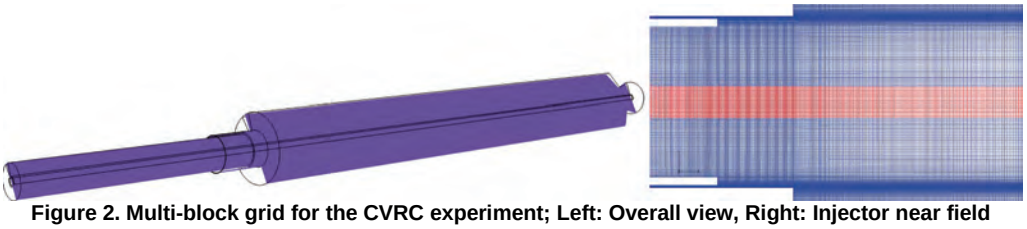


Figure 2. Multi-block grid for the CVRC experiment; Left: Overall view, Right: Injector near field

The CVRC computational domain starts slightly downstream from the oxidizer and the fuel choked inlets which are replaced by characteristic constant mass inflows. The total number of grid-points is about 12 million with a minimum spacing of 40 μm. Two types of LES have been performed with LESLIE3D on the CVRC configuration: axisymmetric simulation and 3D-LES. The axisymmetric LES was performed on a slice of the three-dimensional (3D) grid with a similar resolution and the flame/turbulence interaction was accounted for by the Linear-Eddy Model. The number of cells in each LEM one-dimensional (1D) domain is chosen to be 12 which resolve scales of the order. A 3D computation with this model is underway but only 3D-LES results with laminar chemistry will be reported in this paper. The combustion between CH_4 an O_2 is modeled with a simple 1-step mechanism taking into account 4 species: CH_4, O_2, CO_2, and H_2O.

4. Results and Discussion

4.1 LOX/GH2 PSU Rig

Initial simulations did not include the holes of the perforated plate, but instead used a simple, uniform and slow co-flow. The overall flow structure was similar to the one shown in Figure 3, except that the oxygen jet would break-up even earlier. However, the overall flame structure was identical with the central LOX core bounded by a trans-critical layer where thermodynamic and transport properties undergo rapid changes. The oxygen jet is surrounded by a flame shell anchored at the injector tip, while the hydrogen annular stream expand outwards and mixes with burned products thanks to strong vortical structures. Further analysis of the transient flow-field shown in Figure 3 suggests that the jets from the faceplate holes display good coherence for 5 to 10 LOX diameters downstream before quickly breaking down into a more uniform co-flow. This coherence helps maintaining the coaxial flow compact and axial, limiting the onset of the helicoidal instabilities and delaying the oxygen jet break-up. However, as soon as the co-flow loses its coherence, the main shear-layer is destabilized and large-scale motions away from the centerline start to appear, which help break up the oxygen jet. Then results suggest that, for maximum fidelity with the experimental rig, it might thus be necessary to model all the holes in the perforated plate.

The helicoidal instabilities start to appear when the dense core starts to narrow and the flame structure becomes more complex. Instead of the perfect diffusion flame structure described in the previous paragraph, significant premixing can occur as the vortical structures start to interact with less dense oxygen. This is illustrated in Figure 3. It is postulated that these changes in the flame structure, and thus the heat release around the oxygen jet, contribute to the helicoidal instabilities and help the jet break-up. The onset of these instabilities occurs earlier in the current simulations compared to what is seen by the experimental high-speed visualization. This highlights the need for advanced turbulent combustion models for trans-critical flames with higher momentum flux ratios.

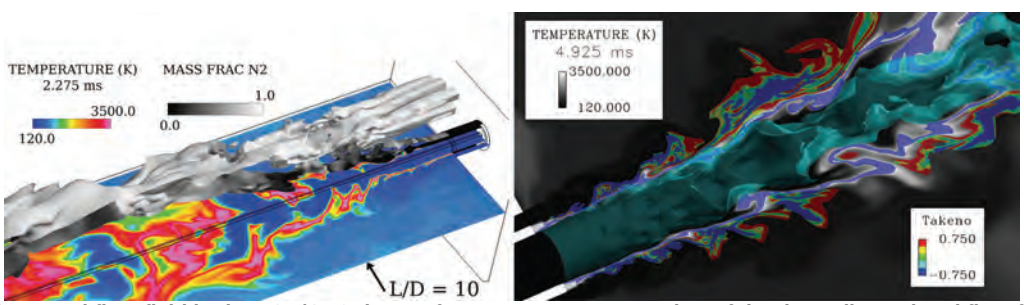


Figure 3. Flame and flow-field in the LOX/GH2 rig. Left: Instantaneous snapshot of the three-dimensional flow-field for the perforated plate case. The temperature field is shown with color contours while the co-flow structures are shown through the iso-surface of 100 m/s axial velocity. This iso-surface is colored by nitrogen mass fraction, and is only shown for half of the domain. Right: Flame structure in the near-field of the trans-critical LOX/GH2 injector. The teal iso-surface represents the compressed liquid oxygen, while the background grayscale contours show the temperature field. The nature of the flame is indicated by the color contours of the normalized Takeno index: a value of -1 indicates a perfect diffusion flame while a value of 1 indicates a perfect premixed flame.

4.2 LOX/CH4 Mascotte Test Case

To illustrate the turbulent flow structure, Figure 4 (left) shows iso-contours of Q criterion colored with the velocity magnitude. The density iso-surface ρ=150 kg/s is also presented in purple to visualize the dense core. Annular coherent-eddies occur in the shear-layer between the methane stream and the quiescent environment. They entrain gas from the low-velocity zone to the high-speed jet. These structures wrinkle the trans-critical oxygen core and break down to

generate a highly-turbulent flow. The LES temperature iso-surface T=1,700K is compared with a visualization of the experimental flame on Figure 4 (right). This qualitative comparison suggests a good agreement between LES and experiment. The flame is anchored on the tip between the CH4 and the LOX injection. The reaction zone is a short cone (around 6 cm) starting with an expansion angle lower than 10° and abruptly terminating with an angle of 20°.

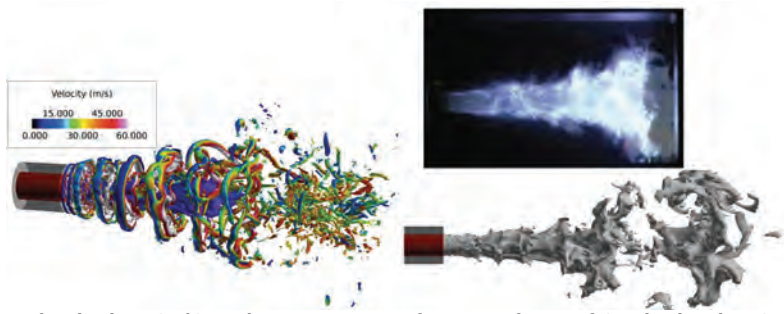


Figure 4. Turbulent combustion in the LOX/CH4 rig Mascotte. Left: Iso-surfaces of Q-criterion (=5e8 s^{-2}) colored by the velocity magnitude and density iso-surface ρ=150 kg/m^3 (purple). Right-top: visualization of the experimental flame (Singla, et al., 2005). Right-bottom: LES temperature iso-surface T=1700K.

Figure 5 (left-top) presents scatter plots of species mass fraction (CH4, O2 and H2O) versus the mixture fraction. The points gather around straight lines intersecting at the stoichiometric mixture fraction, ζ=0.2, suggesting that an infinitely fast chemistry flamelet model is valid for this case. The diffusion regime is confirmed by the flame criterion $\Gamma_f = \nabla Y_{O_2} . \nabla Y_{CH4}$ (Figure 5 left-bottom). Negative values of Γ_f correspond to a diffusion flame and positive values define zones of partially-premixed combustion. The criterion shows very high negative values along the stoichiometric mixture fraction, while nearly no positive value is observed elsewhere in the flame. This kind of situation may be specific of the Mascotte test conditions and is not representative of the actual rocket operating conditions. Since the Reynolds number in real rocket engines is much higher, the flame experiences a higher amount of strain. Therefore, the Damköhler number might take a small-value and finite-rate chemistry must be taken into account.

The radial variations of fluid properties across the flame are shown in Figure 5 (right). All these quantities present sharp gradients in the shear-layer (0.5=<r/D_{LOX}<0.7) where both mixing and combustion occur. The compressibility factor $Z=PV/RT$ measures the departure of the fluid from the ideal gas assumption. In the methane stream and in the far-field (r/D_{LOX}>0.6), Z remains close to 1, while in the oxygen core it is close to 0.2. In this zone, density predicted by the Peng Robinson equation of state is greater than 1,000 kg/m^3 whereas the perfect gas equation of state would provide a density around 230 kg/m^3. Even if the perfect gas assumption might be reasonable in most of the computational domain, applying this equation of state would actually entail important discrepancies on the flame expansion. This conclusion is strengthened by the plots of Prandtl, Lewis, and Schmidt numbers, all of which show several orders-of-magnitude variation through the flame. As observed by Oefelein (2006) for the LOX/H2 flame, the comparison of these three numbers shows that the mass diffusion is the slowest transport phenomenon in the liquid oxygen and is rate-limiting for the combustion on this side of the flame.

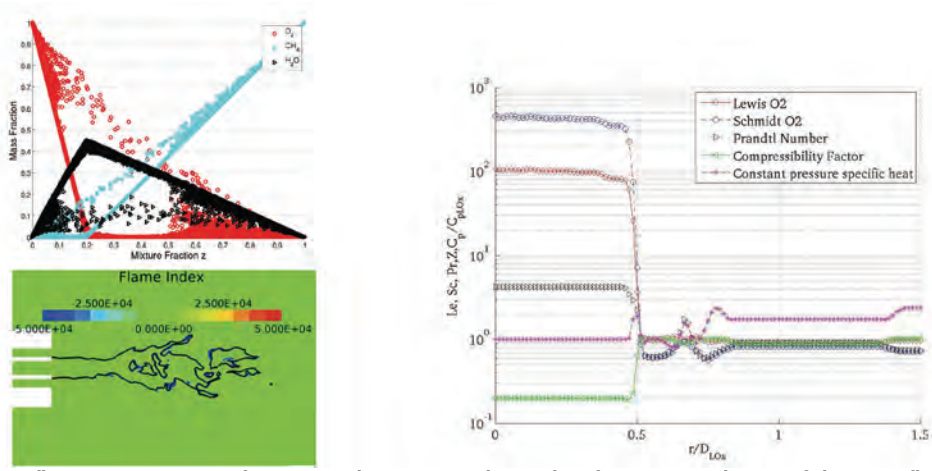


Figure 5. LOX/CH4 flame structure. Left-top: species versus mixture fraction scatter plot. Left-bottom: flame index. Right: Radial variation of compressibility factor, kinetic viscosity, Prandtl, Schmidt and Lewis numbers at an axial location x/D_{LOX}=1.

4.3 LES of Combustion Instabilities in the CVRC Experiment

4.3.1 Mean Flow-Field

Figure 6 presents time-averaged velocity fields for the axisymmetric and the 3D computations. The averaging time is 20 ms for the axisymmetric LES and 10 ms for the 3D, which corresponds to 12 and 6 flow-through-times in the combustion chamber, respectively. An additional azimuthal averaging is also carried out for the 3D case. The black line in Figure 6 corresponds to the iso-contour U=0 m/s. It identifies the limit of the recirculation zone, which takes place behind the step. Important differences between the two cases are noticeable. The recirculation zone predicted by the axisymmetric LES is three-times longer than the 3D one. It also shows a very long, high-velocity core which extends from the injector to the nozzle while the 3D mean flow is more similar to a jet. The shear-layer converges toward the center line and the potential core collapses 5 diameters downstream from the injector. Then the axial velocity on the center line decreases up to the supersonic nozzle. Such discrepancies are expected and illustrate the limits of the axisymmetric approximation: the boundary condition on the center line prevents the oxidizer jet from flapping, and the vortical structures are stronger in the axisymmetric case since there is no third-dimension to provide development and vortex stretching effects.

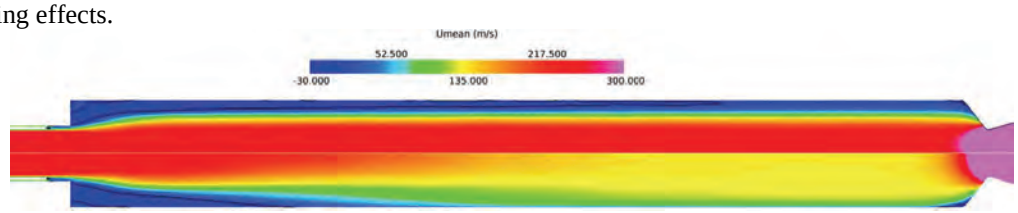


Figure 6. Average axial velocity: (top) Axisymmetric, (bottom) 3D; Black line: velocity iso-contour U=0 m/s

These differences directly impact the turbulent flame brush. Figure 7 presents the average temperature fields. The black line corresponds to the iso-contour T=2,000K and highlights the reaction zone limits. For each case, the flame is anchored at the corner between the injector and the combustor step. But the axisymmetric flame extends from the injector all the way to the nozzle, whereas the 3D flame is compact and is closed after less than 5 injector diameters.

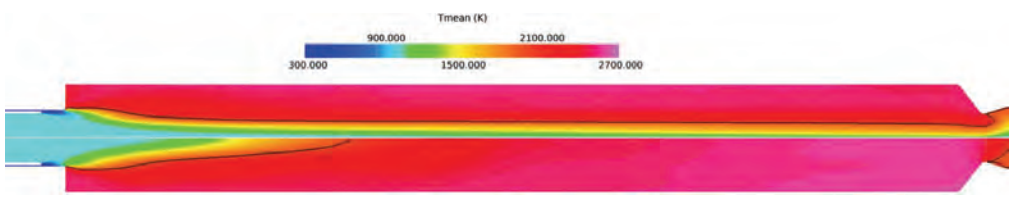


Figure 7. Average temperature: (top) Axisymmetric, (bottom) 3D; Black line: temperature isocontour T=2000K

4.3.2 Flame Regime and Stabilization

To illustrate the flame structure, Figure 8 shows instantaneous fields of methane reaction rates (top) and flame index $\nabla Y_{CH4} \cdot Y_{O2}$ (bottom). The white line represents the iso-contour T=2,000K and the black line corresponds to the stoichiometric iso-contour z=0.095. The CVRC experiment is, at first sight, a simple coaxial configuration where reactants are injected separately. Such a setup usually generates a diffusion flame that can be lifted or attached to the injector lip. The presence of a very long recess in the CVRC case leads to a different flame structure with a more complex stabilization mechanism. First the flame is anchored to the step corner. The long recess delays the combustion and generates a shear zone where methane is mixed with the oxidizer without burning. Then, the recirculation zone in the step corner (Figure 6) allows the reactant mixture to ignite by bringing back hot products. This stabilization process yields a very intense partially-premixed flame. Further down in the combustion chamber, this initial reaction zone is replaced by a diffusion flame which follows the stoichiometric mixture fraction iso-contour. This second flame is weaker, especially in the 3D case, where it nearly vanishes.

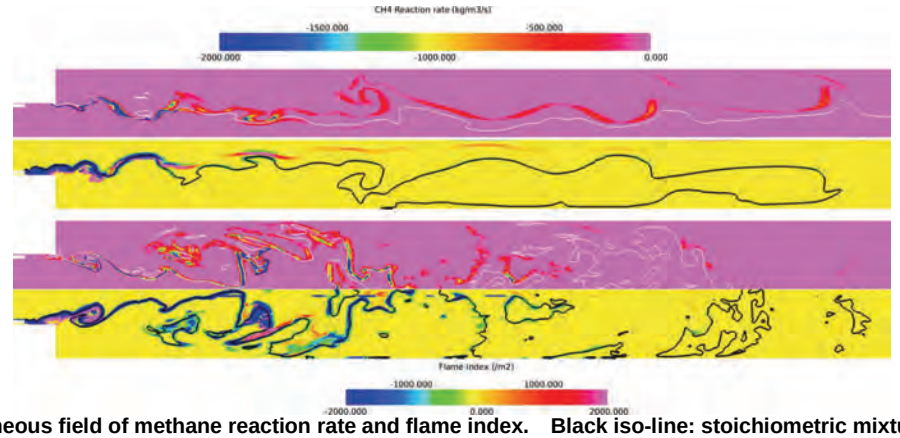


Figure 8. Instantaneous field of methane reaction rate and flame index. Black iso-line: stoichiometric mixture fraction ($\zeta=\zeta_{st}$); white iso-line: temperature T=2,000K; top: Axisymmetric, bottom: 3D.

This flame structure might explain the origin of the combustion instabilities. A large part of the heat release is produced by a partially-premixed flame which makes this configuration analog to a gas turbine configuration where combustion instabilities are clearly related to premixed combustion. In a rocket engine, combustion instabilities are often explained by a transverse coupling between the injectors. This single-element experiment suggests that the flame structure might also play a role by switching from diffusion to premixed combustion. Such a complex scenario requires applying finite-rate chemistry and a robust LES sub-grid combustion model independent of the combustion regime. LEMLES has been shown able to fulfill this requirement. In order to support this assertion, Figure 9 shows the 3D reaction rate iso-surface predicted by LESLIE3D with LEM (left) and without any model for the filtered reaction rate (right). The LEM captures the flame front in a much more meaningful manner showing a connected surface, whereas the no-model case shows broken up structures that are highly grid-dependent.

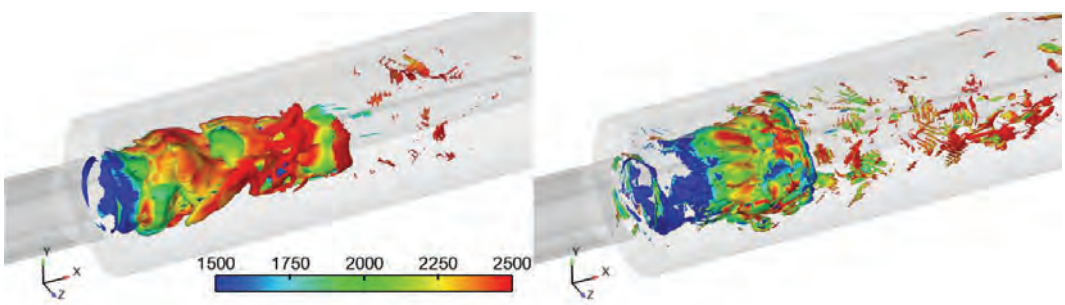


Figure 9. Reaction rate iso-surface $\dot{\omega}_{CH4}$ = –1000 kg/m^3/s colored by temperature; left: LEM-LES, right: LES with no-model for combustion

4.3.3 Unstable Modes

Figure 10 presents the pressure signal at x=1.25 cm and the spatial mode structure of the acoustic field. The pressure signal is essentially composed of two main harmonics $F_1^{LES} = 1519$ Hz and $F_2^{LES} = 3028$ Hz. These frequencies are slightly shifted from the experimental values $F_1^{exp} = 1400$ Hz and $F_2^{exp} = 2800$ Hz. This might be explained by the small shift between the simulation and the experimental operating points. The LES is using adiabatic walls and predicts a mean operating pressure P=1.6 MPa, while the experiment may have temperature losses at the walls and operates at 1.34 MPa. The spatial structure of the acoustic field is identical between the two LES, but the 3D simulation is providing a signal with amplitude nearly twice as much as the two-dimensional (2D) computation. The two first unstable frequencies correspond respectively to a half-wave and a quarter-wave mode shape in the combustion chamber while they are mainly propagative in the oxidizer tube. The LES acoustic field structure is also in good agreement with the experiment. Figure 11 compares LES and experimental power spectral densities in the oxidizer tube and in the combustion chamber. The 3D-LES and the experiment show that the first mode is dominant in the combustor, while the pressure spectrum in the injector contains several harmonics.

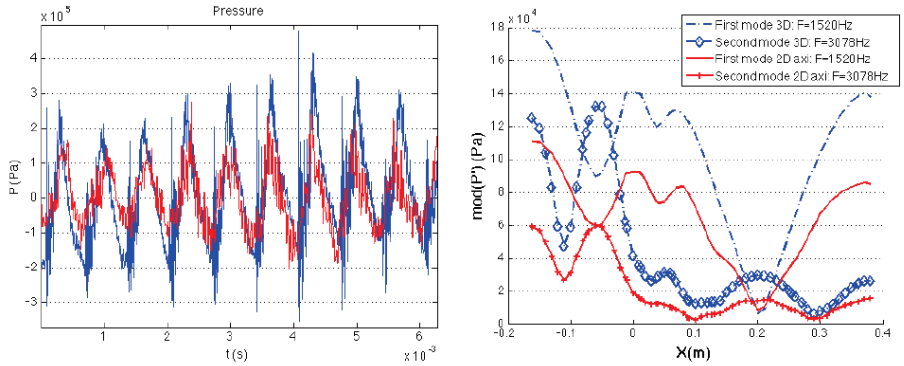


Figure 10. Unstable modes characterization (red: Axisymmetric, blue: 3D). Left: Pressure signal at x=1.25 cm. Right: Acoustic field structure.

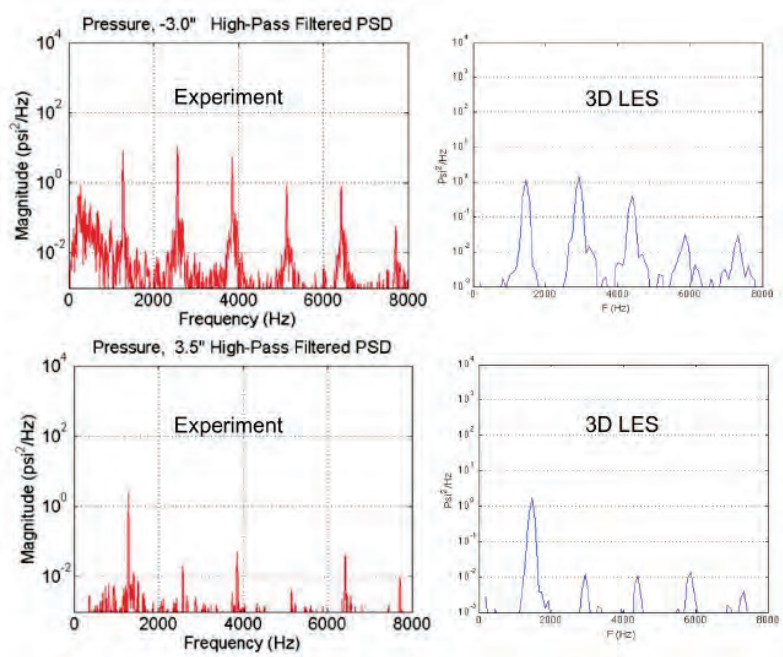


Figure 11. Pressure PSD calculated at various locations: (Top) oxidizer tube x=-7.6 cm, (Bottom) combustion chamber x=8.9 cm. Comparisons between experimental measurements (Yu, 2009)(Left) and 3D LES (Right).

5. Conclusion

This paper highlights the effort to develop a capability to model engine-level combustion instabilities using the LESLIE3D code. The solver uses a hybrid upwind-central scheme to capture the large-density gradients found in the trans-critical injection of oxygen through a coaxial injector, as well as the turbulence created by the shear-layers and the breakdown of the central jet. The modeling of the thermodynamics of such trans-critical injection is made possible through the use of cubic equations of state which provide a good compromise between cost and accuracy for Large-Eddy Simulations. Massively-parallel LES of trans-critical combustion and thermo-acoustic instabilities were performed on three laboratory scale setups. LESLIE3D is able to capture the strong real gas effects in the LOX/CH4 and LOX/GH2 trans-critical cases and to predict the combustion instabilities observed in the CVRC experiment. Moreover, various flame structures were observed on the three configurations: from diffusion flame with infinitely fast chemistry to partially-premixed combustion with important finite-rate chemistry effects. To provide a better description of the flame/turbulence interaction, the LEM needs to be applied to the trans-critical cases. The next step of the Challenge Project, *Engine-Level Simulation of Liquid Rocket Combustion Instabilities*, consists in extending this numerical approach to a full-scale LOX/CH4 engine (Jensen, et al., 1989). The device exhibited a rich range of stability behavior, including spontaneously stable, spontaneously unstable, bombed stable, and bombed unstable behavior. The computation of a reduced sector of

this experiment, which includes 5 of the 82 injectors, is under preparation and the grid for this simulation is shown in Figure 12. A complete closure using both LEM and detailed chemistry remains too expensive for simulations of real gas combustion in this full-scale geometry. Methods such as the artificial network methodology are therefore investigated to speed-up the chemistry kinetics calculation and circumvent this issue.

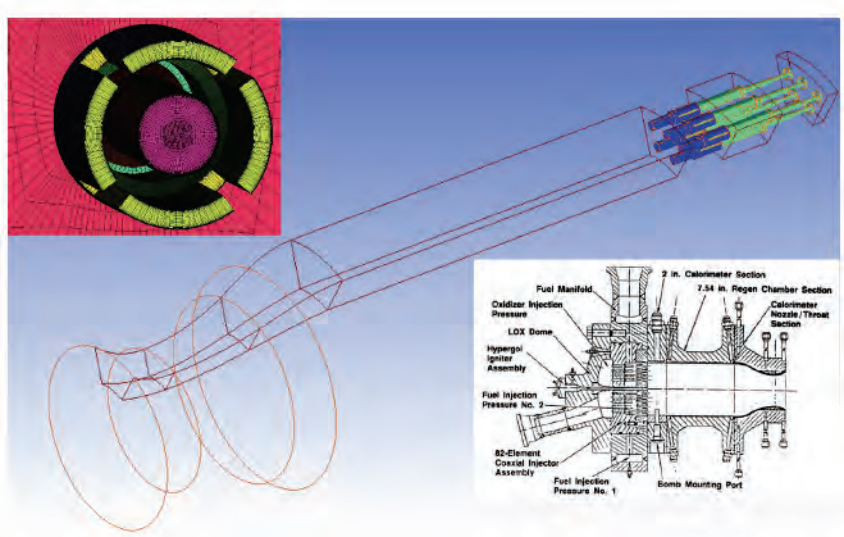


Figure 12. 82 LOX/Methane injector and thrust chamber. Bottom-right: sketch of the experiment (Jensen, et al., 1989); Center: grid of a reduced sector including 5 injectors; Top-left: detailed view of one coaxial injector.

6. Significance to DoD

This project directly supports the ALREST combustion instability program. Combustion instabilities are a serious risk to Air Force liquid rocket engines. The technology demonstrated here will also apply to other DoD and non-DoD applications.

Acknowledgments

This research was supported by the US Air Force Research Laboratory (AFRL) and Pratt & Whitney Rocketdyne. The computational resources at the DoD Supercomputing Resource Center (DSRC) and the US Army Engineering Research and Development Center (ERDC) were provided under a DoD HPCMP Challenge Project.

References

Bellan, J., "Supercritical (and subcritical) fluid behavior and modeling: drops, streams, shear and mixing layers, jets and sprays." *Progress in Energy and Combustion Science*, 26, pp. 329–366, 2000.

Brossel, P., S. Eury, P. Signol, H. Laporte-Weywada, and J.B. Micewicz, "Development status of the Vulcain engine." *31st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, San Diego, CA, 1995.

Chakravarthy, V.K. and S. Menon, "Subgrid modeling of premixed flames in the flamelet regime." *Flow, Turbulence and Combustion*, 65, 2001.

Genin, F. and S. Menon, "Simulation of turbulent mixing behind a strut injector in supersonic flow." *AIAA Journal 2010*, 48, No. 3, pp. 526–539, 2010.

Guézennec, N., M. Masquelet, and S. Menon, "Large eddy simulation of flame-turbulence interactions in a LOX-CH4 shear coaxial injector." *50th AIAA Aerospace Sciences Meeting*, Nashville, TN, 2012.

Jensen, R.J., H.C. Dodson, and S.E. Claflin, "Performance and stability analyses of rocket thrust chambers with oxygen/methane propellants." *Technical report NASA CR 182249*, 1989.

Locke, J.M., *High speed diagnostics for characterization of oxygen/hydrogen rocket injector flowfields*, Ph.D. thesis, The Pennsylvania State University, May 2011.

Masquelet, M., N. Guézennec, and S. Menon, "Numerical studies of mixing and flame-turbulence interactions in shear coaxial injector flows under trans-critical conditions." *50th AIAA Aerospace Sciences Meeting*, Nashville, TN, 2012.

Menon, S. and N. Patel, "Subgrid modeling for simulation of spray combustion in large-scale combustors." *AIAA Journal*, 44 (4), pp. 709–723, April 2006.

Oefelein, J., "Mixing and combustion of cryogenic oxygen-hydrogen shear-coaxial jet flames at supercritical pressure." *Combustion Science and Technology*, 178(1–3), pp. 229–252, 2006.

Poling, B.E., J.M. Prausnitz, and J.P. O'Connell, *The properties of gases and liquids*, McGraw-Hill, 5th edition, 2001.

Selle, L., G. Lartigue, T. Poinsot, R. Koch, K.-U. Schildmacher, W. Krebs, B. Prade, P. Kaufmann, and D. Veynante, "Compressible LES of turbulent combustion in complex geometry on unstructured meshes." *Combust. Flame*, 137(4), 2004.

Singla, G., P. Scouflaire, C. Rolon, and S. Candel, "Transcritical oxygen/transcritical or supercritical methane combustion." *Proceedings of the Combustion Institute*, 30, pp. 2921–2928, 2005.

Smith, R., G. Xia, W. Anderson, and C.L. Merkle, "Computational studies of the effects of oxidiser injector length on combustion instability." *Combustion Theory and Modelling*, 2011.

Staffelbach, G., L.Y.M. Gicquel, G. Boudier, and T. Poinsot, "Large eddy simulation of self-excited azimuthal modes in annular combustors." *Proc. Combust. Inst.*, 32, pp. 2909–2916, 2009.

Yu, Y.C., *Experimental and analytical investigations of longitudinal combustion instability in a continuously variable resonance combustor (CVRC)*, Ph.D. thesis, Purdue University, 2009.

High Resolution Near-Real-Time Image Processing and Support for MSSS Modernization

R. Bruce Duncan
Boeing Company, Kihei HI
robert.b.duncan@boeing.com

Chris A. Sabol
US Air Force Research Laboratory, Directed Energy Directorate(AFRL/RD), Kihei, HI

Kathy Borelli
KJS Consulting, Haiku, HI

Scott Spetka
ITT Corp, Rome, NY

Jason Addison
HNu Photonics, Wailuku, HI

Adam Mallo and Brad Farnsworth
Pacific Defense Solutions, LLC; Kihei, HI

Ron Viloria
Maui High Performance Computing Center, Kihei, HI

George Ramseyer
US Air Force Research Laboratory, Information Directorate (AFRL/RI), Rome, NY

Abstract

This paper describes image enhancement software applications engineering development work that has been performed in support of Maui Space Surveillance System (MSSS) Modernization. It also includes R&D and transition activity that has been performed over the past few years with the objective of providing increased space situational awareness (SSA) capabilities. This includes US Air Force Research Laboratory (AFRL) use of an FY10 Dedicated High Performance Investment (DHPI) cluster award—and our selection and planned use for an FY12 DHPI award. We provide an introduction to image processing of electro-optical (EO) telescope sensors data; and a high-resolution image enhancement and near-real-time processing and summary status overview. We then describe recent image- enhancement applications development and support for MSSS Modernization, results to date, and end with a discussion of desired future development work and conclusions. Significant improvements to image processing enhancement have been realized over the past several years, including a key application that has realized more than a 10,000-times speedup compared to the original R&D code—and a greater than 72-times speedup over the past few years. The latest version of this code maintains software efficiency for post-mission processing while providing optimization for image processing of data from a new EO sensor at MSSS. Additional work has also been performed to develop low-latency, near-real-time processing of data that is collected by the ground-based sensor during overhead passes of space objects.

1. Introduction

This document addresses and represents the work of a dedicated team that has been working over the past several years to transform image enhancement software from basic research to usable and additionally-enhanced high performance computing (HPC) software applications. The applications software and utilities addressed by this publication are utilized to process atmospherically-blurred raw images of space objects obtained from ground-based electro-optical telescope sensors, such as Maui Space Surveillance System (MSSS), and perform additional image processing to obtain high-resolution imagery information. The Air Force Maui Optical & Supercomputing (AMOS), including MSSS, is helping to increase the nation's space situational awareness (SSA) capabilities using advanced electro-optical-based collection and analysis technologies and is a national focal point for ground-based space SSA. This is a project of the US Air Force Research Laboratory, Directed Energy Directorate (AFRL/RD); with the work being accomplished by a distributed team led and performed by the AFRL/RD; US Air Force Research Laboratory, Information Directorate (AFRL/RI), and AFRL contractor personnel.

2. High-Resolution Image Enhancement and Near-Real-Time Processing

AMOS is continuing to provide enhanced image quality, utilizing the PCID image processing and user interface software packages that were developed by AFRL[1]. PCID stands for Physically-Constrained Iterative Deconvolution, and is an implementation of a multiple-frame blind deconvolution technique.

The PCID algorithm uses little or no *a priori* information about the objects that are being imaged and incorporates features that increase the probability and speed of finding the global minimum, achieve super-resolution by de-aliasing, process non-idealities such as spatially-cropped images due to jitter and/or object size relative to the detector size, and generate many of the PCID inputs automatically[2]. The forward model used in the PCID algorithm is the standard linear imaging model. Multiple frames of data are collected by sensors using telescopes, such as at MSSS, while the object, in this case the shuttle, passes overhead. Each collected frame is blurred by the earth's atmosphere. The name given to this blurring is point-spread function (PSF). Frames are grouped together to form an ensemble where the object is constant throughout all frames. PCID jointly estimates the common object and the blurring from each frame by iteratively estimating the object and the blurring per frame. The object and PSF estimates more closely match the collected frames with each processing iteration cycle. Physical constraints, such as object positivity, use of support for the object and PSF, as well as noise-weighting, contribute to fast convergence to the object estimate. PCID is best accessed by image analysts and other users through a Net-centric, or Web user, interface and job management system that has been co-developed with PCID called ASPIRE (Advanced Speckle Image Restoration Environment). The output of interest is an enhanced image, or recovery, of the object. PCID developers also researched and created Cramér–Rao estimation theory and statistics lower-bound (CRB) expressions for blind deconvolution, and carried out a comparison study that showed that PCID closely approaches these theoretical limits when run in the non-blind mode, and should be able to approach the theoretical limits in the blind mode when regularization and positivity is employed. A high-level graphic of PCID and ASPIRE is shown in Figure 1.

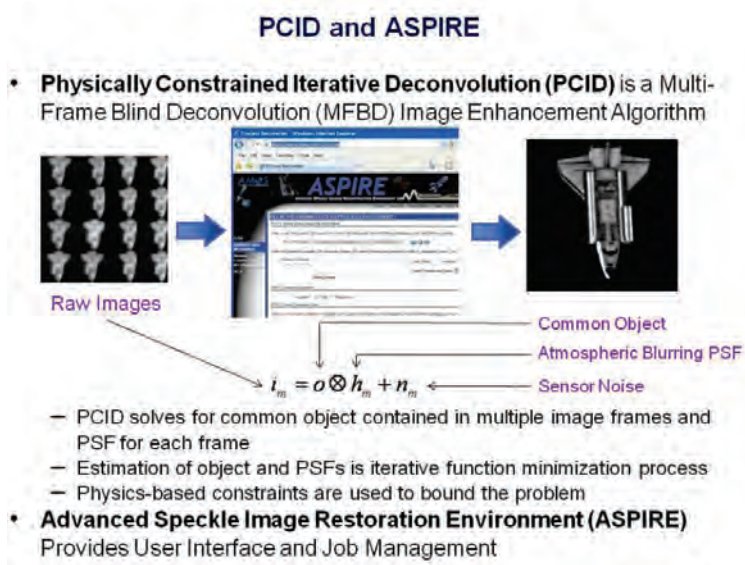


Figure 1. PCID and ASPIRE

Isoplanatic Mulit-Frame Blind Deconvoluton (MFBD) is another version of the MFBD algorithm. This algorithm performs a joint estimation of object and PSF (or its Fourier transform, the transfer function) parameters, and can use either Zernike or Spatial model PSFs. Like PCID, this algorithm also uses the Conjugate Gradient minimizer and is also based on Limited-Memory Broyden Fletcher Goldfarb Shannon Bounded (LBFGSB) Limited-Memory Quasi-Newton routines.

PCID and ASPIRE are continuing to utilize a DoD High Performance Computing Modernization Program (HPCMP) FY10 Dedicated High Performance Computing Project Investment (DHPI) platform whose characteristics are outlined below.

- DoD HPCMP FY10 DHPI Award − *Kaku*
- Repurposed and Upgraded Former MHPCC DSRC *Jaws* Nodes
- 150 Power Edge 1955 Compute Nodes
  - Intel Xeon Woodcrest Dual-Core Processors
  - 8GB/Core Upgraded Memory

- 600 Cores
- Single Data Rate Infiniband, Gig-E Interconnect
- Linux Red Hat OS
- 36TB Direct-Attached Storage

The *Kaku* platform is shown in Figure 2.

Figure 2. FY10 DHPI *Kaku* system

PCID v9 and ASPIRE were installed on *Kaku*, and transitioned for use by image analysts during 2010. This milestone was documented in the Air Force Research Laboratory *AFRL Technology Milestones* "New Focus on Space Object Identification," dated 15 June 2010. http://www.wpafb.af.mil/news/story.asp?id=123209405

Our image enhancement team has continued to maintain and utilize PCID and ASPIRE to continue to perform R&D for improved performance. The PCID performance improvements over the recent years are summarized in the results to date discussion and significance section of this paper. These packages are also used as described directly above to obtain high-quality images and to provide quick-turnaround high-throughput post-mission processing of an entire space object overhead pass in less than 30 minutes. The increased resolution and quality of imagery, which has been processed with the enhanced PCID algorithm and the FY10 DHPI HPC platform at Maui High Performance Computing Center DoD Supercomputing Resource Center (MHPCC DSRC), is shown in Figure 3.

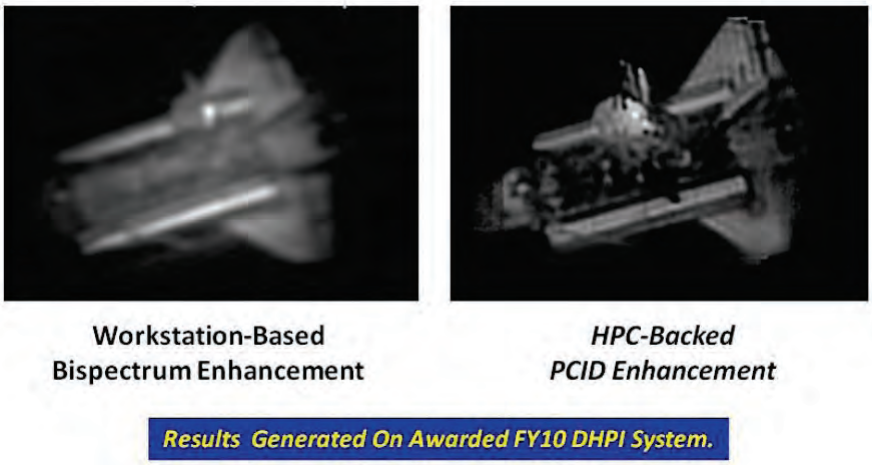


Figure 3. HPC-Backed PCID provides increased image quality

Our team has also made significant developmental progress on the next generation of image enhancement service called the Inline Speckle Imaging Reconstruction Engine (INSPIRE). While we've leveraged a great deal of what we did for PCID/ASPIRE, the INSPIRE software suite supports a fundamentally different concept of operations (CONOPS). ASPIRE was designed for batch post-processing, where the emphasis was on total-throughput. It did not matter how long an individual reconstruction took, as long as the entire pass was processed within 30 minutes. This throughput requirement drove the processing technique to a coarse-grain parallelization approach where individual image reconstructions were performed almost serially. This current CONOPS is depicted in Figure 4.

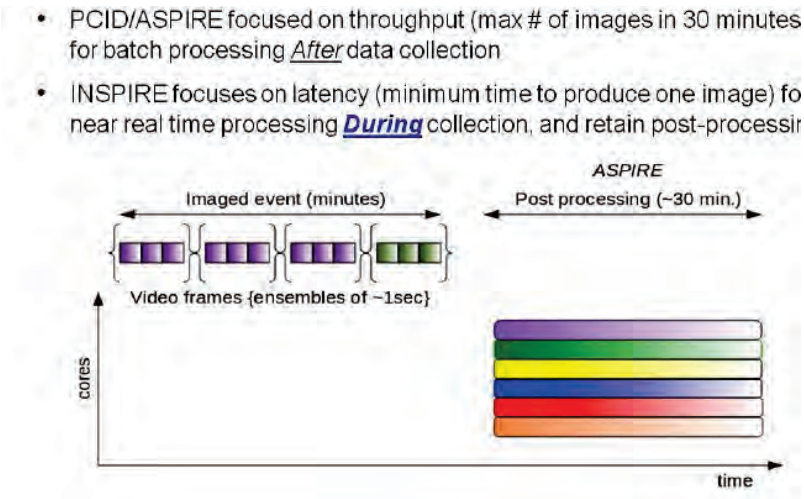


Figure 4. PCID/ASPIRE high-resolution post-processing

PCID and related software utilities have been upgraded with algorithm improvements, code streamlining, using faster minimizers and optimized math libraries, improved parallelization and improved calibration codes. The software has been accelerated to the point where we can change the concept of operations to low-latency, near-real-time (NRT) processing. INSPIRE is being developed to provide low-latency, high-resolution NRT processing so we can process images—as they flow from the sensor. To achieve this, we are pushing on multiple levels of parallelism. We have determined that NRT processing is best-suited to multiple instances of parallel PCID which spreads frames in the ensemble across nodes to reduce latency to provide the highest processing/resolution and exploitable imagery. The INSPIRE goal is to achieve NRT processing during telescope data collection—with processing pipeline latencies of less than 12 seconds. We use parallelization within the image reconstruction code to drive-down the total processing time on a per-image basis. We then use multiple reconstructors, processing in parallel in "cliques" to keep up with the cadence of the input data rate. The end-result is that the operator on the loop can change parameter settings on-the-fly, and almost process the imagery like one does with movies. The Fawkes/Phoenix near-real-time information management services software that the AFRL Information Directorate has implemented and brought to our collaborative team[3], has also been a key enabler for our significant INSPIRE achievements. As such, INSPIRE combines PCID, PARSEC, and MFBD algorithms and Web service concepts; and also utilizes the Fawkes/Phoenix middleware and Publish, Subscribe, Query (PSQ) schema.

Under INSPIRE processing, if we are able to realize a process latency of less than one second, then all compute cores operate on, and as, a single "clique" of image frames data as illustrated in Figure 5.

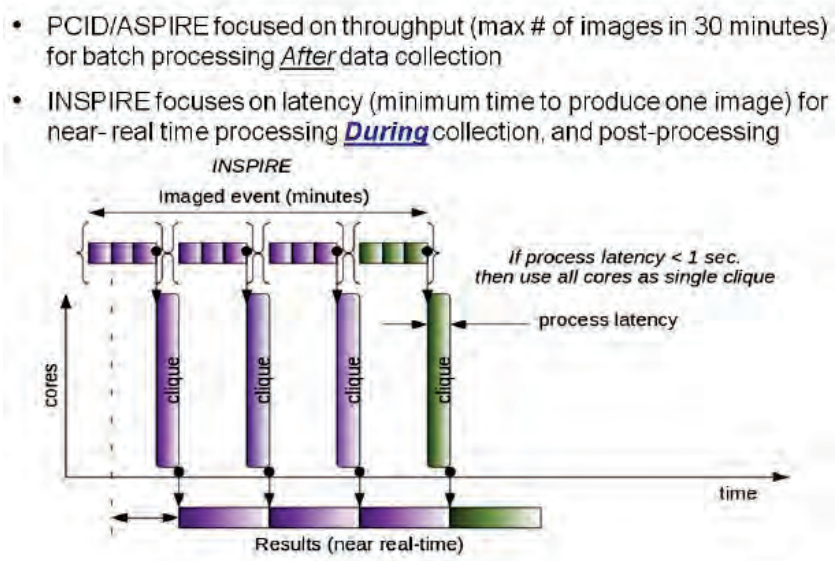


Figure 5. INSPIRE processing for less than one-second latency

If process latency is greater than one-second, then INSPIRE will assign multiple, staggered cliques for compute processing as shown in Figure 6. The ability to perform additional post-processing is still retained.

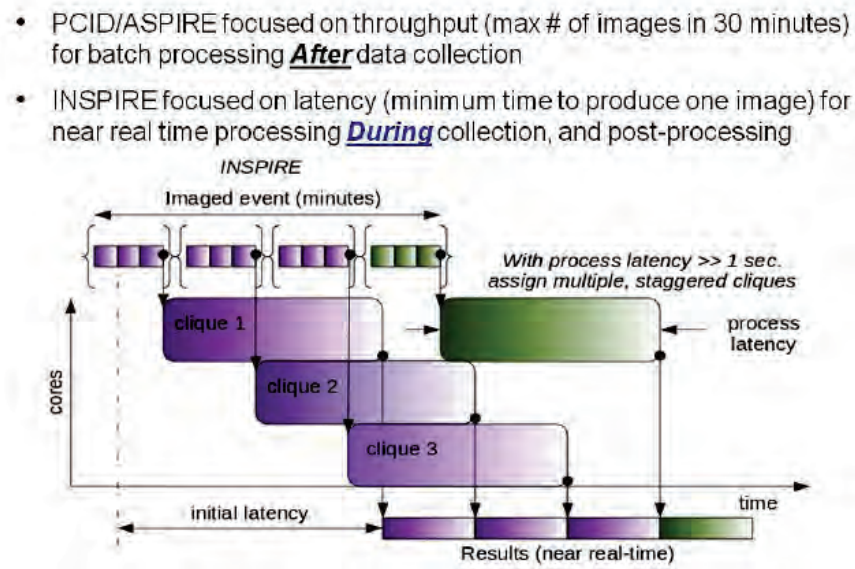


Figure 6. INSPIRE processing for greater than one-second latency

The developmental version of INSPIRE has been demonstrated with the Fawkes communications system several times since calendar year 2011 to various stakeholders that involved the image enhancement algorithm processing time, and the total end-to-end latency. Two particular experiments were performed, and subsequently addressed in papers and presentations[3,4,] that demonstrated system behavior; with details of the processing, communications, and metrics captured. To summarize, for these experiments 20 ensembles of imaging data were input into the system. One node was used for executing the clique-coordinator software (CC) and four clique-master (CMs). Two additional nodes with 8 cores each were used to implement four processor cliques, with each clique having 4 cores. Each of the 4 cliques was assigned to a respective CM. The experiments were performed at the Maui High Performance Computing Center on the *Mana* dual-quad Xeon Nehalem HPC system, with 24GB of RAM on each node. PCID has been optimized to run in parallel, with runtimes reduced from hours to seconds. Overall timing results for the 20 Ensemble/2 (every two-seconds, one of four CMs launched a PCID run on its respective back-nodes) second experiment showed overall end-to-end processing timing of 4.93 to 8.57 seconds. INSPIRE demonstrations and experiments have also included use of a Web-based thin client INSPIRE Viewing Utility (IVU) capability that was developed. The INSPIRE capability has also been executed and demonstrated on the AFRL/RI heterogeneous supercomputer, *Condor*. This platform includes 78 Intel Xeon X5650 dual-socket hexa-core processors; 1,716 Sony Playstation III® (PS3) game console processors; and 168 General-Purpose Graphical Processing Units (GPGPUs).[5].

The MSSS modernization and upgrades previously discussed in this paper, which have occurred in calendar years 2011 and 2012, included new/upgraded infrastructure and sensors, including the 1.6 meter telescope new "*FLASH*" sensor and camera, which provides imaging and Wide Field-of-View (FOV) for multiple development and mission applications. This camera provides several new benefits that include using Electron-Multiplying Charge Coupled Devices (EMCCD) for increased gain and lower readout noise. It operates at 35 frames-per-second for full-resolution, but can perform up to 549 frames-per-second. Modifications to PCID, ASPIRE, and another image enhancement algorithm, Parallel Bispectrum (PARSEC) were needed, as many new data headers in the camera sensor information files were not the same as previous MSSS sensors. The new MSSS *FLASH* sensor uses more of the instrument array as the FOV is increased so that sampling at larger fields of view is correct. This results in *FLASH* frames that vary from 128×128 pixels up to 512×512 pixels. Each pixel is 16×16µm with a maximum readout rate of 10 MHz. Many tasks were identified regarding the ASPIRE, PCID, and PARSEC software engineering that needed to be performed on the image enhancement calibration, ensemble-generation, algorithms, and other utilities modifications needed. These items are addressed further below.

3. Image Enhancement Applications and Support for MSSS Modernization

As introduced in the previous section above, a review of calibration, ensemble-generation (ensembles of camera picture frames), and modification of algorithms and other software utilities work was needed and performed as outlined below.

- Review *FLASH* sensor and calibration process to Flexible-Image Transmission (FITS) files
  - Review subtracted background level values written to header for data debias
  - Modify PCID, PARSEC as needed, obtain proper read noise values for each sensor, calculate ideal threshold values for display, and develop PSF initial setting for PCID for all FOV
  - Develop code within ASPIRE to reduce the calibration data and calibrate FLASH data files, and to convert HDF formatted files to FITS
  - Modify the ensemble generator and ASPIRE, as needed to function properly with *FLASH* and SOR data
    - PCID and Parallel Bispectrum (PARSEC) codes require adding the photon count of the bias removed from each frame to the header for that frame for the new *FLASH* sensor. The read noise for the sensors is also needed.
  - Correctly read exposure time from header so ensembles can be formed using seconds
  - Remove the derotate step when unnecessary
  - Allow embed and no star option for PARSEC when separate ensembles are formed (derotate on) copy all data headers to separate ensembles
  - Adjust run-times for recoveries and batch job submittals
  - Ensure that calibrated FITS files are allowable entries for object and star data, and that the correct sensor type is read from FITS headers
  - Ensure that the Q data parameter (scaling parameter based upon sampling rate of the data and value used to scale all Fourier domain quantities) is correctly set in the namelist.
- Review PARSEC and PCID submit pages for efficient and valid processing for these sensors
- Create new release versions PCID v11.0 and PARSEC v8.0 (the new PARSEC version will add an embed function to handle clipped frames and/or large object data) for *FLASH* sensor data, regression test and install on *Kaku* ensuring GEMINI and AEOS data reduction is still valid
- Ensure the data recovery movie viewer works correctly with various sensor sizes with and without embed, ensure recoveries display correctly in ASPIRE, where analysts do not have any way to change the range of the display, and keep recovery integrity by not suppressing fine detail of the recoveries
- Ensure FITS-to-Tagged-Image File Format (TIFF) conversion utility works correctly to write all required parameters to TIFF output files, and that all header tags pass through calibration to TIFF-PL data.

4. Results to Date Discussion and Significance

Significant improvements to image processing enhancement have been realized over the past several years. The PCID software has realized more than a 10,000-times speedup compared to the original R&D code, and a speedup of greater than 72-times has been achieved between calendar years 2008–2011[6]. Example performance speedup metrics information is shown in Figure 7.

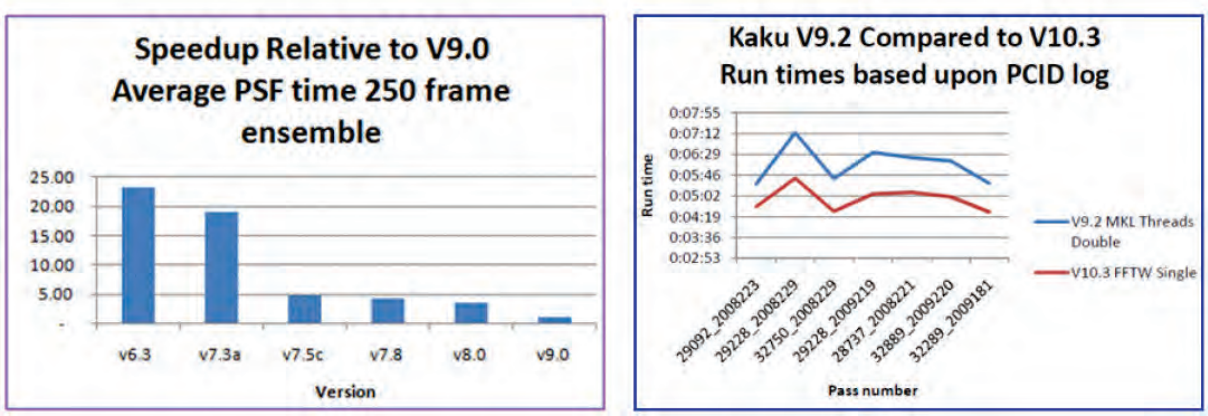


Figure 7. PCID v9 and PCID v10 optimizations results

PCID v11.0 and PARSEC v8.0 development was completed by April 2012. PCID v11.0 is comprised of 11,970 source lines of FORTRAN, C, and other files. PARSEC v8.0 is comprised of 3,463 source lines of FORTRAN, C, and other files. The PCID v11.0 image processing regression testing performance using standard data sets provides equal or better image recoveries than the previous PCID version. PCID v10 provided a 20% speedup in code execution and PCID v11.0 provides more than a 10% additional speedup over the PCID v10 version. The team ported all software to *Kaku* and completed regression testing on this DHPI platform environment. The PCID automatic calculations for object support, filter cutoff, point-spread function support, and other items were verified, and adjustments made as necessary to these parameters to deliver the best overall recoveries using the automatic parameters.

MSSS modernization support tasks were performed so that this latest version of PCID further improves the efficiency of the code, as noted above, while providing optimization for the new *FLASH* sensor at MSSS. ASPIRE was also updated to include a "no star" option, if reference star calibration is not needed or viable. ASPIRE, PCID, and PARSEC ensemble generator and parameters selection software modifications and new developments to process calibration data, calibrate FLASH data files, and convert HDF formatted files to FITS were successful. Work also involved analyzing and understanding HDF5 file format and array dimension sizes and order, and the HDF5 data constructor. Code modifications were also performed to the validate file Common Gateway Interface (CGI) script code component to account for the various new data types so that the software can flexibly read the header of FITS files to set the sensor in the validate CGI software.

Our latest image enhancement software provides significant capabilities for yet further improved delivery of high-resolution products. We would also like to continue development, improvement, and demonstration of INSPIRE, which has the capability to run two image processing algorithms simultaneously, as priorities and funding allows. A screenshot from previous INSPIRE demonstrations that were addressed previously in this paper is shown in Figure 8.

Figure 8. INSPIRE Viewing Utility and INSPIRE

Additional INSPIRE work includes; developing a robust, user-ready application for missions and hardening the existing algorithms, improving the integration and documentation, and providing more customer-focused demonstrations.

Further development also includes incorporating a richer feature set for an updated and New INSPIRE Viewing Utility (NiVU) Web-based thin-client for remote viewing, management, and integration with databases for archiving appropriate input and output data. NiVU utilizes Google Window Toolkit (GWT) open-source development software development kit for building and optimizing complex browser-based applications. GWT provides a set of core Java APIs and Widgets that allow developers to write Asynchronous JavaScript and XML (AJAX) Web applications in Java and then compile the source to highly-optimized JavaScript that runs across all browsers. The foregoing INSPIRE software items will then be available for porting the updated software to the AMOS *Kaku* system, then to the upcoming *'Lilikoi'* platform. Other user interfaces, cross-platform application framework technologies, and management tools may also be evaluated and implemented.

5. Conclusion

Our team has successfully updated image enhancement software for MSSS modernization that is available for use. We continue to provide improved SSA technologies and capabilities that leverage HPC. We anticipate the arrival of the new FY12 DHPI HPC platform, *Lilikoi*, during the third quarter of calendar year 2012, when our team plans to prepare for, and port, image enhancement software and other utilities to the new platform for continued R&D and availability for mission support. We would also like to continue to progress further towards providing even more capable SSA HPC Service-Oriented Architecture (SOA) services and end-to-end NRT processing and demonstrations, as may be directed by the government.

Acknowledgements

The authors would like to acknowledge and thank the AFRL/RD and the AFRL/RI for the direction, execution, and sponsorship of this work.

References

1. Matson, Charles, Charles Beckner, Kathy Borelli, Tom Soo Hoo, Brandoch Calef, Maria Murphy, and Ron Viloria; "PCID and ASPIRE 2.0 – The Next generation AMOS Image Processing Environment." *AMOS 2007 Technical Conference*, September 2007

2. Matson, Charles, Kathy Borelli, Stuart Jefferies, Charles Beckner, E. Keith Hege, and Michael Lloyd-Hart; "A Fast and Optimal Multi-Frame Blind Deconvolution Algorithm for High-Resolution Ground-Based Imaging of Space Objects." *Journal of Applied Optics, Optical Society of America*, January 2009

3. Spetka, Scott and George Ramseyer, "Fawkes Information Management for Space Situational Awareness." *AMOS 2010 Technical Conference*, September 2010

4. Spetka, Scott and George Ramseyer, "Using a Heterogeneous Interactive HPC to Enhance Streaming Images." *DoD High Performance Computing Modernization Program 2011 Users Group Conference*, June 2011

5. Barnell, Mark, "Integration, Development and Results of the 500 TeraFlop Heterogeneous Cluster (*Condor*)." *2011 HPC Users Forum*, September 2011

6. Duncan, Bruce and Chris Sabol, "Advanced Electro-Optical Image Enhancement with Fawkes Information Management for Near-Real-Time Space Situational Awareness." *DoD High Performance Computing Modernization Program 2011 Users Group Conference*, June 2011.

High-Fidelity Analysis in Support of the HIFiRE-1 and -2 Programs

F.J. Malo-Molina
US Air Force Research Laboratory (AFRL/RB), Wright-Patterson AFB, OH
faure.malo-molina@wpafb.af.mil

D. Risha
US Air Force Research Laboratory (AFRL/RZ), Wright-Patterson AFB, OH
daniel.risha@wpafb.af.mil

Houshang B. Ebrahimi
Flow Modeling and Simulations, LLC, Carson City, NV
houshang.ebrahimi.ctr@wpafb.af.mil

R. Yentsch and D.V. Gaitonde
Ohio State University, Columbus, OH
gaitonde.3@osu.edu, robertyentsch@gmail.com

Abstract

Challenge Project resources have fostered important progress in understanding some of the fundamental phenomena that support the HIFiRE-1 and -2 programs. These analyses have been integrally connected to extensive ground and post-flight test data. Considering both the internal and external aero-thermo propulsive phenomena advances this body of work from the last three years. The HIFiRE-1 flight test was successfully analyzed and compared to the computation for several points on the flight path. The modeling and simulations cover an array of cases essential to the program's success, such as transition to turbulence and shock/turbulent boundary-layer interactions.

For the first case, we extend our previous research on different fueling schemes to explore the unsteady and combustion instability effects of the scramjet flow path of the HIFiRE-2. Results show that the heat transfer through the isolator and combustor walls, primarily at lower Mach numbers, has an impact on energy loss, isolator shock, and stream thrust. The Challenge efforts have supported the HIFiRE-2, Medium Scale Critical Components (MSCC), and influenced the X-51 experimental and flight test program. Several highly-scalable codes are employed for this analysis, including GASP, CFD++ and HEAT3D. Combustion is modeled using a finite-rate, ethylene/methane-air chemistry model. These results provide insight into the stability and transition of the scramjet combustor, and into the flow environment, while yielding crucial data on post-processing of flight measurements, as well as the influence of modeling parameters.

1. Introduction

This Challenge Project exploits parallel computational capabilities of the DoD HPCMP to provide a significant computational element to an international program known as HIFiRE (HF), administered by the US Air Force and Australian Defense Science and Technology Organization (DSTO), with NASA support. The recent pioneering success of the X-51 flight test has triggered substantial interest in making such flight routine. To realize this vision; however, there is a critical need to generate enabling insight into the physics of the accompanying fluid, combustion and structural interactions. The HF program provides a unique opportunity to obtain such understanding because of its emphasis on scientific exploration rather than on engineering demonstration. This project leverages a wide range of high-fidelity simulations to extract as much insight and predictability from difficult and expensive ground/flight tests. We seek a clearer understanding of the manner in which shock boundary-layer interactions and/or combustion dynamics interact to adversely impact thermal load, drag, flow distortion and thrust generation (examples of a supersonic injection shown in Figures 1 and 2). Upon successful conclusion, the results of this effort will guide vehicle evolution to mitigate each of these complex factors to achieve routine sustained hypersonic flight.

As in previous efforts, we have extensively explored the broad phenomena encountered in high-speed propulsion devices under different conditions for the HF-1 and -2. The HF-1 post-flight test was one of our primary objectives, and for these simulations, the far-field was conditioned at several altitudes taken from the BET and BEA. For simplicity, most simulations assumed perfect gas air, since the highest measured temperatures were near 500K. Although a complex

chemical dissociation model for air is unnecessary under these conditions, vibrational relaxation might exist, therefore, one case was performed using Park's 5 species, non-ionization model.[1]

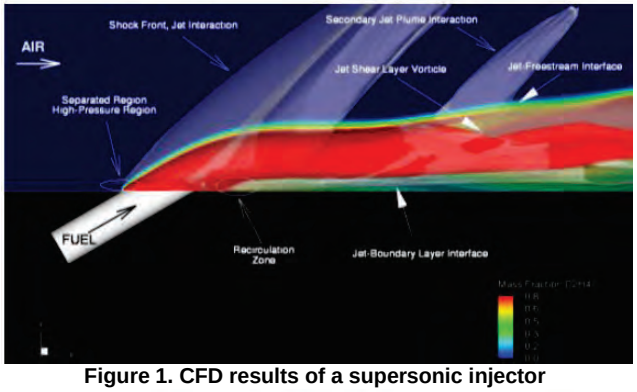


Figure 1. CFD results of a supersonic injector

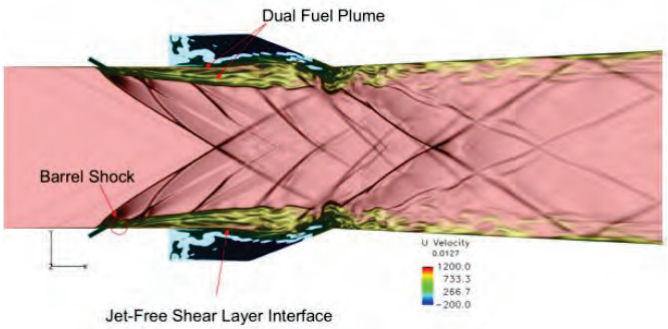


Figure 2. CFD results of a scramjet combustor

This Challenge played a primary role in the scramjet engine development for HF-2 and fueling scheduling, by providing the necessary modeling and simulation support in coincidence to the ground (experimental) and flight test analyses. Chemical reacting and non-reacting assumptions in supersonic combustors were compared during a starting situation and with constant fueling. In earlier work on fully-coupled geometries, shock/boundary-layer interactions encountered in the inlet resulted in a slight increase in the mixing efficiency; however, higher distortion upstream and larger shear separation also increased the tendency to unstart the engine. In the present effort, we consider a ground test situation with uniform flow entering the combustor region. In this section of the HF support, distortions associated with the inlet are ignored and the focus is on complex mixing phenomena, non-equilibrium transfer of turbulence energy, and interactions between turbulence and chemical kinetics that may impact both the chemical reactions and turbulence field downstream of the system.

Additional calculations complement the design of components and systems to support the flight and experimental tests. This portion of the work focuses on material thermal conductivity, thermal barrier-wall coating techniques (TBC), and active cooling using fuel and/or water on several components during testing. A detailed example of these computational efforts is described herein on the material sensitivities between the experimental and flight hardware (in terms of physics, operation, and performance).

1.1 HF-1 Post-Flight Test Support

Figure 3a illustrates the HF-1 payload with a full-scale length of 1.916 m (75.45 in) and a maximum outer diameter of 0.3556 m (14.00 in). The front cone (shown in green) has a 7° half-angle with a nose radius of 2.94 mm (0.098 in). The cylindrical section following the front cone (shown in yellow), serves as a separation zone to isolate the primary and secondary experiments. At the rear of the payload is a flared shoulder (shown in blue) that triggers the Shock-Wall Boundary-Layer Interaction (SWBLI) experiment. Set 180° apart into the shoulder are two Tunable Diode Laser Absorption Spectroscopy (TDLAS) test channels (shown in tan). Beyond that, the Orion-Terrier sounding rocket that provides thrust to power the payload through the flight envelope joins smoothly to the flare (not shown). The second-stage booster does not separate from the payload during the ballistic portion of the trajectory, and this is not modeled in the simulation. The vehicle is instrumented to measure pressure and surface heat loads for transition on the fore-body cone

(primary experiment) and shock/boundary-layer interaction on the cylinder flare (secondary experiment). For both of these, significant ground tests, analytical investigations, and computational research have been performed.[2,3]

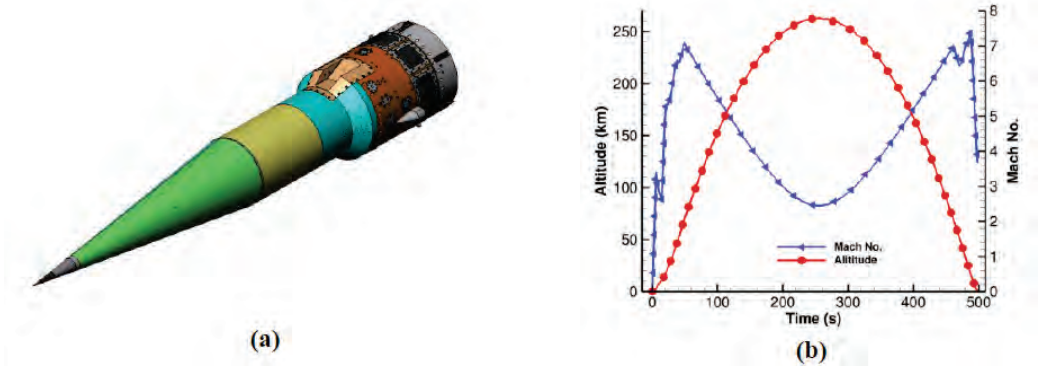


Figure 3. HF-1 payload and BET for Mach number and altitude vs time

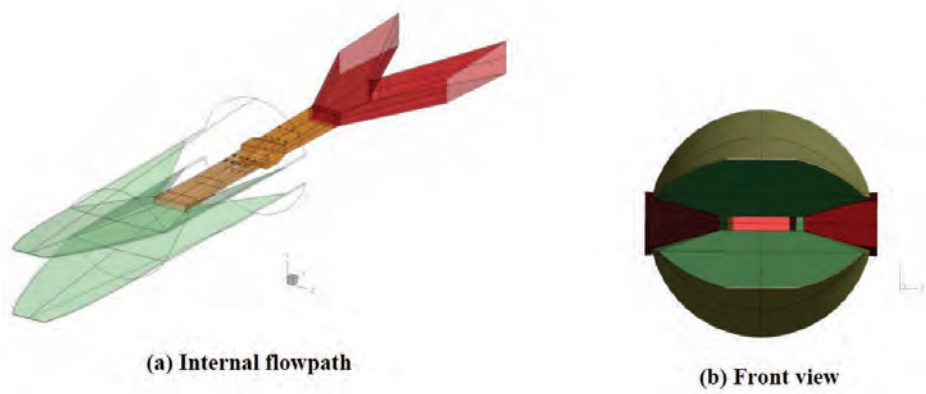


Figure 4. HF-2 forebody/payload configuration (not to scale)

The best estimate trajectory (BET) flight envelope of the HF-1, as shown in Figure 3b, was derived from telemetry data.[4,5] Several different stages can be identified in the flight path. The Terrier stage fires from approximately t=0-6 seconds, followed by a coasting period where the spent first-stage is drag-separated. At approximately t=15 seconds, the Orion stage ignites, causing the vehicle to accelerate until approximately t=43 seconds. It then coasts through its turning maneuver and begins its ballistic trajectory through the atmosphere, ending with a hard landing at approximately t=500 seconds. Unexpectedly-high angles-of-attack have rendered the descent less amenable to analysis (Reference 4).

Best estimate atmospheric (BEA) data has also been provided for the entire flight path from the Air Force Weather Agency. A flight test point of t=15 seconds was chosen as the simulation point of interest, due to subsequent telemetry issues with the TDLAS data. At this selected point in the flight, the altitude (h) is 11.1 km and the static pressure and temperature are equal to 23.3 kPa and 226K, respectively. The Mach number (M=2.63) is low enough for chemistry effects to be considered negligible, yet high enough for the aerodynamic phenomena of interest to be considered fully-developed. This point is also of interest because it is at the end of a coasting period between the firing of the first- and second-stages. This was intended as a lower "noise" environment to better highlight the subtleties of the measurement data. Subsequent to the choice of this flight point, analysis performed by Kimmel, et al.,[6] of the flight test data indicated that a laminar-to-turbulent transition was likely occurring at this time. Furthermore, heat transfer rates achieved their lowest values in the flight test, increasing noise in the experimental measurements and complicating the comparison, as discussed later.

1.2 HF-2 Ground Test Support

One of the major challenges for scramjet designers is to properly mix the high-speed air with the fuel to induce efficient and rapid combustion, since the residence time in the device is of the order of milliseconds, and equivalent to the chemical time. An additional complication is presented by the flow-field distortions in the inlet, which have a substantial and complex impact on combustor performance. Entropy generation, resulting in pressure losses, can also engender the

combustion efficiency, and thus performance, under certain conditions. During this Challenge, many of the key phenomena encountered in high-speed propulsion devices have been examined.[7,8] Among these are the effects of injector configuration in a rectangular cross-section combustor with a single-cavity flame-holder. Although non-reacting assumptions can provide valuable comparisons in parametric analysis and sensitivities in supersonic combustors, they could wrongly predict the mixing efficiency, depending on the given geometry. Thus, our analyses were extended to reactive finite-rate conditions.

The hypersonic vehicle design is depicted in Figure 6. The HF-2 consists of an inward-turning inlet coupled to a rectangular cross-section combustor. The configuration exhibits vertical and horizontal symmetry, and the inlet conditions the air from its on-design flight to yield temperatures at auto-ignition levels at the combustor entrance. A simplified dual-split nozzle is designed to exhaust the flow downstream of the payload, away from the flight vehicle. Figure 6 shows the analyzed geometry of the HF-2 payload as follows: 1) the metallic yellow delineates the outer surfaces of the un-shrouded inlet with its slightly blunt lips, 2) the interior walls of the inlet are colored green, 3) the internal inlet section and isolator region are in yellow, 4) the burner section is colored orange, and 5) the nozzle is portrayed in red.

Several additional objectives were evaluated during this portion of the simulation. A series of analyses evaluated the transient fueling effects in the combustor during a point at the starting process, as in the flight test. Two additional objectives aimed to determine the sensitivity of the isolator-combustor-nozzle physics, operation, and performance as a function of wall-thermal conductivity and Mach number. The primary objective consisted of 25 reacting simulations each requiring approximately ~22 Khrs of computational resources. The secondary objective considered the sensitivity, operation, and performance of 5 flame-holder materials with different thermal conductivities during 5 different operational Mach numbers (see Table 2).

2. Analysis and Numerical Models

2.1 HF-1 Post-Flight Support

For most of the HF-1 simulations, the chemistry was fixed to perfect gas for simplicity. Since the highest-temperatures observed in the simulation are near 500K, the complexity of a dissociation model is unnecessary. However, since vibrational relaxation is possible, a simulation was also performed using Park's 5 species, non-ionization model[9] for comparison. Surface temperatures were specified using best-estimate surface temperature data provided by Dr. Roger Kimmel of US Air Force Research Laboratory, Air Vehicles Directorate (AFRL/RB) based on robust windowed averaging of raw experimental data. Although these temperatures are only applied in a one-dimensional (1D) sense (along the x-axis), based on previous results,[10] it is expected that the error in heat transfer associated with the specified surface temperature will be small relative to the error due to the turbulence model.

During our initial assessment, *Cart3D* was used to give a baseline assessment of the wave structure for further examination of the effects of viscosity and turbulence modeling. Primarily intended for rapid aerodynamic design, *Cart3D* is a NASA code package that facilitates high-fidelity, inviscid simulations via highly-optimized parallel code and automated mesh generation and adaptation. *Cart3D* is a highly-integrated package that requires only a watertight surface definition and basic parameters such as Mach number and angles-of-attack as inputs, and can typically simulate even highly-complex geometries on a workstation computer. The viscous simulations were performed with *GASPv5*. Several simple simulations were performed to ensure that the code parameter specification was correct. The Roe flux scheme with a *WENO* (r=2) limiter, used previously in other simulations in *GASPv5*, was employed for its robust and non-dissipative properties. For turbulence modeling, the Wilcox (2006) $k-\omega$ model was invoked with a compressibility correction to account for mixing effects in supersonic shear layers.[11]

The complexities of the flow, particularly the large pressure gradients that cause separation regions, along with uncertainties like free-stream turbulence intensity in the flight conditions, necessitate the adjustment of turbulence model parameters. Turbulence models were calibrated to better match the measurements inside the channel by:

1. Specifying the point of laminar-to-turbulent transition. Adjustments were made to this parameter to capture both the transition region along the cone (as indicated by the experimental heat transfer data) and the laminar separation point in front of the flare (as indicated by the experimental surface pressure data).
2. Limiting the ratio of the eddy-to-laminar-viscosity (the only value from the turbulence model that directly affects the Navier-Stokes equations). Changes made to this parameter primarily influenced the laminar separation region at the flare. Finally, the turbulent kinetic energy of the free-stream was adjustable.

The structured, multi-block grid was prepared with approximately 17 million cells used to form the finest grid. Mesh resolution effects are characterized by considering sequenced medium- and coarse-grids having 2.1 million and 264 thousand grid-points, respectively. Although important in evaluation of the descending trajectory, angle-of-attack effects were not considered in this work. Therefore, the model used quarter symmetry with a singular axis convergence at the tip as shown in Figure 5.

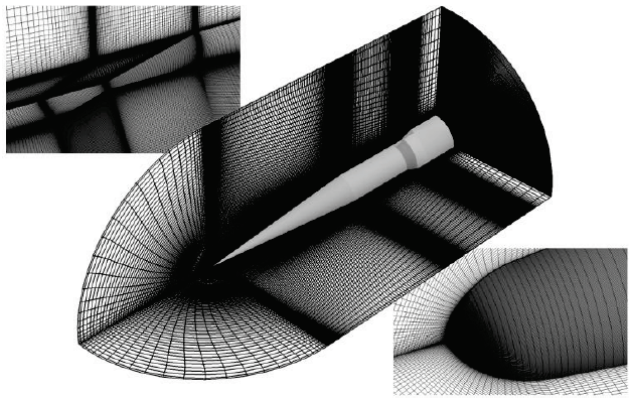


Figure 5. HF-1 1/4th symmetric structure fine-grid

2.2 Unsteady, Fueling Scramjet Analysis of HF-2

For this study, only one geometry is analyzed. It is taken from a previous parametric assessment where: 1) the injection location is away from the cavity-step, and 2) a partially-closed cavity configuration is investigated to examine the effects of air-fuel mixing, penetration, and overall performance. The gas mixture inside the cavity is not premixed, and is similar to that used in previous research. The injection port is neither aligned nor staggered with the structure of the main recirculation zone, but rather injected prior to the step.[12,13] As in previous work, these cases are anticipated to yield a combustion zone within the cavity region and against the subsequent expansion wall. The studied configuration is a medium-sized design cavity with a 30° open cavity and a L/D (length/depth) ratio equal to 5 (D=1 in=0.0254 m) used as a flame-holder, as shown in Figure 6. As described in the literature, the L/D is a very important factor for proper mixing and achieving an optimum stable combustor under these flight conditions.[14,15,16] Studies conducted using supersonic scramjets have concluded that open cavities result in a smaller drag penalty.[17] The longest cavities (L/D>10) produce vortex shedding, creating cavity oscillations and unstable flames. On the other hand, the shortest cavities do not produce vortex shedding issues, but lack sufficient air entrainment needed to sustain the combustion process, and could also induce instability and thermal choking. The fuel is injected through a circular port having a diameter of 0.00381 m, centered at 3D upstream of the step and angled at 30°. The uniform inflow conditions are fixed at the entrance, 2L upstream of the step with a height of 1L. The symmetry condition is implemented at both sides (z_{min} and z_{max}) of the study configurations and to the upper-wall opposite the cavity (y_{max}). For both cases, the cavity is followed by a ~3° expansion from the bottom-wall towards the exit plane, where a first-order extrapolation is implemented. The overall dimensions of these geometries are about 7×1L, where the combustor might exit into the nozzle section.

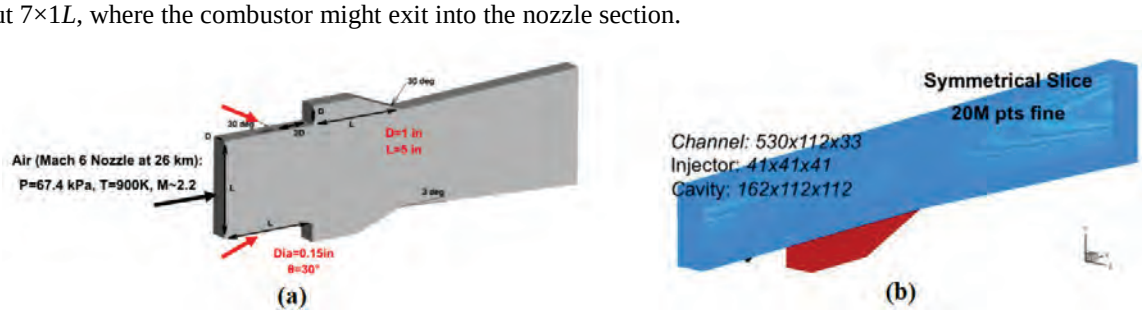


Figure 6. Analogous supersonic combustor a) configuration, and b) structured-grid

Figure 6b shows the structured multi-zone grid with a subset of the mesh density for clarity in the combustor region. The configurations are divided as colored into three individual zones: 1) the channel with 530×112×33, 2) the fuel injector with 41×41×41, and 3) the cavity with 162×112×112. This fine-grid (~20M pts), with an average grid-sized cell-area of 7×10^{-10} m^2, yields a y^+<0.5 under the assessed conditions. As shown in Figure 6b, the points are distributed to resolve the

flow structure near the walls, in the cavity, and around the fuel injectors. This structured body-conforming mesh is employed with emphasis on ease-of-generation, and limited to just a slice or portion of a symmetrical array of injectors. The mesh is clustered, and also angled towards the outer-walls and around the injectors to capture the proper flow-physics and their interactions (shock-to-shock/shear layer). For these cases, the inlet employs a fixed-uniform-inflow condition. The outflow boundary condition at the combustor exit is modeled using first-order extrapolation, and no-slip adiabatic wall conditions are applied on all solid walls.

The Reynolds-averaged Navier-Stokes (RANS) approach with $k-\omega$. Wilcox closure is adopted.[18] Previous efforts have indicated that for the class of flows under current examination, the RANS approach may capture some of the key trends, though details of vortex shear layer interactions in the vicinity of the cavity are not adequately reproduced.[19,20] The code employed in this work is HEAT3D, which has previously been employed to solve a variety of supersonic and subsonic combustion problems.[21] Extensive validation of HEAT3D treating fully-coupled, three-dimensional (3D) flows with finite-rate chemistry has been reported in previous studies.[22,23] The code uses an implicit-time integration and strongly conservative finite-volume formulation. Combustion is modeled using a finite-rate, Ethylene chemistry model and Gordon/McBride thermodynamic curve fits. Unlike previous years, this model was further reduced to 11 gaseous chemical species (C_2H_4, C_2H_2, CO_2, CO, OH, O_2, O, H_2, H, H_2O, and N_2) with 17 chemical reactions for reactants, and products for Ethylene fuel and air oxidizer (see Table 1).

Table 1. Chemical mechanism for: fuel (Ethylene) and air mixture

Reaction 1:	$C_2H_4 \leftrightarrow C_2H_2 + H_2$	Reaction10:	$H + OH + M \leftrightarrow H_2O+M$
Reaction 2:	$C_2H_2 + O_2 \leftrightarrow 2CO + H_2$	Reaction 11:	$2H + M \leftrightarrow H_2 + M$
Reaction 3:	$CO + O + M \leftrightarrow CO_2 + M$	Reaction 12:	$H + O + M \leftrightarrow OH+M$
Reaction 4:	$CO + O_2 \leftrightarrow CO_2 + O$	Reaction 13:	$2O + M \leftrightarrow O_2+M$
Reaction 5:	$CO + OH \leftrightarrow CO_2 + H$	Reaction 14:	$H_2 + O_2 \leftrightarrow 2OH$
Reaction 6:	$OH + H_2 \leftrightarrow H_2O + H$	Reaction 15:	$2H + H_2 \leftrightarrow 2H_2$
Reaction 7:	$O + OH \leftrightarrow O_2 + H$	Reaction 16:	$2H + H_2O \leftrightarrow H_2+ H_2O$
Reaction 8:	$O + H_2 \leftrightarrow OH + H$	Reaction 17:	$2H + CO_2 \leftrightarrow H_2+ CO_2$
Reaction 9:	$2OH \leftrightarrow O + H_2O$		

The selected gaseous fuel injection condition for the constant fueling case corresponds to a choked flow rate with a fuel temperature of 300K and a density of 2.3 kg/m^3. The frozen and reactive cases with varying injection pressure sustain the same temperature and total mass-flow-rate during the 0.5 msec transient injection process. The main airflow corresponds to a Mach number of 2.2, and a static temperature and pressure of 900K and 67.4 kPa, respectively. This causes the equivalence ratio, Φ, to lower and vary from 0.0 to around 0.8.

2.3 HF-2 Material Analysis

Scramjet engine development is dependent on analytical, computational, experimental, and flight test analyses. The present work focuses on material thermal conductivity, thermal barrier-wall coating techniques (TBC), and active cooling using fuel and/or water. Material sensitivities between the experimental and flight hardware in terms of physics, operation, and performance, are the primary purpose of this study. Results show that the heat transfer through and temperature of the isolator and combustor-walls, primarily at the lower-Mach numbers, impact energy loss, isolator shock, and stream thrust, supporting the HF-2, Medium-Scale Critical Components (MSCC), and the X51 experimental and flight test program. The analysis makes comparisons in terms of I-D average plots, centerline contour plots, and wall contour plots. Trends in percent of kinetic energy loss (% KE loss), percent of heat transfer loss (% HT Loss), percent change in stream thrust (% ΔST), stoichiometric mixing-efficiency (η_{mix}), and enthalpy combustion-efficiency (η_{comb}) were also used to quantify the effects of material thermal conductivity. The selected computational code for all the following simulations was computational fluid dynamics (CFD^{++}), which is capable of conjugated heat transfer. This CFD tool was developed by METACOMP, Inc. The CFD^{++} suite provides several cutting-edge computational fluid dynamics modules and tools, which work seamlessly with existing mesh-generation and post-processing tools across all flow regions and grid topologies. The unified-grid framework (sometimes referred to as a grid-transparent framework) unifies the treatment of different cell shapes and grid topologies so that the flow solver becomes equally applicable to all cell topologies. In addition, its unified-physics treatment capabilities provide the flexibility to solve many combinations of governing equation sets, enabling the same computational framework to handle flows from incompressible to hypersonic, with or without additional equation sets for the modeling of reactions, turbulence, acoustics, etc. CFD^{++} also contains a unified-computing capability that allows total portability between all the different multi-computing platforms at DoD Supercomputing Resource Center (DSRC).

On this section of the work, we employed another finite-rate JP7 ($C_{12}H_{24}$) + Air reduced-chemistry model made of 17 species with 44 reactions. Combustion Science and Engineering, Inc. (CSE) developed this mechanism, which expands our requirements with a variable Prandtl (*Pr*) and Schmidt (*Sc*) capability and a k-e turbulence model with wall sanctions. The solution is advanced with a second-order implicit dual-time-step and a second-order approach in space, operating a min-mod limiter. The walls are computed with a resistive Heat Transfer (T_∞=300 K^0, thickness=1 inch) for the five different thermal conductivities (see Table 2). The outflow region is treated with a supersonic 1st-order extrapolation and the non-premixed fuel is added into the combustor at a T_{fuel}=940° F and a constant $\dot{m}_{fuel}$ for an equivalent of φ=1.0. The grid was made of ~16.7 million unstructured cells, yielding a y^+<10 for a conserved convergence of mass (~0.01 kg/s) and residuals (~0.001).

Table 2 shows the five free-stream conditions taken from the HF-2 flight test trajectory for which five materials are evaluated with different thermal conductivity. Five different cases were evaluated for each of the five flame-holder materials, each using about ~21×10^3 hrs of computational resources. For example, Inconel and copper have thermal conductivities of 9 and 224 Btu/h-ft-F, respectively; with finite-rate chemistry (see Table 2).

Table 2. Free-stream conditions for HF-2 and wall-thermal conductivities for 5 materials

HF6 Cases	Q(psf)	P(psf)	T(R)	AoA (deg)	Mach	Materials	Thermal Conductivity (W/m-K, Btu/h-ft-F)
M4a0	2,000	178	390	0	4	**Inconel**	15, 9
M5a0	2,000	114	390	0	5	**Bm-alloy**	86, 50
M6a0	1,500	60	398	0	6	**Tj-alloy**	186, 108
M6a4	1,500	60	398	4	6	**Gl-alloy**	263, 152
M7a2	1,000	29	406	2	7	**Copper**	388, 224

3. Results

3.1 Comparison of HF-1 Inviscid and Viscous Simulations to Experimental Data

Figure 7a shows the inviscid results from Cart3D on the upper-half. All the expected features of an inviscid simulation are present, including the initial conical shockwave, expansion at the shoulder sections, and complex shock structures with near stagnation at the flare. The insert depicts the grid discretization near the tip, including the unstructured triangular-surface definition and adaptively-refined Cartesian flowfield mesh. Many of the features described by Oberkampf and Bertel[24] are clearly evident as the flow expands around the sharp HF-1 trailing-edge.

The Mach number on the symmetry plane and surface-pressure coefficient distributions for the viscous simulations are compared in Figure 7a, bottom-half. The most striking difference is observed in the vicinity of the flare, where the viscous simulation shows significant upstream influence from the flare. This is expected as a result of the *SWBLI*, which results in a spreading of the interaction region.[25,26] The local effects are similar to those measured by Brown, et al.[27] The highest surface pressures in the viscous simulation are seen away from the cylinder where the shear layer intersects the flare. In the inviscid simulation; however, the comparatively undisturbed high-speed flow strikes the bottom of the flare, producing a high-pressure rise at the bottom, and a shockwave that causes a more uniform pressure rise on the rest of the flare. In the channel, the pressure rise is much more gradual with the viscous simulation because of the substantial upstream influence associated with flow separation.

Figure 7b shows the separation of the boundary-layer through calculated surface oil flows and the magnitude of the gradient of the density (shear layer/shock structures). Acting as a pressure outlet, the separation region moves aft on the plane of the channel, which decreases the upstream influence of the flare. Examination of the shock structure shows the very well-defined shear layer over the separation bubble, as well as a clearly-defined separation shock. Also present is a weaker third shock aligned with the base of the flare. The shear layer presents a flow to the free-stream that resembles a double-flare, and the resulting structures resemble those observed experimentally by Holden, et al.[28] The flow field is significantly more complex in the channel plane than in the lateral plane. “Lip” vortices are evident in the flow exiting the sharp edges of the channel, and their presence can also be noted in the surface oil lines shown in Figure 7b. These structures can be seen in the inviscid simulations as shadows in the Mach and pressure contours as shown in Figure 7a and 7b. The mechanism is similar to the vortex production noted by Rizzi and Eriksson[29] for a solution to the incompressible Euler equations.

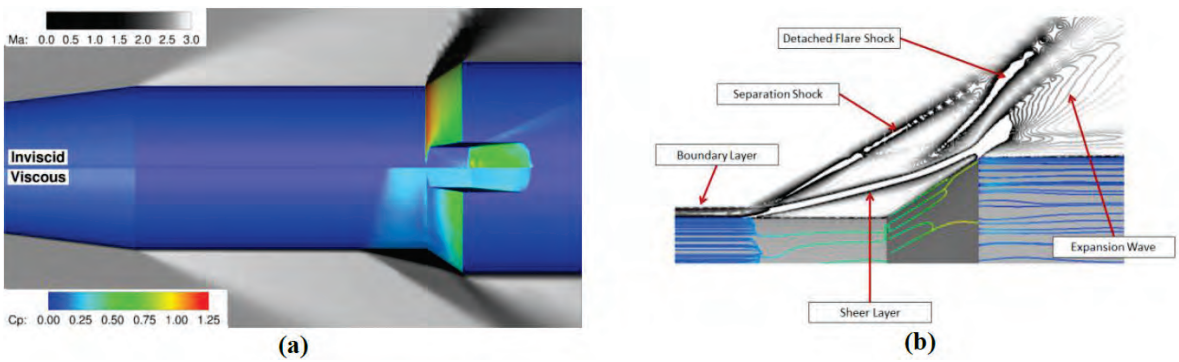


Figure 7. HF-2: (a) Invisicid vs. viscous simulation, and (b) Complex SSLS at flare

Figure 8a compares surface pressures from different simulations to experimental data prepared by Kimmel. The surface profile is aligned to the right y-axis, although it is provided for reference only, as it is not to scale with the x-axis. The experimental data are shown with unconnected X's. The "Unlimited viscosity" case is the baseline turbulent simulation where the free-stream eddy-viscosity (as a ratio of molecular-viscosity) is set to 0.1, and the free-stream turbulence is set to 0.05%. Peak eddy-viscosity ratio values reached slightly over 2,000. The "Low-viscosity" case limits the eddy-viscosity ratio to 500. The third simulation, denoted "Low-viscosity, lowered-Mach number" examines the sensitivity to free-stream Mach number by reducing it from the nominal value of 2.63 by 5%, while keeping the eddy-viscosity ratio at 500. The effect of chemistry is examined through the use of Park's as previously described, and denoted "Low-viscosity, Park's air model" in Figure 8a. Although certain phenomena are still beyond the capabilities of the turbulence model, the agreement between simulation and experiment is good. In all cases, the separation point is predicted further forward than observed in the experiment. This may be due to the incomplete laminar-to-turbulent transition mentioned previously, although grid-effect simulations shown later indicate that a more dissipative numerical scheme may be advantageous. Overall, the unlimited-eddy/laminar viscosity case reproduces the data most accurately. The viscosity ratio case caused the separation point to move forward slightly and further under-predict the pressure on the face of the flare. Reasoning that an error in prediction of the free-stream conditions could lead to over-prediction of separation, the free-stream Mach number for one simulation was lowered by 5%, but this also had no positive effects. Chemistry effects did not influence the pressure distribution, as anticipated. Although adjustment of the forced turbulent transition point did allow for much more accurate realization of heat transfer on the cone (as discussed later), it was ineffective in preventing early separation near the flare. Increases in the free-stream turbulence also had no effect.

Figure 8b compares the heat transfer with experiment for the different numerical simulations on the conical nose (top) and flare (bottom) of the HF-1. Since transition has a substantial impact on heat transfer rates, the turbulent transition point was specified on the cone to coincide with the apparent transition point from the experimental heat transfer data. This gives good agreement overall, both in magnitude and position, with the experimental data on the cone as shown in the upper-half of Figure 8b, although extension of the transition region could perhaps give slightly better agreement. Specification of the turbulent transition point proved crucial, as the unspecified transition case over-predicted the heat transfer in the laminar region of the cone. Near the flare, agreement is much weaker, as shown in the bottom-half of Figure 8b. The experimental data shows higher C_h (heat transferred) values in the separation region than on the face of the flare, counter to expectations.

3.2 Fueling Analysis, (LES)

Figure 9a-b shows the instantaneous contours for the mass-fraction of fuel (C_2H_4) and temperature at the center z-plane, respectively at t=2.4 msec. The vertical-plane shown is positioned at the center of the injector at the z=0.5 inch plane. In addition, the plot traces a group of streamlines from the fuel port out into the main flow field. The shock system produced in this case shows a low-penetration of around 40%. As shown previously (Reference 20), open cavities tend to enhance mixing and sustain reactions as a trap-vortex at the expense of a lower fuel penetration in scramjet combustors. Unlike these earlier cases, the displaced boundary-layer and interactions across the cavity are noticeable further into the flow. The constant-fueling case yields a large-gradient of momentum ratio in a smaller region past the step and downstream from the ramp. The front of the ramp generates a high-pressure region towards the core and past the free-shear layer. Although the mixing is nearly 65% at the exit profile, the fuel penetration past the cavity was determined to be inefficient.

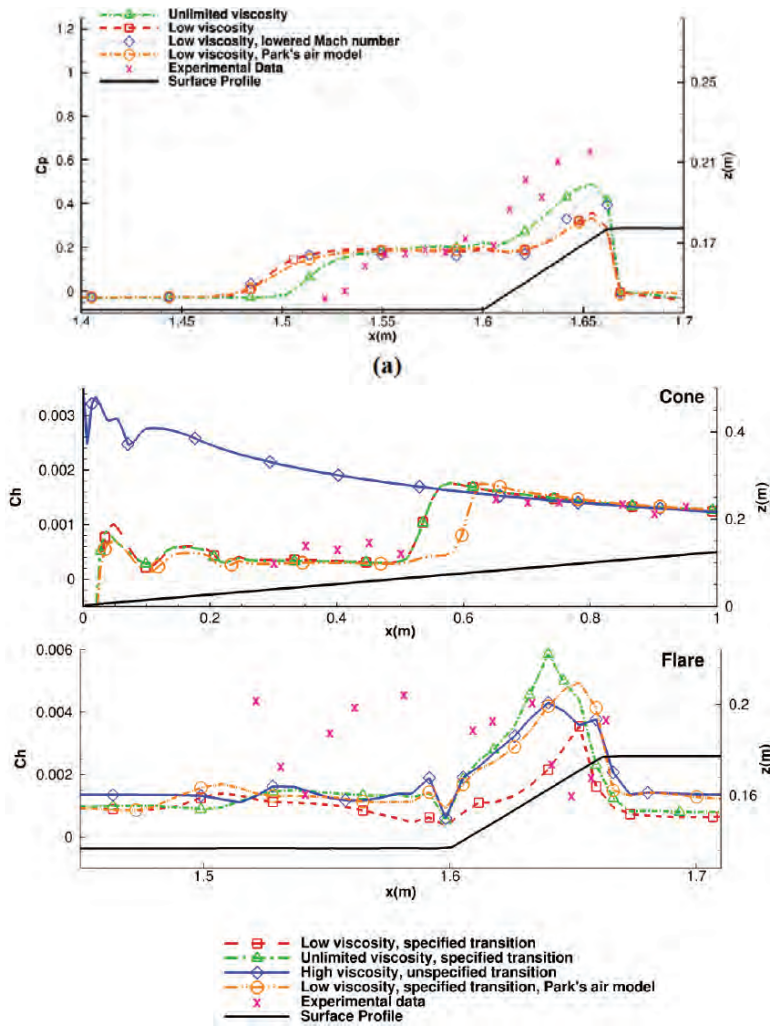


Figure 8. HF-1 Comparison to experimental (a) surface pressure and (b) heat transfer measurements

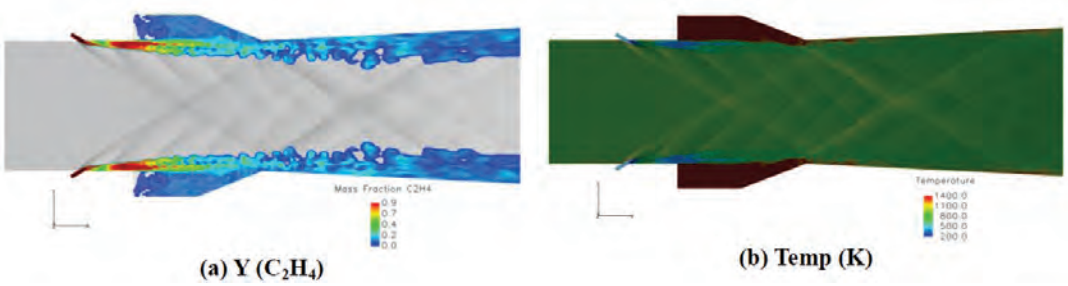


Figure 9. Mach number and temperature contours at the center z-plane, iso-surfaces and streamlines for the frozen case with constant injection at an asymptotic/periodic time of 24 msec

Although the special scales are gridded in the finer-mesh considering the required *Re* values, the employed sgs-subgrid simplifies solving the larger scales by filtering, as in a coarser-grid. Also, this frozen analysis (as shown in Figure 9) largely reduces the computational expense to finite-rate due to the need for a closed formulation using a *PDF* model. This numerical analysis is highly-efficient, thus this methodology is very attractive when analyzing internal flow-paths of complex configurations at high-speed. This simpler stress model assumes that much finer scales are isentropic in nature and slight physical changes will result from the interaction of the injection and free-shear layer downstream to the main flow. Since much finer scales might misrepresent the effects near the wall and around the cavity, the grid is uniformly-refined near this region.

Figure 10 shows ten frames of the fuel mass fraction contours, allowing insight into its concentration and distribution during a rapid injection cycle (pulsed under 0.5 msec of its selected Φ=0.8). The cavity seems to fill rapidly during its entire cycle, while the fuel plume shows a distinctive pair of stalks as in the previous case. One stalk travels over the shear layer, away from the wall towards the core, and disperses with the incoming air, producing double- and triple-eddies downstream (see Figure 10). The second, more differentiated stream of fuel travels along the free-shear layer near the top of the opening. It reaches the rear ramp of the cavity, where it is then lifted by the high-pressure interaction from the wave coinciding with the rear of the cavity-edge. The injection pressure remains constant during the rest of the simulation. The circulation inside the cavity shows larger regions and amplitudes over most of the volume within the free-shear layer, compared to the constant fueling case with more isolated and smaller regions throughout. Figure 10 sheds light on how the trap vortex acts in these actively pulsating cases by indicating the irregularities, circulation intensity and predominant direction of the flow. Unlike constant fueling, 1) larger distortions are produced that persist farther downstream and into the core past the cavity's ramp, and 2) the pulse affects the amount of fuel that penetrates and circulates inside the cavity.

3.3 HF-2 Material Results

In this analysis, the sensitivity of the isolator-combustor-nozzle physics, operation, and performance as a function of wall-thermal conductivity and Mach number are determined. The material analysis objectives are shown below in Figure 11a-d, for the trends in terms of % HT Loss (Heat Transfer, Equation 1) and % ΔST loss (Stream Thrust, Equation 2) versus the Mach number and the different materials' thermo-conductivities. The heat transfer through the walls is extracted from the computational solution obtained from CFD^{++}.

$$diff\,\%\left(HT_{loss}\right) = {}^{HT_{watts}}/_{\left(E_{flow}+HT_{watts}\right)} \tag{1}$$

$$diff\,\%\left(\Delta ST\right) = {}^{\left(ST_{out}+ST_{in}\right)}/_{ST_{in}} \tag{2}$$

Figure 11a is a plot of the thermal conductivity versus the percent of heat transfer through the walls for the five cases considered. In all cases, the percent of heat transfer increases with thermal conductivity and Mach number. For cases m6a0 and m6a4, the increase in mass flow tends to increase the percent of heat transfer through the walls, indicating that the percent of heat transfer in a scramjet isolator-combustor-internal nozzle configuration is a function of material thermal conductivity, Mach number, and mass flow. The plot of thermal conductivity versus the change in stream-thrust shown in Figure 11b shows that the change in stream-thrust is not significantly affected by wall-thermal conductivity for a given Mach number.

Figure 12a-b plots the static pressure (P_s) and temperature (T_s), mass fraction of CO_2 and H_2O on the strut-wall for the m4a0 and m5a0 cases when using: the Inconel, copper and alloy strut materials (top to bottom). The static pressure and temperature indicate that the shock moves toward the throat with the decrease in thermal conductivity (copper to Inconel). These effects were also seen in the above results and provide some insight to the burning characteristics of the cavity as well. Based on these simulations, the flame-holding material has minimal impact on combustor performance.

4. Summary

Multiple computational investigations of key aero-propulsive phenomena encountered in hypersonic vehicles were accomplished over this three-year long Challenge Project. Under the current efforts, a chosen point along the flight-path of the HF-1 vehicle was investigated numerically and compared with experimental results, using *Cart3D* (an in-viscid, rapid analysis tool) and *GASPv5* (a high-resolution RANS simulation tool). The phenomena explored included the flow over fore-body cone, *SWBLI*, and channel flow on the HF-1. Agreement between the simulations and experimental data on surface pressure and heat transfer on the fore-body cone was good; however, discrepancies are evident in heat transfer predictions in the separation region. Some of these issues are associated with the limitations of the turbulence model. However, the specific point in the trajectory chosen for simulation is undergoing laminar-to-turbulent transition, causing it to exhibit an upstream propagation of high-heat-transfer values that does not appear at points in the trajectory prior to or after the chosen point. Another point of the trajectory not in transition was analyzed, and showed better agreement with the experimental heat transfer data. Investigation of other points in the ascending trajectory is currently underway and will be presented alongside simulations of the descending trajectory, with its associated high-angles-of-attack in subsequent publications.

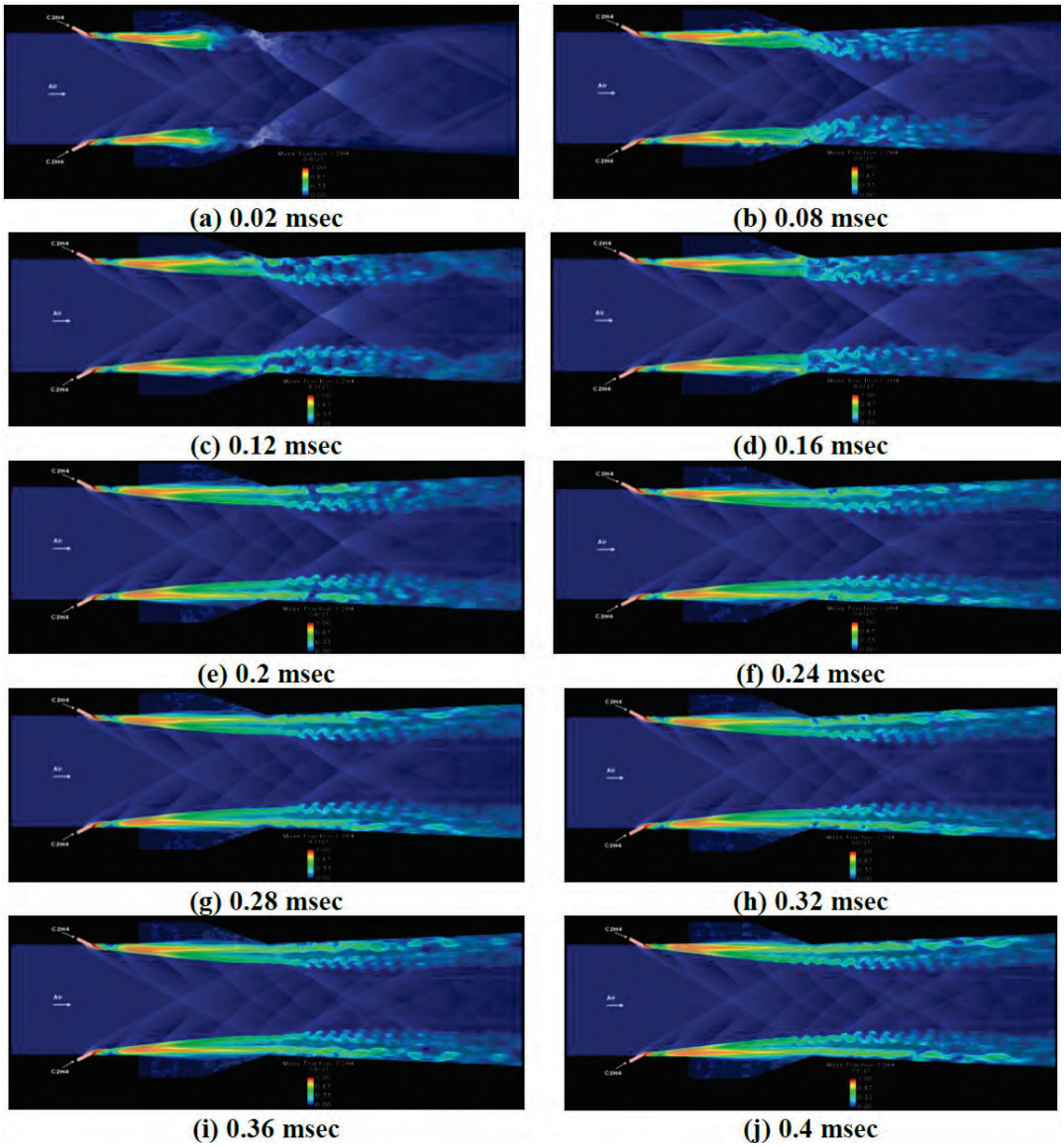
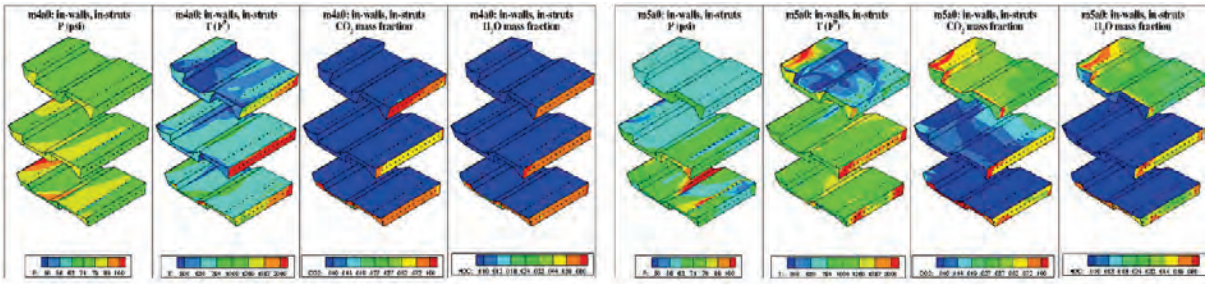


Figure 10. Fuel mass fraction contours at 10 equally-distributed instants

Figure 11. Percent differentiation of (a) HT Loss vs. thermal conductivity, and (b) ΔST vs. thermal conductivity



(a) m4a0 (b) m5a0

Figure 12. Strut-Wall Pressure (psi), Temperature (F), Y (CO_2) and Y (H_2O) contours

Another effort in support of the HF-2 extended the transient fueling analysis on an azimuthal symmetrical configuration. The effects of an asymptotic (quasi-steady) and fully-transient (pulsed) fueling injection with Φ~0.8 with a single-port located upstream of the cavity flame-holder with frozen assumption have been delineated and compared. The flow-path characteristics, such as mixing, concentration and circulation inside the cavity, and penetration were appreciably different. Although the baseline case shows a large entrainment of the fuel towards the bottom and ramp of the cavity, the pulsating case increases the fuel concentration over Φ=1 inside the cavity and increases penetration towards the core over 40%. The generated vorticity magnitudes in the pulsing injector present a multitude of distortions that persist farther downstream.

Several materials with different thermo-conductivities were analyzed. The variation in shock location was an important consideration in the development and material selection of scramjet engines. Although the material thermal conductivity had minimal effects on the stream-thrust, the isolator-shock location was mostly affected at lower-Mach numbers. Thus, the material selected for flame-holding hardware has minimal impact on the overall combustor performance and should be based on other factors such as cost, manufacturability, durability, and availability.

5. Importance to DoD

The recent success of the X-51 program has demonstrated the feasibility of scramjet flight. To make such flight routine; however, the complex physical phenomena in the flow-path must be understood and controlled. The joint USAF-DSTO HIFiRE program is designed to realize these objectives by acquiring data for discovery of fundamental physics issues. Recent advances in computational techniques, scalable software and large systems with massive numbers of processors have enabled CFD techniques to become a coequal partner with ground and flight tests. This project followed this paradigm, leveraging the ongoing ground and flight test program. Fueling strategies and the effect of increasing fueling in a cavity-based flame-holding environment have been examined during the past three years. Improper mixing combined with shock/combustion interactions triggered the potential for catastrophic unstart. Multiple validation studies were performed to ensure that the results represent reality. Finally, system performance and drag were estimated in a planned test over a broad operational range, resulting in a reduced risk while enhancing payoff. By simulating HIFiRE configurations that are already undergoing extensive testing, this Challenge Project has a direct influence on the evolution and analyses of current programs.

Acknowledgments

The authors are grateful for the computational resources of the DoD High Performance Computing Modernization Program (HPCMP, Subproject: AFVAW10102C4D). The computations were performed at the US Air Force Research Laboratory, DoD Supercomputing Resource Center (AFRL DSRC) and the US Army Engineer Research and Development Center (ERDC) DSRC. Flow Modeling and Simulation, Inc. made the computational tool, *HEAT3D*, available. In addition, the authors wish to thank Dr. Roger Kimmel of AFRL's Air Vehicles Directorate for providing the HF-1 Flight test data, geometry details, and assistance in evaluating and interpreting the experimental data. Several conversations with Dr. Michael Brown of AFRL's Propulsion Directorate are also gratefully acknowledged.

References

1. Aerosoft, Inc., VA 24060-6363, *GASP Version 5.0 Reference Guide*, February 2010.

2. Berger, K.T., Greene, F.A., Kimmel, R., Alba, C., and Johnson, H., "Aerothermodynamic Testing and Boundary-Layer Trip Sizing of the HIFiRE Flight 1 Vehicle." *Journal of Spacecraft and Rockets*, Vol. 46, No. 2, pp. 473–480, March–April 2009.

3. MacLean, M., Wadhams, T., Holden, M., and Johnson, H., "Ground Test Studies of the HIFiRE-1 Transition Experiment Part 2: Computational Analysis." *Journal of Spacecraft and Rockets*, Vol. 45, No. 6, pp. 1149–1164, November–December 2008.

4. Adamczak, D. and Kimmel, R L., "HIFiRE-1 Flight Trajectory Estimation and Initial Experimental Results." *17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, pp. 1–13, April 2011.

5. Smith, M., "HIFiRE-1 Best Estimated Trajectory Creation." Tech. Rep., NASA Dryden Flight Research Center, March 2011.

6. Kimmel, R. L. and Adamczak, D., "HIFiRE-1 Preliminary Aerothermodynamic Measurements." 41st AIAA Fluid Dynamics Conference and Exhibit 27 - 30 June 2011, Honolulu, Hawaii, USAF AFRL and DSTO AVD, June 2011, p. 35, AIAA 2011-3413.

7 Malo-Molina, F.J., Gaitonde, D.V., and Ebrahimi, H.B., "Numerical Analysis of a Direct Connect Supersonic Combustor." *AIAA paper 2010-403*, January 2011.

8. Malo-Molina, F.J., Gaitonde, D.V., and Ebrahimi, H.B., "Three Dimensional Analysis of a Fully Coupled Hypersonic Air-breathing Inlet-Combustor Flowpath." *AIAA paper 2010-412*, Jan 2010.

9. Aerosoft, Inc., 2000 Kraft Drive, Suite 1400 Blacksburg, VA 24060-6363, *GASP Version 5.0 Reference Guide*, February 2010.

10. Zha, G.-C. and Knight, D., "Three-Dimensional Shock/Boundary-Layer Interaction Using Reynolds Stress Equation Turbulence Model." *AIAA Journal*, Vol. 34, No. 7, pp. 1313–1320, July 1996.

11. Aerosoft, Inc., 2000 Kraft Drive, Suite 1400 Blacksburg, Virginia 24060-6363, GASP Version 5 Technical Reference, February 2010.

12. Malo-Molina, F.J., Gaitonde, D.V., Ebrahimi, H.B., and Ruffin, S.M., "High Fidelity Flowpath Analysis of a Supersonic Combustor Coupled to Innovative Inward-Turning Inlets." *AIAA Paper 2009-131*, January 2009.

13. Malo-Molina, F.J., Gaitonde, D.V., Ebrahimi, H.B., and Croker, B., "Flowfield Structure in Generic Inward-Turning-Based Propulsion Flowpath Components." *AIAA paper 2007-5744*, July 2007.

14. Vinagradov, V., Kobigsky, S., and Petrov, M., "Experimental Investigation of Kerosene Fuel Combustion in Supersonic Flow." *Journal of Propulsion and Power*, Vol. 11, No. 1, pp. 130–134, 1995.

15, Ortweth, P., Mathur, A., Vinogradov, V., Grin, V., Goldfeld, M., and Starov, A., "Experimental and Numerical Investigation of Hydrogen and Ethylene Combustion in a Mach 3-5 Channel with a Single Injector." *AIAA-96-3245*, July 1996.

16. Owens, M., Tehranian, S., Segal, C., and Vinogradov, V., "Flame-Holding Configurations for Kerosene Combustion in a Mach 1.8 Air Flow." *Journal of Propulsion and Power*, Vol. 14, No. 4, pp. 456–461, 1998.

17. Gruber, M., Donbar, J., Carter, C., and Hsu, K-Y., "Mixing and Combustion Studies Using Cavity-Based Flameholders in Supersonic Flow." *ISABE-2003-1204*.

18. Wilcox, D., "Turbulence modeling - an overview." *AIAA 2001-0724*, 2001.

19. Ebrahimi, H., "An Overview of Computational Fluid Dynamics for Application to Advanced Propulsion Systems." *AIAA Paper 2004-2370*, 2004.

20. Ebrahimi, H., Gaitonde, D., and Malo-Molina, F., "Parametric Study of 3-D Hydrocarbon Scramjet Engine with Cavity." *AIAA Paper 2007-0645*, January 2007.

21. Malo-Molina, F.J., Gaitonde, D.V., Ebrahimi, H.B., and Ruffin, S.M., "Three-Dimensional Analysis of a Supersonic Combustor Coupled to Innovative Inward-Turning Inlets." *AIAA Journal (0001-1452)*, Vol. 48, No. 3, pp. 572–582, 2010.

22. Ebrahimi, H.B., "Numerical Investigation of Jet Interaction in a Supersonic Freestream." *AIAA Journal of Spacecraft and Rockets*, Vol. 45, No. 1, pp. 95–103, 2008.

23. Ebrahimi, H.B., "Validation Database for Propulsion Computational Fluid Dynamics." *AIAA Journal of Spacecraft and Rockets*, Vol. 34, No. 5, pp. 642–650, September 1997.

24 Oberkampf, W.L. and Bart el, T.J., "Symmetric Body Vortex Wake Characteristics in Supersonic Flow." *AIAA Journal*, Vol. 18, No. 11, pp. 1289–1297, November 1980.

25 Kubota, H. and Stollery, J., "An Experimental Study of the Interaction Between a Glancing Shock Wave and a Turbulent Boundary Layer." *Journal of Fluid Mechanics*, Vol. 116, pp. 431–458, 1982.

26. Adamson, T. and Messiter, A., "Analysis of Two-Dimensional Interactions Between Shock Waves and Boundary Layers." *Ann. Rev. Fluid Mech.*, Vol. 12, pp. 103–138, 1980.

27. Brown, J., Brown, J., Kussoy, M., Holt, M., and Horstman, C., "Two-Component LDV Investigation of 3-Dimensional Shock/ Turbulent Boundary Layer Interactions." *AIAA Paper 87-0553*, 1987.

28. Holden, M.S., Wadhams, T.P., Harvey, J.K., and Candler, G.V., “Comparison between Measurements in Regions of Laminar Shock Wave Boundary Layer Interactions in Hypersonic Flows with Navier-Stokes and DSMC Solutions.” *AIAA Paper 2002-0435*, January 2002.

2.9 Rizzi, A. and Eriksson, L.-E., “Computation of inviscid incompressible flow with rotation.” *Journal of Fluid Mechanics*, Vol. 153, pp. 275–312, 1985.

Investigation of Total Pressure and Swirl Distortion in a Single-Stage Transonic Fan

Michael G. List
US Air Force Research Laboratory, Fans and Compressors Branch (AFRL/RZTF), WPAFB, OH
michael.list@wpafb.af.mil

Rebecca M. Howard
Universal Technology Corporation, Dayton, OH
rebecca.howard@wpafb.af.mil

Abstract

The impact of modern diffusers on embedded engine performance, operability, and lifing is an active area of research. Flow non-uniformities in total pressure and swirl-angle present challenges for designers, and a solid description of the physics are required. Analysis of a transonic-fan stage subjected to a representative total pressure and swirl distortion has been conducted. Preliminary results show that swirl distortion provides an incremental degradation of fan performance through several physical mechanisms. Understanding the detailed physics of distortion generation, transfer, and mitigation will enable more robust design cycles and more reliable engine products for future aircraft.

1. Introduction

Embedded turbine engines operate in flow-fields of non-uniform total pressure, total temperature, static pressure, and flow-angle (swirl). Sources of distorted flow include maneuvers, boundary-layer formation, separations, secondary flows, and hot-gas ingestion. Performance and operability degradation occur when such non-uniformities are transmitted through the fan or compressor. To adequately quantify the impact of distortion generated by a modern serpentine diffuser upon a high performance transonic an, a combined experimental and numerical program has been formulated to conduct detailed investigations, extract pertinent physics, and formulate design guidance to mitigate the negative impacts of distortion. The current document specifically addresses the computational fluid dynamics (CFD) analysis of an isolated transonic fan which augments experimental testing, and ultimately supports preparation for a combined diffuser-fan experiment. The fan conditions were selected to emulate traditional total-pressure-only distortion testing, but also included swirl distortion.

A four-phase program has been established by the Compressor Aero Research Lab (CARL) at Wright-Patterson Air Force Base to address the fundamental research questions associated with diffuser-fan interaction. Each phase is preceded by computational predictions and closed with computational post-dictions (using data to guide simulations and calibrate models). The first-phase detailed the diffuser flowfield through rigorous CFD studies[1] and experimentation. The results of the diffuser study serve as the inputs for the second-phase where the transonic fan stage will be analyzed in isolation using the total pressure distortion pattern from the diffuser discharge. The third-phase couples the diffuser and the fan to determine coupling effects resulting from combined total pressure and swirl distortion. The final phase adds flow control to the diffuser to actively improve performance.

The current work supports the second-phase and seeks to be able to predict and quantify distortion transfer. The early second-phase activities included simulation, and will follow with experimental quantification of the transonic fan subjected to total-pressure-only distortions generated by traditional distortion screens[2]. Follow-on computational activities, described here, included simulation of the fan with total pressure and swirl distortion in order to quantify the incremental impact of swirl distortion upon a fan already experiencing a total pressure distortion with multiple spatial frequencies.

The fan stage is the Rotor 4 configuration from the Parametric Blade Study[3], an effort during the 1980s which developed a number of high-performance transonic rotors. Rotor 4[3] was designed to be a state-of-the-art swept fan with diffusion control at the tip section. Its advanced design makes it still relevant today. More information on the design of the rotor can be found in Reference 3. The complete stage has been simulated with the spinner cone, providing a complete flow-field from the Aerodynamic Interface Plane (AIP) location to the exit instrumentation plane. The stage efficiency during test was 88.5% and pressure ratio was 1.92.

2. Methodology

The fan stage was simulated using TURBO, a 3D Unsteady Reynolds-averaged Navier-Stokes (3D URANS) solver[4,5,6]. It employs a finite-volume Roe scheme to obtain up to third-order spatial-accuracy and an implicit time- integration to obtain second-order temporal accuracy. The turbulence model is a CMOTT $k-\varepsilon$ model specialized for turbomachinery flows[7,8]. TURBO integrates to the wall, in the case of y^+ values below 10.5 and uses wall- functions for higher y^+ values. The turbulence model includes a near-wall damping term which should allow the use of the $k-\varepsilon$ model at higher grid resolutions. Each blade row grid is solved in the rotating reference frame, requiring no grid motion, and absolute frame flow variables are passed at each time-step with interpolations being performed to allow time-accurate multi-stage interactions. Parallelization was achieved through Message Passing Interface (MPI), and the code has been executed on almost 9,000 processors. It has been shown in many instances to be appropriate for multi-stage transonic turbomachinery analysis[9–12] and also analysis including inlet pressure distortion[13,14].

A mesh generation process employing the Turbomachinery Gridding System (TGS) and a procedure similar to Kamp, et al.[15] was used to develop elliptically-smoothed, structured-passage H-meshes for the rotor and stator. This also included an upstream portion which contained a spinner cone and extended forward to the AIP location. The downstream stator mesh extended to the exit of the axial portion of the experimental facility. O-grids were used as tip blocks for the rotor tip clearance.

The TURBO inputs were partitioned for computation on 1,882 processors, based upon a balance of memory usage and total turnaround time. The initial simulations required approximately 250,000 CPH per operating condition on the Hawk SGI Altix 4700 at the US Air Force Research Laboratory, DoD Supercomputing Resource Center (AFRL DSRC). The same partitioning was used on the Raptor Cray XE6 system. Subsequent operating points required much less computation time, as the previous point was a strong initial solution. Past experience was used to judge the adequacy of the time-step size and mesh size for capture of the pertinent physics.

2.1 Boundary Conditions

The inlet boundary condition was an isentropic inlet with total-pressure, total-temperature, and radial- and tangential-flow-angles specified. In order to accommodate a two-dimensional pattern, the boundary was modified to store and interpolate from an input file any of the four boundary condition quantities[2].

Total pressure was set to match the diffuser discharge, shown in Figure 1. Because static-pressure is computed from the interior of the flow-field, a non-uniform static pressure was able to develop at the inlet due to the rotor operating condition variations. This was a significant difference from the diffuser simulations which utilized a constant static pressure imposition at the exit. Additionally, List[2] showed an unsteady static pressure variation at the inlet plane due to the rotor passing.

For all simulations, the total temperature was set to the constant total temperature used in the experimental facility for computing corrected flow. This sets the corrected conditions, including the rotor corrected speed, to match the facility.

The inlet radial and tangential flow angles were set to zero previously[2], but for the present simulations were set to match the flow-angles from the isolated diffuser simulations[1]. In doing so, swirl and radial distortion that formed in the diffuser simulations were included. Previous simulations lacked this coupling of total pressure and swirl distortion, and comparison of total-pressure-only versus total pressure plus swirl distortion will allow differentiation of the impact of total pressure and swirl distortion upon the performance of the stage.

The exit boundary condition held an exit-corrected mass flow by adjusting local back-pressure to set velocity vector magnitudes. This methodology has been used in both APNASA and TURBO for a variety of cases[16,12]. Such a condition applied at the exit is less restrictive in terms of required input information. Rather than forcing the user to include a full, two-dimensional (2D) pressure field, a single parameter is set which allows the back pressure to vary around the annulus and from hub to casing—essentially allowing a prediction of the distortion transfer through the turbomachine.

2.2 Distortion Parameters

In order to quantify in a consistent manner the distortion presented to a fan by an upstream source, the SAE S-16 committee published a set of traditional distortion descriptors for total pressure deficit in AIR 1419[17]. These descriptors take advantage of data at the AIP, essentially the boundary at which responsibility shifts from an airframe manufacturer to an engine company. The S-16 committee recommended an array of 8 rakes of total pressure measurements be made at this plane with five (5) probes spaced at centers of equal areas along each rake. In doing so, this 5×8 probe configuration

provided five concentric rings of data, which could be used in computations and then averaged to form parameters for an entire engine inlet condition. Additional descriptors for swirl distortion were developed after a long-series of engine operation issues were tied to flow angularity at the AIP[18].

Figure 1 shows the diffuser exit pattern, and Figure 2 shows the face-averaged total pressure and swirl-angle versus circumferential location from the serpentine diffuser. Two probe configurations are shown in Figure 2, one matching the minimum recommend practice (5×8), and the other matching the maximum resolution of the facility measurements using 1° incremental rotation of the diffuser. Details of the equations may be found in References 17 and 18, and summarized in References 19 and 1. The CARL Python Utilities[20], a set of Python-based CFD tools, have been augmented with the capability to compute these descriptors, described briefly here:

- Circumferential distortion intensity, denoted by $\left(\frac{\Delta PC}{P}\right)$, couples the average total pressure around the annulus and the area-averaged low pressure regions to provide an understanding of the magnitude of the pressure deficit regions within a pattern.
- The extent of all the total pressure deficit region is computed per ring, and the extent, θ^{-}, associated with the maximum deficit provides information regarding the sector-size of distortion through which the rotor will pass.
- The multiple per-rev content (MPR) gives an indication of the number of large-deficit regions around the annulus.
- Radial distortion intensity, $\left(\frac{\Delta PR}{P}\right)$, provides a measure of the distortion experienced at each ring, relative to the average total pressure value over the whole face. The sum over all the rings should be zero.
- The swirl intensity, SI, represents an averaged swirl-magnitude.
- The swirl directivity, SD, represents the equivalent bulk swirl-motion of the flowfield. This indicates the tendency of the entire face to rotate in one direction.
- The swirl pairs (SP) indicates the number of swirl-pairs which exist in the flowfield.

3. Results

3.1 Distortion Pattern

The total pressure pattern at the AIP, shown in Figure 1, contained a number of flow features. At Top Dead Center (TDC) two tightly-wound vortices were generated by a separation region in the diffuser. At the sides, vortices were present due to top-to-bottom asymmetries at the diffuser throat. Lastly, a boundary-layer thickening occurred at Bottom Dead Center (BDC).

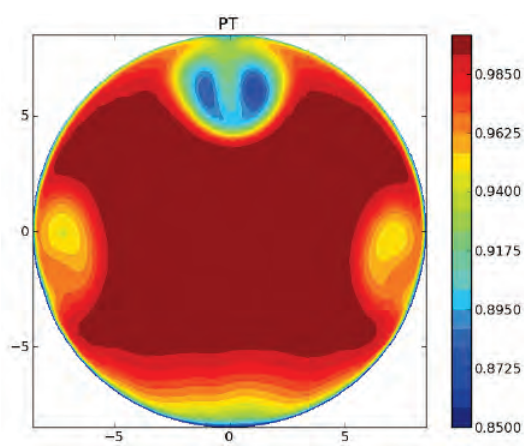


Figure 1. Total pressure pattern imposed at the inlet

Total pressure and swirl angle conditions were quantified in terms of the distortion descriptors. The CFD solution was sub-sampled in the same manner as discrete probes would sample in an experiment, in order to collect similar specific data-points. The face averages of total pressure and swirl (including all of the probe rings) for two specific probing configurations are shown in Figure 2. Recall that these represent the minimum recommended and the maximum facility resolutions. Clearly, increasing the circumferential resolution provided better information on the pressure deficits, and included additional features in the swirl distribution.

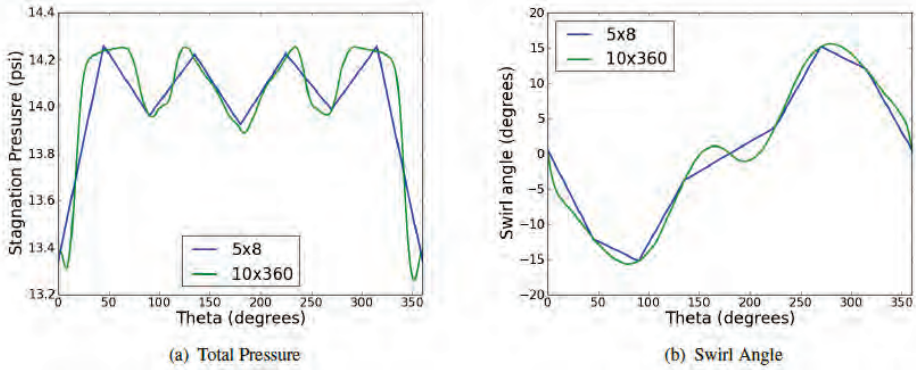


Figure 2. Face average values of total pressure and swirl computed from multiple-probe configurations

Table 1 shows that as more information was collected from the CFD by the probe rake, the distortion descriptors converged to values greatly different from the minimum recommendation. Typically, engine companies apply a coarse set of data to engine models to determine performance degradation. This effort has shown that for modern diffusers with multiple frequencies, higher spatial resolution is required in order to accurately set boundary conditions for distortion modeling and simulation. Conditions at the AIP are further discussed in Reference 1.

Table 1. Summary of distortion descriptors for several probe configurations

Configuration	$\left(\frac{\Delta PC}{P}\right)$	θ^{-}	MPR	$\left(\frac{\Delta PR}{P}\right)$	SI	SD	SP
5×8	0.020735	66.835100	1.100150	0.000000	7.765591	0.004413	0.995973
5×32	0.044084	48.207868	1.225215	0.000000	8.340429	-0.003692	1.008652
30×360	0.047666	45.845135	1.276907	-0.000000	8.390568	-0.004874	1.008564

Of primary interest from the table is that the number of swirl-pairs (SP) in the flow-field is very nearly one. Because of this fact, the swirl distortion can be represented by a flowfield similar to a sinusoid. Previous work in decomposing the distortion pattern[2] showed that the total pressure had a finite-number of important spatial modes. This overall reduced pattern would enable use of harmonic balance as a means to simulate a fan stage with a simplified version of the total pressure and swirl patterns. Such a procedure could be implemented as part of detailed design iteration, or for rapid analysis when a particular distortion pattern has been identified for scrutiny.

3.2 Fan Performance

The performance was compared to experimental data taken without distortion, as shown in Figure 3. The impact of total-pressure-only and of total-pressure-plus-swirl distortion upon the fan has not yet been fully-quantified, but a change in the pressure-rise was identified. As the fan stage was throttled (exit-corrected mass flow is reduced), the behavior of the two distorted inflow cases began to diverge. In order to understand the change in performance and relative differences, previously determined effects of total-pressure-only distortion transfer were used as starting points for the analysis. Many similar effects shown by Gorrell, et al.[14], Yao, et al.[21], and List[2] were seen and confirmed when inlet swirl was added to the boundary condition.

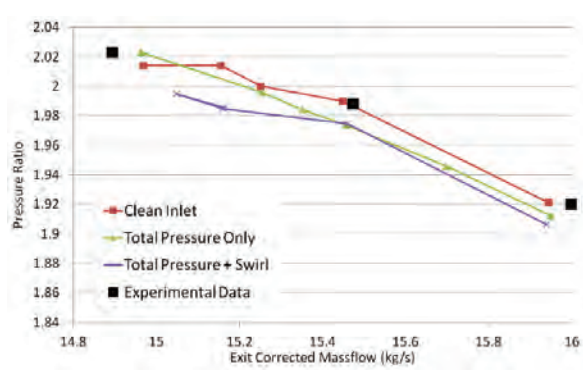


Figure 3. Comparison of full annulus CFD to experimental data

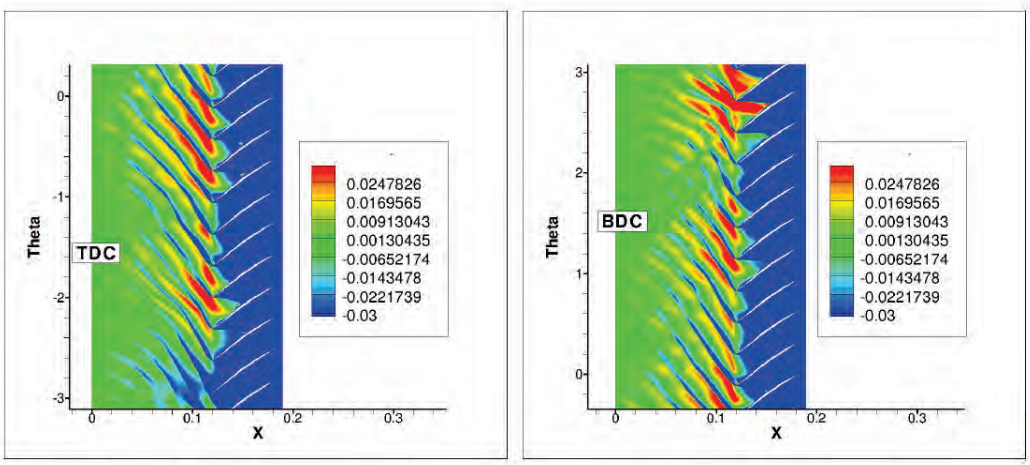
As discussed by a number of authors cast typically in terms of reduced-order models (such as[19,22]), co-swirl (swirl in the direction of rotor rotation) shifts pressure rise characteristics down and left. This is caused by a reduction in the incidence-angle of the rotor and a subsequent decrease in loading. Pressure-ratio necessarily drops and flow-rate may be decreased as well. Counter-swirl (swirl opposite the direction of rotor rotation) shifts pressure-rise characteristics up and right. While an increase in incidence improves the performance to an extent, additional losses are encountered with counter-swirl, and premature-stall is typically experienced.

Figures 4 thru 9 shows a cut near 90% span of the rotor unwrapped at the top and bottom of the annulus. In these figures the flow moves from left-to-right and the direction of rotor rotation is downward. Figures 4 and 5 show the tangential velocity for the total-pressure-only distortion, and the total-pressure-plus-swirl distortion at representative unsteady snapshots. Both simulations are in choke, but are not at the same exit-corrected mass-flow condition. Co-and counter-swirl regions are labeled on applicable figures. Likewise, Figures 6 and 7 show the entropy flux, and Figures 8 and 9 show the static pressure. Entropy flux, given by Equation 1, was used by Gorrell, et al.[23] and List, et al.[12] to demonstrate loss-propagation of various mechanisms, including wakes and vortices. In each figure, the TDC is the location of the tightly-wound vortices created by the diffuser separation and the BDC is the thickened- boundary-layer. At the top and bottom of each subfigure are the side-vortices.

$$f^s = \frac{\rho V \Delta s}{c_p} \tag{1}$$

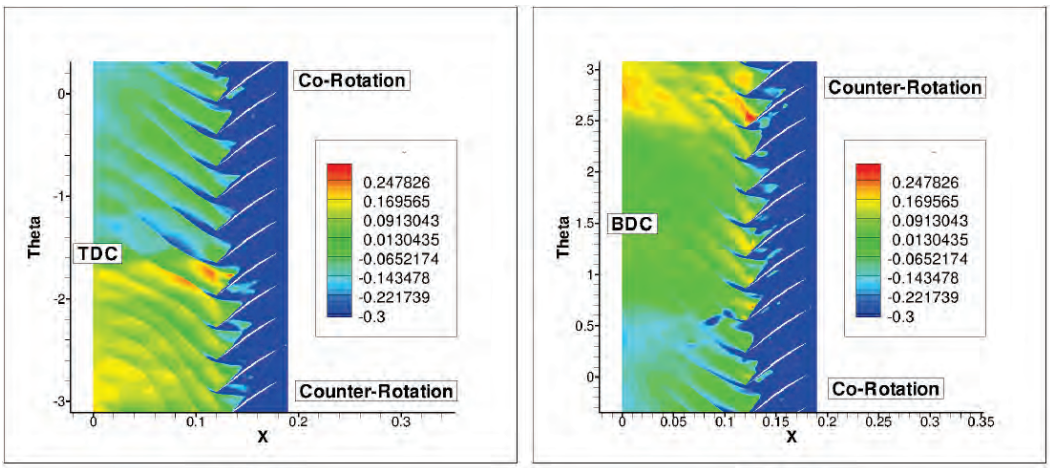
Figure 4 shows the swirl upstream of the fan. The contour levels were magnified relative to those of Figure 5 to more clearly show the physics. As the rotor passed TDC, BDC, or the side-vortices the swirl-angle just upstream of the rotor changed—upstream swirl was induced by the varying operating conditions of the rotor as it passed through the pressure deficit regions, even when no swirl component was specified as an inlet boundary condition. Figure 5, however, shows that when inlet swirl was included in the simulation, a drastic change in the swirl upstream of the rotors was seen. Additionally, it was clear that the shock-structure was greatly changed by the presence of co- or counter-swirl as one traveled around the annulus.

Shock-structure and losses can be more clearly seen in Figures 6 and 7 where regions of high-entropy flux indicate more loss transmission. For the total-pressure-only case the shock-structures were relatively consistent around the annulus. Perturbations in the bow shocks can be seen just after the rotor passed through deficit regions which also led to some differences in the exit-flow of the rotor. Contrarily, variations were easily observed with total-pressure-plus-swirl distortion. The regions of co-swirl resulted in lower-loss-production at the bow-shock, but as the rotor entered the counter-swirl region, especially as it passed through the deficit region of the TDC vortices and the side vortex, the shock produced much more loss and exhibited more positional variation indicating a larger range of operating conditions experienced around the annulus.



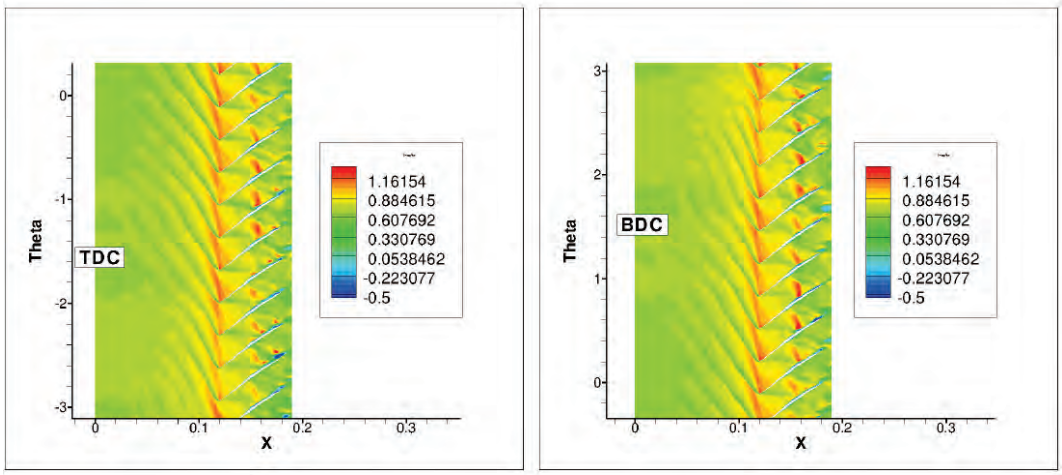
(a) Top view (b) Bottom view

Figure 4. Tangential velocity near 90% span unwrapped for the fan with total-pressure-only distortion



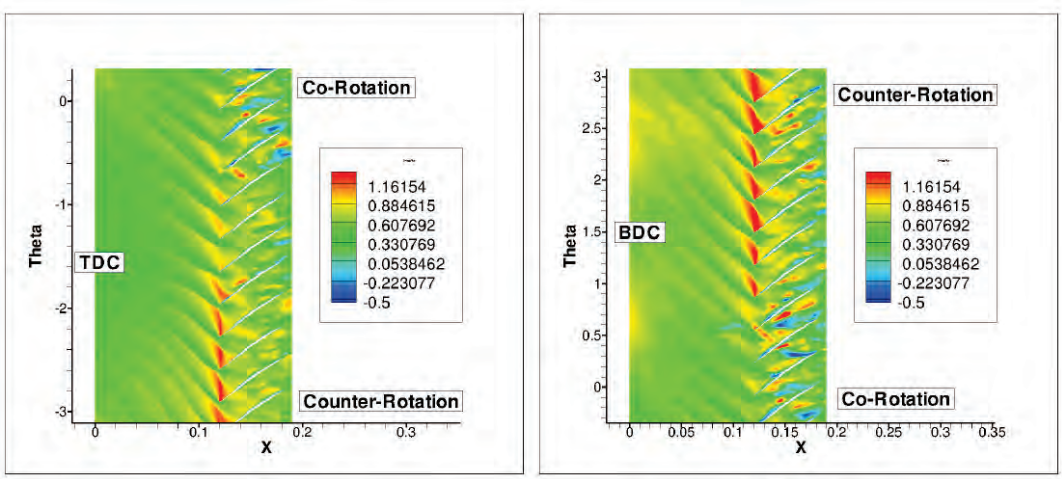
(a) Top view (b) Bottom view

Figure 5. Tangential velocity near 90% span unwrapped for the fan with total pressure and swirl distortion



(a) Top view (b) Bottom view

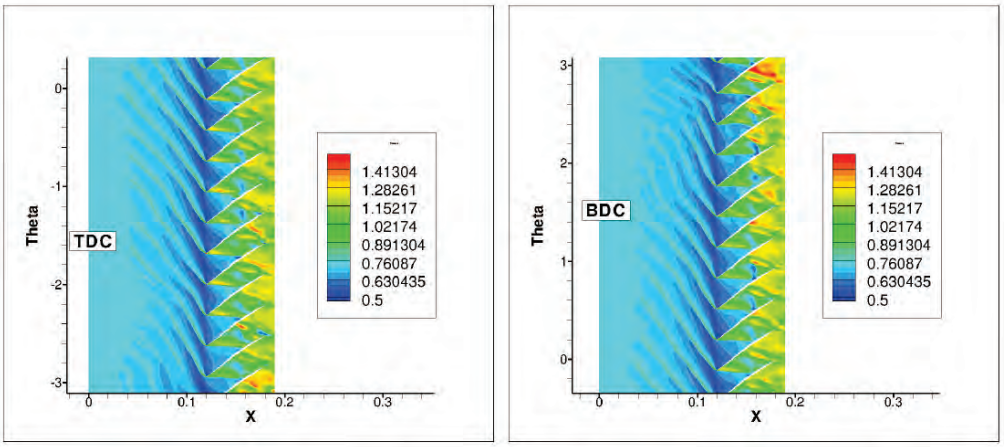
Figure 6. Entropy flux near 90% span unwrapped for the fan with total-pressure-only distortion



(a) Top view (b) Bottom view

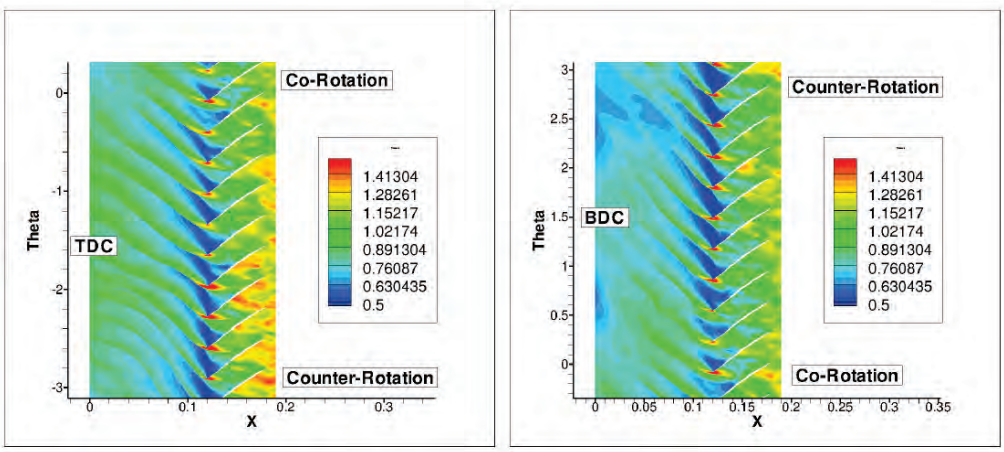
Figure 7. Entropy flux near 90% span unwrapped for the fan with total pressure and swirl distortion

The static pressure contours in Figures 8 and 9 also demonstrate the variation of operating conditions around the annulus. Without inlet-swirl, most of the variation occurred as the rotor passed through the side-vortex on its way toward BDC. This is a possible indication that as a rotor passed through that sector it had not fully-recovered after encountering the deficit region at TDC, and the subsequent deficit region of the side-vortex drove the rotor far outside of design operation. With inlet-swirl, the issues were compounded as a static pressure variation also occurred at the inlet. The addition of the co-swirl and counter-swirl regions resulted in a time-averaged decrease in pressure-rise (see Figure 3). This variation in static pressure indicated that when coupling the diffuser and fan an impact on the upstream diffuser will be observed.



(a) Top view (b) Bottom view

Figure 8. Static pressure near 90% span unwrapped for the fan with total-pressure-only distortion



(a) Top view (b) Bottom view

Figure 9. Static pressure near 90% span unwrapped for the fan with total pressure and swirl distortion

Due to the size of a full-annulus, unsteady dataset, post-processing and analysis require extensive planning and resources. As such, the results presented are considered preliminary and will be more vigorously pursued as new methodologies and tools for analysis are developed.

4. Conclusion

Inclusion of swirl at the inlet plane resulted in additional performance degradation beyond that of the total-pressure-only pattern. The mechanisms associated with additional distortion generation as the flow moved through the fan were still seen.

Complex distortion patterns including multiple components and large numbers of spatial frequencies result in significant issues for aircraft operability through reduced aerodynamic performance, increased aeromechanic forcing, and premature engine stall or surge. Modeling of these phenomena first requires development of a detailed understanding of the flow physics. Subsequent designs may then account for distortion transfer which will help mitigate these issues in future aircraft systems.

As a better understanding of the distortion patterns affecting modern propulsion systems becomes available, the possibility of using reduced-order methods to very accurately model specific configurations will blossom. Specifically, the harmonic balance technique becomes a viable candidate for modeling of flow-fields based upon the values of the SAE distortion descriptors. This ultimately reduces large datasets, relieves data-mining constraints, and hastens the pace of analysis to enable an impact in the design process.

Acknowledgments

The authors would like to thank the staff of the AFRL/DSRC for tirelessly working to support, train, and provide resources to the HPCMP users. The authors would also like to thank the HPCMP for its support through the award of a Challenge Project and generous allocations.

References

1. Sanders, D.D. and List, M.G., "Simulation of Distortion Generation in a Modern Serpentine Diffuser to Improve Experimental Effectiveness." *High Performance Computing Modernization Program Contributions to DoD Mission Success*, 2012.

2. List, M.G., "Distortion Pattern Analysis for Diffuser-Fan Interaction." *DoD HPCMP Users' Group Conference*, 2011.

3. Law, C.H. and Puterbaugh, S.L., "Parametric Blade Study Test Report Rotor Configuration Number 4." *Air Force Wright Aeronautical Labs AFWAL-TR-88-2110*, Nov. 1988

4. Chen, J., Celestina, M.L., and Adamczyk, J.J., "A new procedure for simulating unsteady flows through turbo-machinery blade passages." *ASME Paper no. 94-GT-151*, ASME Turbo Expo 1994, 1994.

5. Chen, J.P. and Barter, J.W., "Comparison of Time-Accurate Calculations for the Unsteady Interaction in Turbo-machinery Stage." *ASME Paper 98-GT-3292*, 1998.

6. Chen, J.P. and Briley, W.R., "A Parallel Flow Solver for Unsteady Multiple Blade Row Turbomachinery Simulations." *ASME Paper GT2001-0348*, 2001.

7. Shih, T.H. and Hsu, A.T., "An improved $k - \varepsilon$ model for near wall turbulence." Tech. rep., Oct. 1991.

8. Zhu, J. and Shih, T.H., "CMOTT Turbulence Module for NPARC." *Contract Report NASA CR 204143*, NASA Glenn Research Center, Lewis Field, OH, 2000.

9. Gorrell, S.E., Okiishi, T.H., and Copenhaver, W.W., "Stator-Rotor Interactions in a Transonic Compressor, Part 1: Effect of Blade-Row Spacing on Performance." *ASME Journal of Turbomachinery*, 125, pp. 328–335, 2003.

10. Gorrell, S., van de Wall, A., and Tsung, F., "Application of Time-Accurate CFD in Order to Account for Blade-Row Interactions and Distortion Transfer in the Design of High Performance Military Fans and Compressors." *Proceedings of the HPCMP Users Group Conference 2005*, June 27–30, 2005.

11. List, M.G., Gorrell, S.E., Turner, M.G., and Nimersheim, J.A., "High fidelity modeling of blade row interaction in a transonic compressor." *AIAA Paper no. 2007-5045*, 43rd AIAA/SAE/ASME Joint Propulsion Conference, 2007.

12. List, M.G., Gorrell, S.E., and Turner, M.G., "Investigation of Loss Generation in an Embedded Transonic Fan Stage at Several Gaps Using High Fidelity, Time-Accurate CFD." *Journal of Turbomachinery*, 132(011014), January 2010.

13. Yao, J., Gorrell, S.E., and Wadia, A.R., "A Time-Accurate CFD Analysis of Inlet Distortion Induced Swirl in Multistage Fans." *Paper No. AIAA-2007-5059*, 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference, 2007.

14. Gorrell, S.E., Yao, J., and Wadia, A.R., "High Fidelity URANS Analysis of Swirl Generation and Fan Response to Inlet Distortion." *Paper No. AIAA-2008-4985*, 44rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference, 2008.

15. Kamp, M.A., Nimersheim, J., Beach, T., and Turner, M.G., "A turbomachinery gridding system." *AIAA Paper 2007-18*, 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007.

16. List, M.G., *Quarter annulus simulations of blade row interaction at several gaps and discussion of flow physics*, Master's thesis, University of Cincinnati, Cincinnati, OH, August, Aerospace Engineering, 2007.

17. Society of Automotive Engineers, "Inlet Total-Pressure-Distortion Considerations for Gas-Turbine Engines." *Aerospace Information Report AIR1419, Rev. B*, 2007.

18. Society of Automotive Engineers, "A Methodology for Assessing Inlet Swirl Distortion." *Aerospace Information Report AIR5686*, 2010.

19. Cousins, W., "History, Philosophy, Physics, and Future Directions of Aircraft Propulsion System/Inlet Integration." *ASME Paper GT-2004-54210*, ASME, 2004.

20. List, M.G. and Car, D., "Tool and process improvement for high-fidelity compressor simulations." *DoD HPCMP Users Group Conference*, IEEE Computer Society, pp. 119–126, 2009.

21. Yao, J., Gorrell, S.E., and Wadia, A. R., “High-fidelity numerical analysis of per-rev-type inlet distortion transfer in multistage fans — part i: Simulations with selected blade rows.” *Journal of Turbomachinery*, 132(041014), October, pp. 1–10, 2010.

22. Davis, M. and Hale, A., “A Parametric Study on the Effects of Inlet Swirl on Compression System Performance and Operability using Numerical Simulations.” *ASME Paper GT2007-27033*, ASME Turbo Expo 2007.

23. Gorrell, S. E., Okiishi, T.H., and Copenhaver, W.W., “Stator-Rotor Interactions in a Transonic Compressor, Part 2: Description of a Loss Producing Mechanism.” *ASME Journal of Turbomachinery*, 125, pp. 336–345, 2003.

Simulation of Distortion Generation in a Modern Serpentine Diffuser to Improve Experimental Effectiveness

Darius D. Sanders and Michael G. List
US Air Force Research Laboratory, Fans and Compressors Branch (AFRL/RZTF), Wright-Patterson AFB, OH
{darius.sanders, michael.list}@wpafb.af.mil

Abstract

Modern serpentine diffusers create distorted flowfields which are ingested by embedded propulsion systems. Fundamental research into the generation and content of such distortion patterns is necessary to better understand inlet conditions for the engine. Simulation of the diffuser provided a number of insights for a complementary experimental program including Particle Image Velocity (PIV) plane placement and seeding locations, necessary resolution of measurement, and expected operational range and conditions. Analysis of the results provided better definition of some experimental requirements and sensitivities for capturing multiple per-rev distortions. In comparing with preliminary experimental results, it was determined that diffuser flowfield simulations may require detached-eddy simulation (DES) or large-eddy Simulation (LES) to properly capture flow feature position and the impact of the separation regions.

Nomenclature

α	Swirl angle
θ	Circumferential extent
C_θ	Tangential velocity
C_x	Axial velocity
i	Ring index
k	Sub-region within a ring
N	Number of probe rings
Q	Number of sub-regions within a ring

1. Introduction

Embedded engine applications subject turbomachinery to nonuniform total pressure, total temperature, static pressure, and flow angle (swirl). Additional sources of distorted flow include maneuvers, boundary-layer formation, separations, and hot-gas ingestion. Tangible changes in performance and operability occur when non-uniformities impinge upon the fan or compressor face. Modern aircraft systems deliver high-spatial-frequency content distortion patterns at the fan-face which are not well understood. To adequately quantify the impact of distortion generated by a modern serpentine diffuser upon a high-performance transonic fan, a combined experimental and numerical program has been formulated to conduct detailed investigations, extract pertinent physics, and formulate design guidance to mitigate the negative impacts of distortion. The current document specifically addresses computational fluid dynamics (CFD) and analysis of the isolated diffuser to augment experimental testing of such, and ultimately to support preparation for a combined diffuser-fan experiment. Numerical investigations of the diffuser preceded the experimental investigation for predictive purposes. Additional CFD analyses will be conducted to correct discrepancies found as the experimental data is collected.

During the first-phase of the program, CFD methods were used to study the flowfield of an inlet diffuser configuration and support experiments conducted in an annular cascade facility (ACF). The ACF is shown in Figure 1(a). The steady-pressure distortion field was measured with total-pressure rakes positioned at the diffuser discharge, or Aerodynamic Interface Plane (AIP) (see Figure 1(b)), in the isolated test configuration. With this data, screens will be developed to simulate this distortion field in front of the transonic fan. Also during this phase, steady and time-resolved velocity and pressure field measurements at the AIP have been obtained through Particle Image Velocity (PIV) measurements and conventional probe arrays[1]. Specific testing details relevant to this discussion are:

- A bell mouth extends from the diffuser throat into the flow barrel to reduce separation entering the diffuser.
- Surface static pressures were measured along the diffuser centerline and around the throat location.
- An array of 8 rakes, each containing 10 total-pressure measurements was located at the diffuser discharge plane.
- The diffuser assembly could rotate in 1° increments, making it possible to create highly-resolved time-averaged datasets.

(a) Annular Cascade Facility (b) Rake Case

Figure 1. Annular Cascade Facility with diffuser and flow barrel and the rake case

The next phases incorporate learning from the first-phase to analyze distortion transfer and impact on a transonic fan, and ultimately to determine coupling effects between the diffuser and fan. Through the use of CFD analysis, a positive impact has been seen on the planning, evolution, and data processing of the corresponding experimental effort. This study expanded the ability to complete rapid CFD analysis of advanced inlet geometries, and provided guidance for future measurement techniques.

2. Methodology

2.1 Computational Fluid Dynamics

Steady and unsteady simulations of the isolated serpentine diffuser geometry, as well as the diffuser with the upstream flow conditioning barrel, were conducted using the commercial CFD solver StarCCM+[2]. A number of additional codes were previously used for Reynolds-averaged Navier-Stokes (RANS) simulation of the diffuser, but the StarCCM+ solution was determined to be typical of a range of the solvers used. The flow barrel was not simulated with the other solvers.

An unstructured polyhedral mesh topology was generated using StarCCM+'s internal mesh-generator. A total cell count of 3.3M cells was used for the isolated diffuser, and a total of 6.8M cells were used for the diffuser with a flow barrel. Prism cells were used to resolve viscous effects from the wall with a $y+$ range of 5 to 30, while pure polyhedral cells were used in the flow-path.

Solutions were obtained using the RANS solver with a 2nd-order upwind spatial discretization. For unsteady simulations, 2nd-order temporal discretization was employed with a time-step of 10 μs. Flowfield convergence was considered achieved after residuals decreased below 1×10^{-4}. The k-ω SST turbulence model was used.

In the isolated diffuser simulations, the inlet boundary condition was an imposition of total-pressure, total-temperature, and flow-angle upstream of the bell mouth. For simulations with the flow barrel a mass-flow inlet was used in order to obtain a desired throat Mach number. In both simulations, a constant static pressure was applied at a plane downstream of the AIP.

Without the flow barrel, a steady-state solution was obtained and the standing distortion pattern was found. Incorporation of the flow barrel required an unsteady solution due to recirculation regions behind the diffuser's bell mouth but within the barrel. To mitigate the impact of time-variation on the analyses, an average was taken over a series of aperiodic events in a manner similar to a running average used in the facility—in fact, the same temporal duration was averaged numerically and experimentally, 0.06s, a lengthy time-scale relative to the fan.

Qualitative flow analysis was completed on the time-averaged and unsteady flowfields, using Fieldview and various flow visualization techniques to investigate the vortical structures produced by the diffuser. The AIP was extracted using Tecplot and its interpolation utilities. The extracted total-pressure and swirl-angle are the basis for the following analyses.

The full process to simulate one facility flow rate steady-state used 48 cores. With the flow barrel included, the unsteady simulations required 80 cores each. Total processor-hour usage for the investigation was 50,000 CPH on the Raptor (Cray XE6) system at the US Air Force Research Laboratory DoD Supercomputing Resource Center (AFRL DSRC). The total turnaround time for each simulation, including post-processing, was 3 weeks.

2.2 Distortion Descriptors

Experimental data is only available at probe locations. In order to simulate the experimental probe locations at the AIP, a mesh was created for comparisons to the experimental measurements. The circumferential location, number of probes per rake, and number of rakes could be set and adjusted. Bilinear interpolation was used to set the value at each probe location. The configuration is denoted as *nprobes*×*nrakes*, where *nprobes* represents the number of probes on a given rake, and *nrakes* is the number of equi-spaced rakes around the annulus. The configuration recommended by the SAE S-16 committee was 5 probes with 8 rakes, 5×8, and is shown in Figure 1(b). The probes are located at the centroids of equal areas of the annulus. The probes located at the same radius are grouped as a ring.

Traditional distortion descriptors for total-pressure-deficit were reported in AIR 1419[3]. The average pressure around the annulus at a given radius is seen in Equation 1, and the area-averaged low-pressure regions are computed using Equation 2. The extent of a region of low-total-pressure on a given ring, *i*, is computed by subtracting the beginning and ending locations and is denoted by θ_{ik}^{-}. The distortion intensity is given in Equation 3 and the multiple per-rev content is given in Equation 4.

$$\left(PAV\right)_i = \frac{1}{360}\int_0^{360} P\left(\theta\right)_i \, d\theta \tag{1}$$

$$\left(PAV\ LOW\right)_i = \frac{1}{\theta_i^{-}}\sum_{k=1}^{Q}\int_{\theta_{ik}^{-}} P\left(\theta\right)_i \, d\theta \tag{2}$$

$$\left(\frac{\Delta PC}{P}\right) = \frac{\left(PAV\right)_i - \left(PAV\ LOW\right)_i}{\left(PAV\right)_i} = \frac{\sum_{k=1}^{Q}\left(\frac{\Delta PC}{P}\right)_{ik}\theta_{ik}^{-}}{\sum_{k=1}^{Q}\theta_{ik}^{-}} \tag{3}$$

$$\left(MPR\right)_i = \frac{\sum_{k=1}^{Q}\left[\left(\frac{\Delta PC}{P}\right)_{ik}\theta_{ik}^{-}\right]}{\max_{k=1,Q}\left[\left(\frac{\Delta PC}{P}\right)_{ik}\theta_{ik}^{-}\right]} \tag{4}$$

The radial intensity of the total-pressure distortion requires computation of the face-averaged total-pressure, Equation 5. The radial intensity is Equation 6.

$$\left(PFAV\right) = \frac{1}{N}\sum_{i=1}^{N}\left(PAV\right)_i \tag{5}$$

$$\left(\frac{\Delta PC}{P}\right)_i = \frac{\left(PFAV\right) - \left(PAV\right)_i}{\left(PFAV\right)} \tag{6}$$

The swirl descriptors were computed using the same Python tools as the traditional distortion descriptors. Equations were taken from Reference 4. Swirl is defined as $\alpha = \tan^{-1}\left(C_\theta / C_x\right)$. The sector swirl (Equation 7) is the average value of the swirl-angle in a region where positive- or negative-swirl exists. The swirl intensity is computed by Equation 8 and represents the extent-weighted swirl-magnitude. The swirl-directivity is shown in Equation 9. The directivity represents the equivalent bulk swirl-motion of the flowfield. The swirl-pairs indicate the number of swirl-pairs which exist in the flowfield and can be computed via Equation 10.

$$SS_{i,k}^{\pm} = \frac{1}{\theta_{i,k}^{\pm}} \int_{\theta_{i,k}^{\pm}} \alpha(\theta)_{i,k} \, \mathrm{d}\theta \tag{7}$$

$$SI_i = \frac{\sum_{k=1}^{m} SS_{i,k}^{+} \times \theta_{i,k}^{-} + \sum_{k=1}^{m} \left| SS_{i,k}^{-} \right| \times \theta_{i,k}^{-}}{360} \tag{8}$$

$$SD_i = \frac{\sum_{k=1}^{m} SS_{i,k}^{+} \times \theta_{i,k}^{-} + \sum_{k=1}^{m} SS_{i,k}^{-} \times \theta_{i,k}^{-}}{\sum_{k=1}^{m} SS_{i,k}^{+} \times \theta_{i,k}^{-} + \sum_{k=1}^{m} \left| SS_{i,k}^{-} \right| \times \theta_{i,k}^{-}} \tag{9}$$

$$SP_i = \frac{\sum_{k=1}^{m} SS_{i,k}^{+} \times \theta_{i,k}^{-} + \sum_{k=1}^{m} \left| SS_{i,k}^{-} \right| \times \theta_{i,k}^{-}}{2 \times \max_{k=1,m} \left(SS_{i,k}^{+} \times \theta_{i,k}^{-}, + \sum_{k=1}^{m} \left| SS_{i,k}^{-} \times \theta_{i,k}^{-} \right| \right)} \tag{10}$$

In order to compute the distortion descriptors, a piecewise linear function class was developed in Python. This class enabled rapid integration and extraction of the regions above and below the function mean value. The CARL Python Utilities[5] have been augmented with this linear spline capability.

3. Results

Figure 2 shows the diffuser and flow barrel system with streamlines indicating the important flow features. Stream-traces are colored by Mach number, while the diffuser discharge is colored by total-pressure recovery. Moving from left-to-right, a number of physical phenomena can be seen.

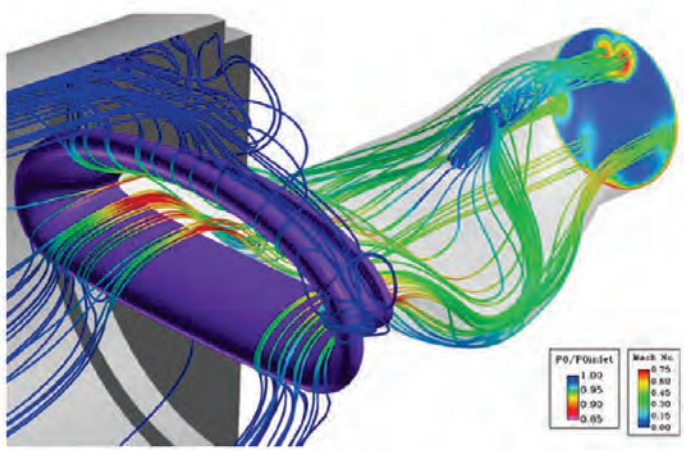


Figure 2. Diffuser flowfield, including upstream flow barrel, shown with stream-traces

Within the flow barrel a large recirculation zone was seen behind the bellmouth of the diffuser. This unsteady region made it necessary to utilize unsteady simulations and time-average to obtain results.

Next, as the flow enters the bellmouth and reaches the throat, a rapid acceleration was seen. The bellmouth performed well in eliminating separations and straightening the flow. The top-to-bottom asymmetry of the diffuser throat generates a vortex at each of the starboard and port sides. These vortices were seen at the diffuser discharge located at the sides.

Aft of the recurve location in the diffuser, the stream-traces converge near top dead center (TDC) and form a pair of tightly-wound vortices. This roll-up was a direct result of the separation region (denoted by the large bundle of blue streamtraces upstream of the vortices), which created adequate pressure-loss to draw the surrounding flow towards TDC. This pressure-loss also generated most of the swirl seen at the diffuser exit.

Preliminary experimental data has been obtained, and the results of the discharge instrumentation rakes have been plotted with the CFD results in Figure 3. The total-pressure distortion pattern is composed of four (4) main flow features. At TDC the tightly-wound vortex created by the upper-wall separation. Near the sides, designated as 90° and 270°, the vortices formed by the asymmetric bellmouth were seen. At bottom dead center (BDC), or 180°, a boundary-layer thickening occurred.

In comparing the CFD results with the preliminary experimental data shown in Figure 3, several discrepancies were noticed. First, the experimental results are not located close enough to the walls to pick up the large total-pressure deficit. Next, more loss was seen in each of the CFD flow features than the experimentally-captured features. Third, the side-vortices were not as tight, and are closer to 0° experimentally, indicating unsteadiness and a smearing of the vortices. Finally, the TDC vortices had lifted off the surface in the experiment while the CFD remained close to the surface. Downstream of the discharge in the CFD these vortices did separate from the surface, but the distance was much greater than what has been observed experimentally.

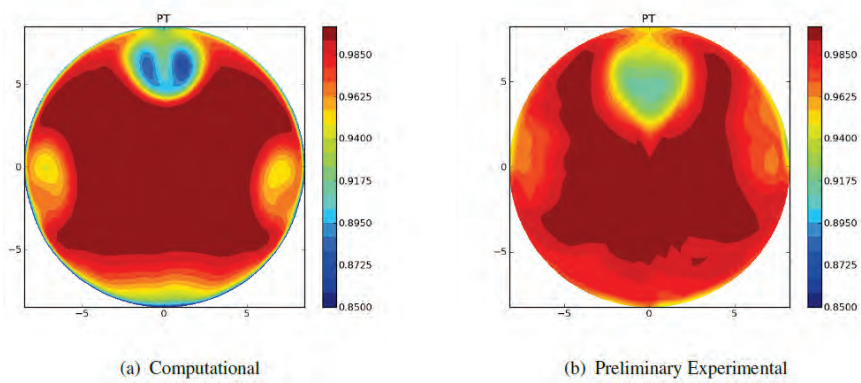


Figure 3. Total-pressure near the diffuser discharge (the AIP)

Figure 4 shows the distribution of static pressure through the diffuser at 0° and 180°. Good agreement has been obtained along most of the lower surface. Along the upper surface a sharp drop in pressure was seen as the flow accelerated around the recurve section and separated. Once the CFD diverged from the experimental data the correct values aft of the separated region were not reattained.

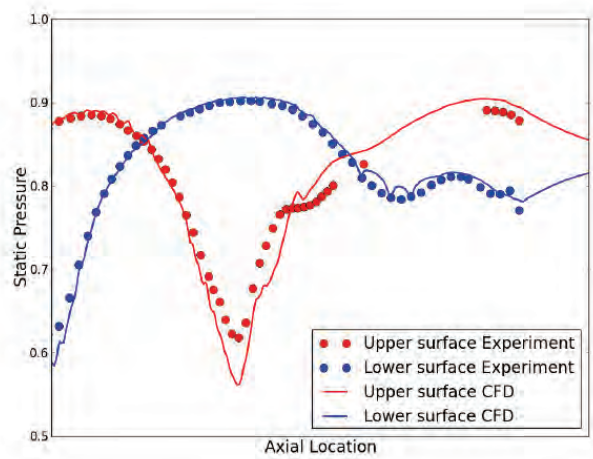


Figure 4. Comparison of CFD and experimental static pressure along the upper and lower walls

3.1 Analysis

As previously discussed, distortion descriptors made possible the consistent analysis of experimental data collected at a series of measurement points. In order to provide *a priori* values for the distortion descriptors which could be measured in the experiment, a number of rake configurations were applied to the CFD results. Two of the rake configurations are discussed here—a 5×8 and a 10×360. The first is the *de facto* standard specified by AIR 1419[3] and represents nominal

measurements taken for a system without the capability to rotate. The second represents the maximum resolution of the annular cascade facility, where there were 10 probes per rake, and the hardware could be rotated at 1° increments to reconstruct the full pattern.

Figure 5 shows the total-pressure field from the CFD solution on the left, with the discrete probe locations shown in black. The discrete pattern which is formed by the probe values is shown on the right. The pattern is clearly degraded from the smooth CFD contours, demonstrating the loss of fidelity when using the standard 8 rake system. Figure 6 likewise shows the swirl-angle. From these discrete probe locations, each of the 5 concentric circles formed by the probes were extracted and used to compute the distortion descriptors. The values per ring have been collected in Table 1.

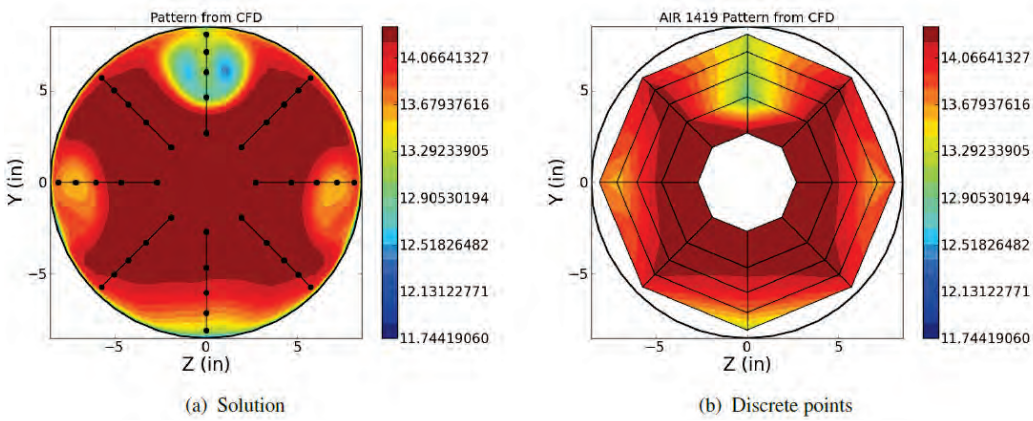


Figure 5. Total-pressure solution and discrete probe values for the 5×8 probe configuration

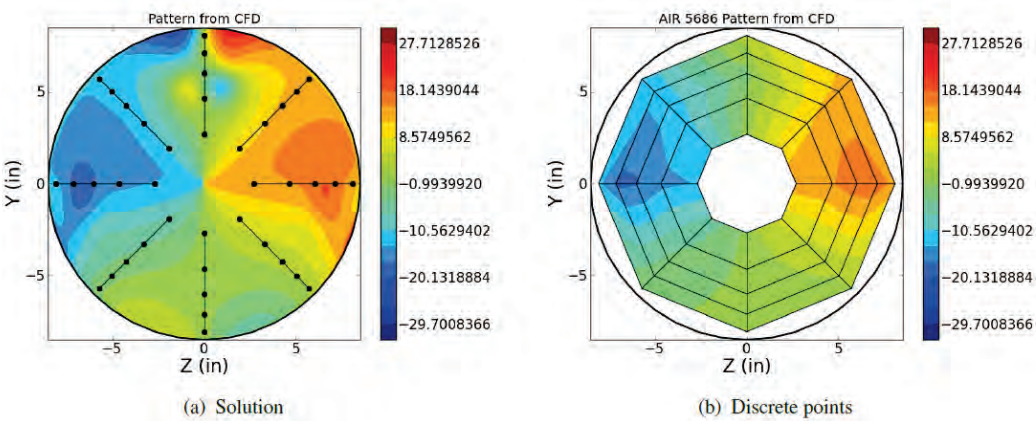


Figure 6. Swirl CFD solution and discrete probe values for the 5×8 probe configuration

Table 1. Distortion descriptors by ring for the 5×8 probe array

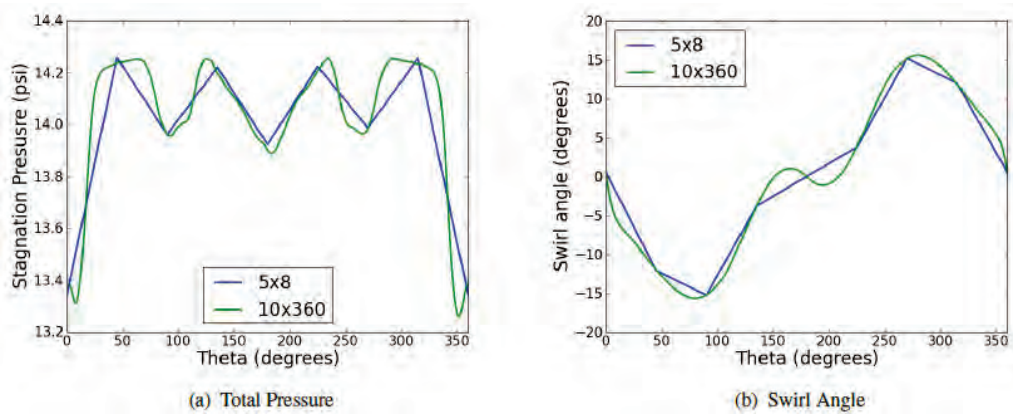
	$\left(\frac{\Delta PC}{P}\right)$	θ^{-}	MPR	$\left(\frac{\Delta PR}{P}\right)$	SI	SD	SP
Ring 1 (hub)	0.000093	78.939763	1.890550	-0.019563	7.681347	0.000102	0.999966
Ring 2	0.043855	78.925576	1.000000	-0.007031	7.662768	-0.007931	0.992181
Ring 3	0.028268	69.361619	1.088735	0.000290	8.247871	-0.040068	0.965881
Ring 4	0.024401	56.903416	1.798594	0.009437	8.169948	0.009213	0.992874
Ring 5 (tip)	0.016503	65.838543	1.851098	0.016867	7.198353	0.067590	0.947581

Table 2 gives the final values for both probe configurations. These values are the average of the values for the concentric circles (recall that the probes are stationed on centers of equal area). There were considerable discrepancies between the two rake configurations, demonstrating the necessity of accurately capturing the flow features through rotation of the distortion pattern when multiple per-rev frequencies exist. In incorporating more integration points, the tangential intensity doubled and the tangential extent was lessened. The pressure distortion is significant but not as high as many observed cases. The swirl distortion parameters indicate that the swirl pattern is close to a 1 per-rev distortion which makes it easier to analyze and can be simulated with the harmonic balance technique. (Note it is clear from Equations 5 and 6 that the sum of the radial intensity over the probe rings should be zero.)

Table 2. Summary of distortion descriptors for both probe configurations

Configuration	$\left(\frac{\Delta PC}{P}\right)$	$\bar{\theta}$	MPR	$\left(\frac{\Delta PR}{P}\right)$	SI	SD	SP
5×8	0.020735	66.835100	1.100150	0.000000	7.765591	0.004413	0.995973
10×360	0.047639	46.114129	1.266264	-0.000000	8.409388	-0.004796	1.009462

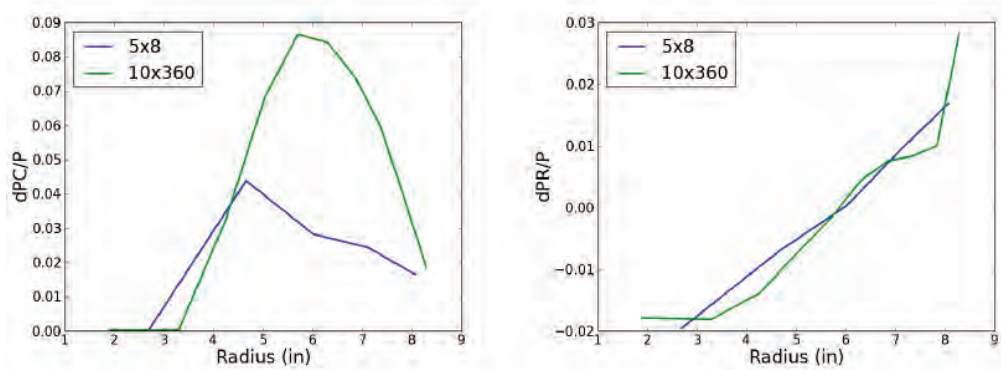
The discrepancies in the distortion parameters can be better understood with the average total-pressure and swirl-angles from the probes, as shown in Figure 7. These averages clearly show the improved spatial resolution and it is clear that integral methods would under-predict values when the full pattern was not taken into account.



(a) Total Pressure (b) Swirl Angle

Figure 7. Face average values computed from both probe configurations

Figure 8 shows the circumferential and radial intensities plotted versus radius. The tangential intensity is greatly-increased at higher-radius, when more tangential measurement points are included. This was expected, because although the tangential extents of the deficit regions decreased, the deficit region integrated increased sufficiently. The radial intensity is rather well captured by the coarser probe array as the face average value of total pressure in the full CFD solution was relatively close to the value obtained by averaging the probe values.



(a) $\Delta PC/P$ (b) $\Delta PR/P$

Figure 8. Circumferential and radial intensities versus probe ring radial location

Figure 9 shows the extent and per-rev components. The circumferential extent, plotted on the left, was computed as the maximum tangential section of distorted flow. Resultantly, increasing the resolution decreases extent as the shape of the flow features was brought into focus. The primary flow features should point the distortion descriptors towards a strong 4 per-rev component which was previously shown by List[2,6]. In both probe configurations half of the probes were stationed in the deficit regions and half were in the higher pressure regions. This resulted in an aliasing of the multiple per-rev content as the descriptor was calculated, thus indicating closer to a 2 per-rev distortion.

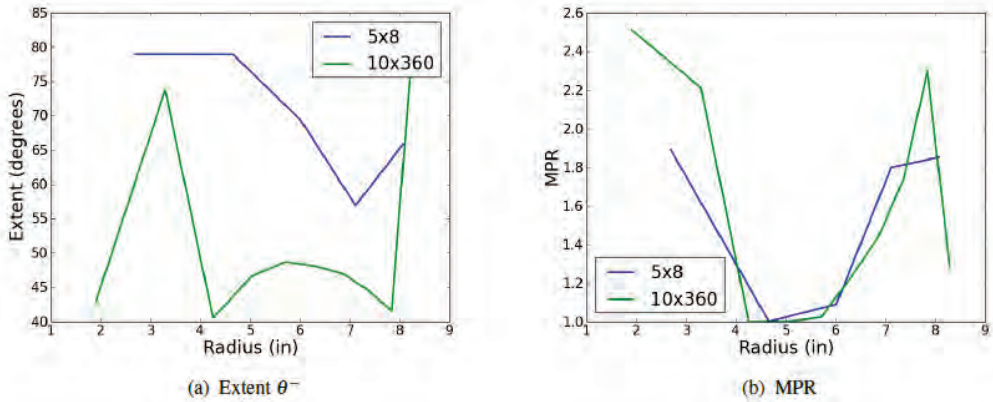


Figure 9. Maximum circumferential extent and multiple per-rev content versus probe ring radial location

3.2 Impact on Experimental Effort

The computational effort has positively impacted the experimental work in several key ways. These are described below.

The performance of the bellmouth was investigated, ensuring that any separations and boundary layers were thin. After indicating that the losses in the designed bellmouth were acceptable an estimated loss factor was provided for use in several diffuser calibration measurements.

The choking condition for the diffuser system was discovered experimentally at a lower throat Mach number than expected. A CFD investigation was conducted using a coarse mesh to rapidly map a range of throat Mach numbers and identify the choking Mach number.

Initial CFD studies were used to determine the necessary size and extent of PIV planes. These data were subsequently employed to size and position optical access windows. The initial studies were also used to determine the rotational resolution necessary for the diffuser. Early identification of these needs drastically improved the initial design phases for hardware and eliminated the need for extensive redesigns after fabrication.

The process of seeding any experiment properly for PIV is difficult, especially when velocity data is sought in separated regions. Such a difficulty was compounded by the physics of the diffuser when trying to seed the TDC vortex. In order to determine the proper placement of the seeding array, it was necessary to perform computations including the upstream flow conditioning barrel. Once this solution was obtained, streamtraces integrated both forward and backward from the regions of interest were utilized to designate seeding locations. It was determined that the locations appropriate for seeding the TDC region of the diffuser exit were small and not intuitive.

The time required to collect time-averaged experimental data at a given point made collection of a full 360° distortion pattern infeasible. To determine if a reduced dataset provided an adequate description of the distorted flowfield, the CFD solution was sampled at various intervals for a sensitivity analysis. The number of tangential positions at which data were to be taken and the initial clocking were varied to compute derivatives of the various distortion parameters. Using this method it was possible to determine that rotation intervals between 3° and 5° were appropriate for resolving the dominant flow features. This resulted in time savings and has dramatically increased the number of conditions quantified experimentally.

4. Conclusion

Computational analysis was used to augment experimental investigations of a modern serpentine diffuser and to better understand distorted flowfields which are ingested by embedded propulsion systems. Such information was beneficial to the experimental program not only in the design and fabrication effort but also in elucidating data collection techniques and requirements.

Cast in terms of the distortion descriptors, the diffuser design creates reasonably well behaved flow relative to the most aggressive designs. This makes the diffuser excellent for the scientific goals of the overall program. All of the expected physics were observed within the diffuser but the total pressure deficit will not be enough to aerodynamically disable the fan to which it will be coupled. This is crucial as a comprehensive investigation of the coupling effects of the combined diffuser-fan system begins.

Although the total pressure deficit was not overly aggressive, the operating environment for a transonic fan in close coupling with a serpentine diffuser is hardly ideal. Implications of the distortion for the fan includes reduced pressure ratio and efficiency due to the nonuniform pressure, varying incidence and additional loss generation due to swirl distortion, and early onset of stall due to additional blockage formation and off-design operation of multiple rotor passages.

The diffuser flowfield simulations using the RANS CFD method showed a difference in the prediction of the separation region compared to the preliminary experimental measurements. This suggests that the LES or DES CFD method could provide better prediction of the position and impact on separation regions on AIP distortion.

Acknowledgments

The authors would like to thank the staff of the AFRL/DSRC for tirelessly working to support, train, and provide resources to the HPCMP users. The authors would also like to thank the HPCMP for its support through generous allocations, an incredible CD-adapco support team, and the CARL experimental team—the members of which work incredible feats in the name of science and technology.

References

1. Copenhaver, W.W., Puterbaugh, S.L., Sanders, D., Nessler, C., and List, M.G., "Overview of serpentine inlet and transonic fan interaction research program." *7th Annual Dayton Engineering Sciences Symposium (Presentation and Abstract)*, 2011.

2. List, M.G., "Distortion Pattern Analysis for Diffuser-Fan Interaction." *Proceedings of the DoD HPCMP Users Group Conference*, 2011.

3. Society of Automotive Engineers, "Inlet Total-Pressure-Distortion Considerations for Gas-Turbine Engines." *Aerospace Information Report AIR1419, Rev. B*, 2007.

4. Society of Automotive Engineers, "A Methodology for Assessing Inlet Swirl Distortion." *Aerospace Information Report AIR5686*, 2010.

5. List, M.G. and Car, D., "Tool and process improvement for high-fidelity compressor simulations." *Proceedings of the DoD HPCMP Users Group Conference*, IEEE Computer Society, pp. 119–126, 2009.

6. List, M.G., "Impact of spatial multi-frequency distortion patterns on fan performance." *7th Annual Dayton Engineering Sciences Symposium (Presentation and Abstract)*, 2011.

Using Kestrel in the Cloud

Josh Calahan
Chenega Federal Systems, University of Utah, SLC, UT
joshua.calahan.ctr@hpc.mil

Todd Tuckey
Air Armament Center, Eglin AFB, FL
todd.tuckey@eglin.af.mil

Alex Cabello and Jason Hirata
Pacific Defense Solutions, Kihei, HI
{alexander.cabello, jason.hirata}@pacificds.com

Steve Caldwell
US Army Engineer Research and Development Center (ERDC), Vicksburg, MS
steve.r.caldwell@erdc.dren.mil

Abstract

The CREATE-AV Kestrel application has been ported to the Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP) Portal. An objective of this new software-as-a-service (SaaS) application environment is to enable acquisition engineers and researchers secure access to Kestrel capabilities without desktop software installation. The Kestrel web user interface is built primarily using the existing Kestrel desktop-client user interface (UI) code, as well as Java Servlet technology, Google Web Toolkit, HPCMP Portal Framework, and Vizify/ezViz software. This paper outlines the Kestrel web workflow, as well as key architecture and design decisions. The Kestrel web UI coupled with the HPCMP SaaS environment hosted at the Maui High Performance Computing Center (MHPCC) will provide significant capability to DoD engineers and scientists without the need to install and configure software or deal with complexity of back-end HPC systems.

1. Introduction

Application services are increasingly delivered via the web browser environment, which has the potential to simplify access, support, and collaboration for both user communities and development teams. This pattern is now being applied to computationally-intensive applications, where the Department of Defense High Performance Computing Modernization Program (DoD HPCMP) is in a unique position to deliver similar application-level capability backed by its significant high performance computing (HPC) resources. The HPC Portal is such an application delivery platform, and aims to provide easy and secure access to key applications running on HPC systems. A goal is to reach out to and serve the larger DoD Science and Technology community, as well as mainstream acquisition. The HPC Portal provides a new way to access computational resources via browsers securely with Common Access Card (CAC)-based authentication and an eye toward other authentication and authorization mechanisms. During the past year, the initial phase of portal development has focused on the foundational infrastructure and initial applications. The CREATE-AV Kestrel application is the first multi-physics application suite ported to this new platform.

Kestrel is a fixed-wing air vehicle simulation suite consisting of three major pieces. The first major piece is its thick-client Kestrel User Interface (KUI), serving as the traditional "face" to the user. The second is the set of components that operate together in a simulation to produce the desired air vehicle behavior. The third major piece is the Kestrel Infrastructure Executive (KIE) that orchestrates a specific instance of components for a simulation. The KUI piece is more than a graphical user interface; KUI is a GUI-engine that is also the basis for other CREATE-AV applications in addition to Kestrel. KUI caters to a class of CFD codes and related multi-physics tools. With KUI, an extensive configuration is supplied to dynamically configure the GUI for a specific set of components and related tools. A configuration directory is composed of input definition XML files, boundary condition scripts, and input validation templates.

Kestrel's flexible design is well-suited for HPC Portal. The KUI was ported to the browser while preserving its extensive configurability. This new piece of Kestrel is called the Kestrel Web UI (KUIWeb), which today lets users setup, modify, visualize, and submit jobs. Kestrel jobs are spawned, managed, and monitored through a generic job management application built atop the foundational portal infrastructure. In this paper, we detail Kestrel job workflow through the lifecycle of a run, from setup to execution and monitoring; we also review key architectural elements of the KUIWeb's design and implementation.

2. Workflow

The HPC Portal is located at https://portal.mhpcc.hpc.mil/[1]. Today there are two methods of authentication. The first is with a DoD CAC and the second is using OpenID against the HPCMP OpenID server. Past the authentication page is the listing of applications, along with ready access to a file manager and a tool to review current and past jobs submitted through any portal application. The first application listed is Kestrel. Launching the Kestrel application brings up the Kestrel Web UI (KUIWeb) in a state ready for preparing new Kestrel inputs or importing existing Kestrel inputs.

2.1 Preparing Kestrel Inputs

The KUIWeb Job Editor is the interface for editing Kestrel inputs. A new input set is started by selecting a specific "use-case" such *Static Simulation* (flow solver only), *Control Surfaces* (flow solver plus moving surfaces), *Prescribed Motion* (flow solver with body motion), et al. (see Figure 1).

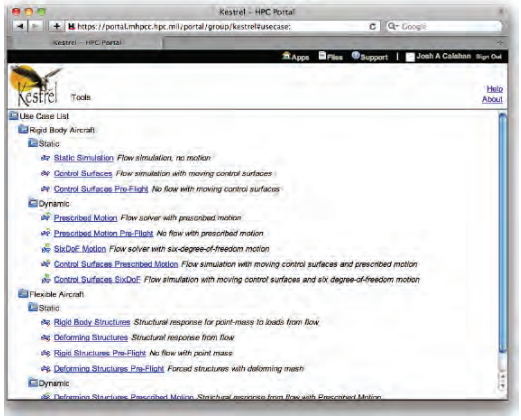


Figure 1. Kestrel Web UI's use case listing

After selecting the desired use case, a tabular listing of components is presented (see Figure 2). The user then enters desired values for each component input.

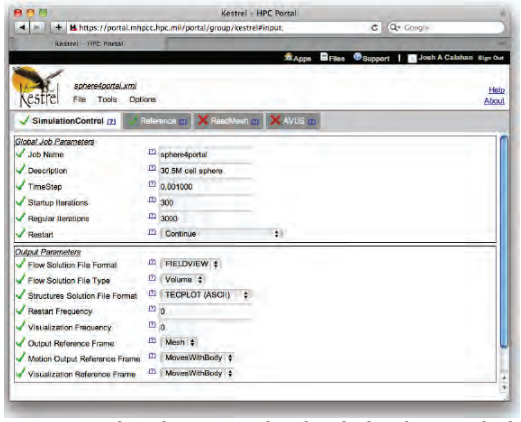


Figure 2. Kestrel Web UI's Static Simulation input tabulation

[1] As of this writing the HPC Portal is in pioneer limited-use phase and not yet open DoD-wide.

As inputs are completed validation occurs and feedback is provided for malformed inputs and inputs that are out of range. For example, a negative Mach number cannot be supplied. Extensive documentation is available for each input and component as a whole. Basic documentation is given as tooltips by hovering on a given input. As inputs are entered, an edit-save-feedback cycle[2] occurs. When an input is changed, metadata of the input along with its value is sent back to the server. Kestrel-specific logic then checks the input for validity and interaction with other related inputs. If the input passes validations, it is committed to disk. Along the way, feedback is supplied to the user via colored "notification toasts". Some inputs are references (or links) to other files, such as computational mesh files; these classes of inputs are handled differently.

The typical Kestrel inputs set references external files, such as computational mesh files, boundary condition files, motion files, and so forth. Typically these files are generated outside the HPC Portal[3] and uploaded via the portal file manager[4]. After external files exist on portal storage they can be linked-up to a specific Kestrel input via a server-side file browser.

2.2 Visualizing Patch Surfaces

KUIWeb contains a Surface Visualizer tool (see Figure 3) for basic visualization of patch surfaces from a computational mesh and associated boundary condition file. This tool is used for basic inspection of surfaces prior to submission of long-running simulations to ensure the correct computational mesh is being used. The surface visualizer is initialized by selecting a computational mesh and associated boundary condition file. By default, all non-symmetry-named, solid-wall-type, patch surfaces are selected for visualization. The user optionally changes the patch surface selections then submits the visualization job. The visualization job runs in a queue for which the HPC Portal contains dedicated visualization nodes, so there is no wait-time unless a large number of visualization jobs are submitted around the same time.

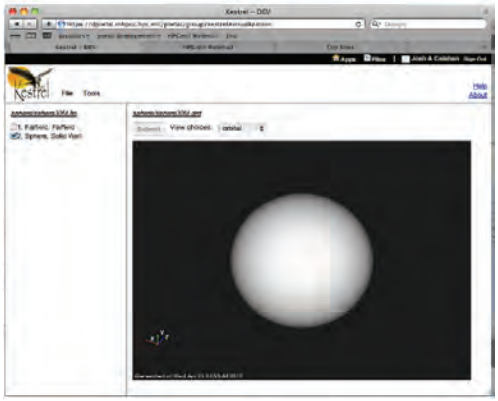


Figure 3. Kestrel Web UI's Surface Visualizer tool showing a 30 million cell mesh

The surface visualizer generates static and animated images. After a visualization job completes, a default "orbital view" animated image is shown. The user may select two other categories of static image sets—orthogonal views and isometric views. Orthogonal view sets show the left, right, front, and back static images. Isometric view sets show the same images, in addition to top and bottom, at a slight angle in order to give perspective.

2.3 Submitting Jobs

Once Kestrel inputs are prepared, jobs may then be submitted from the inputs. When KUIWeb detects that all inputs pass Kestrel's extensive validation checks, it presents an option to let the user submit a job from the set of inputs. Figure 4 shows the transition interface between KUIWeb and the generic job manager. When a Kestrel job is loaded into the job manager application, the user is presented with options to enter information that feeds into the automatically-generated queue submission script such as number of cores, wall-clock upper-limit and an identifying string that serves as a name to

[2] This edit-save-feedback cycle occurs similar to the way Google Documents and Spreadsheets works.
[3] Except for boundary condition files, as there is a boundary condition editor with the Kestrel Web UI.
[4] Files may be uploaded via traditional means (such as Kerberized Filezilla, FTP, or SCP), then placed on the portal storage file system.

the job instance. At this point the job is submitted to a back-end HPC system[5] for parallel execution and the job enters the monitoring state.

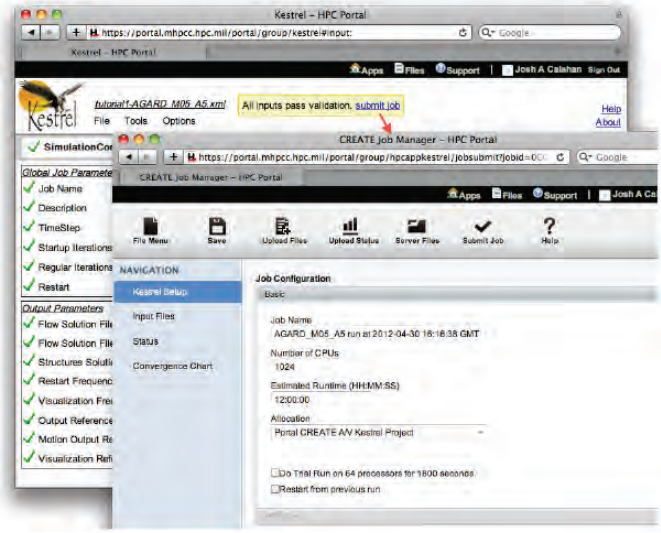


Figure 4. Kestrel Web UI's to Job Manager transition

2.4 Monitoring Job Output

Once a Kestrel job has been submitted for execution there are a few methods for monitoring output. The first is from the job overview application that contains a listing of all current and past jobs. This application shows what state a job is in, such as running or completed. To get detailed status, a job is loaded into the job manager application. This application's status pane shows the current iteration number and the total number of iterations in the form of, say, 10 of 100 iterations complete. A convergent chart of any Kestrel tracking data item can also be automatically generated as shown in Figure 5.

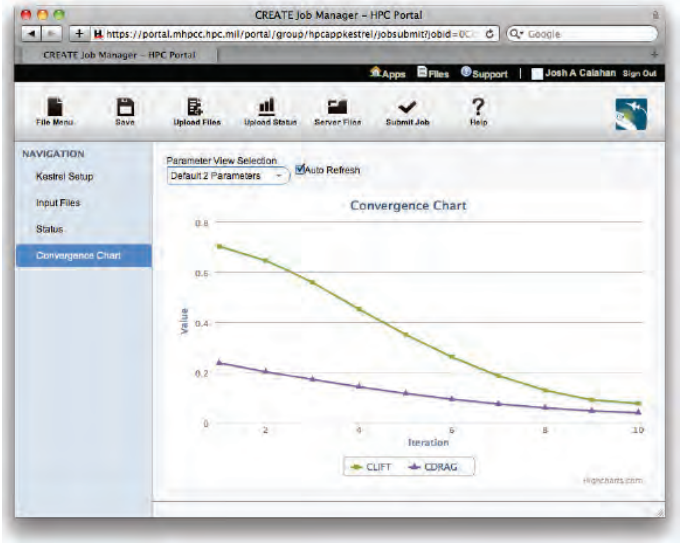


Figure 5. Job Manager Convergence Chart showing lift-and-drag for 10 iterations

When a Kestrel job completed, the job manager application can be used for downloading results. Most importantly, the lightweight tracking and log files can be quickly downloaded. In addition to those, a zip of visualization files may be downloaded. Also, standard output and standard error data may be obtained if needed, such as in the case of a solver crash or to obtain diagnostic information.

[5] Currently MHPCC's Mana Supercomputer is the only back-end.

3. Architecture

The Kestrel Web UI is built on a stable infrastructure consisting of widely-used open-source technologies and packages. First it is built atop Kestrel core libraries, then other supportive libraries that taken together consist of a short-stack that is loosely-coupled to the HPC Portal. The first aim is to apply this architecture to other CREATE applications. The second aim is to merge KUIWeb back into Kestrel.

These aims are sought by keeping core software services separate from crisp interface boundaries. Within this paradigm, the goal is to let core services set below any number of interfaces and permit the interfaces to vary. The most common interface points today are command-line interfaces (CLIs), traditional GUIs, application-programming interfaces (APIs), web apps, and web services. Mobile devices fall into this list as well. This principle of separation is repeatedly applied across KUIWeb internals and back-end pieces, starting with the interface to Kestrel itself.

3.1 Kestrel Java APIs

Kestrel makes extensive use of cross-language patterns that is common in scientific computing today. Scripting languages that drive compiled language parallelized kernels is now the norm. Kestrel implements Python scripting above C/C++/Fortran90 kernels. KUIWeb adds Java to the mix of languages that Kestrel implements. This is a natural choice for web/network/middleware programming and the MHPCC Portal Framework, discussed later. KUIWeb is built primarily on core, existing Kestrel Python code that is very well-tested and already shared between Kestrel's native KUI and KIE (see Figure 6). KUIWeb's interface layer is built on the Model-View-Presenter design pattern that is time-tested and enables a feature-scalable internal design. Coupled with Google Web Toolkit (GWT), discussed next, communication between a web server and a browser is a breeze, as is asynchronous request/response handling, referred to as AJAX.

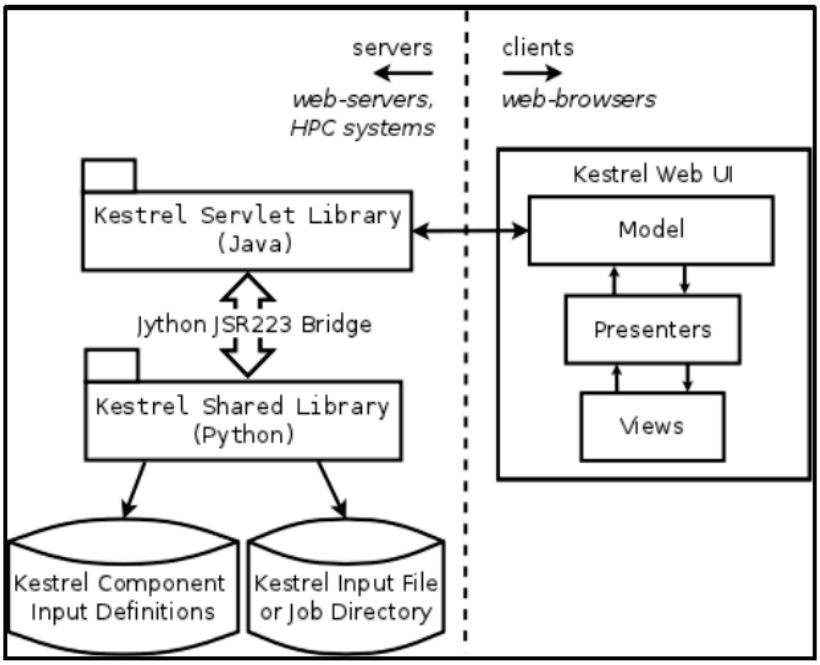


Figure 6. The Kestrel Java/Python binding and their place within KUIWeb

The back-end of KUIWeb, or the piece that runs on the web server, is built on traditional Java servlet technology. Servlets are a mature, standardized, decade-old method of developing web applications. The front-end of KUIWeb, or the part that runs in a browser, is built on GWT's UI libraries. GWT is a stable, 5+ year old library, which many web applications both inside and outside of Google are built upon. Additionally, many open-source libraries are built on GWT. With GWT, application code is written in Java then "compiled down" to the assembly language of the web—HTML, JavaScript, and CSS—such that is portable across browser flavors. For the most part code can be written that works well across everything from older Internet Explorer versions (like IE7) to the latest Chrome and Firefox releases.

3.2 MHPCC Portal Framework

KUIWeb interfaces with the MHPCC Portal Framework. The framework enables deployment to the HPC Portal environment. It handles communication and submission between web servers and HPC compute machines. The portal framework consists of clean and consistent API that abstracts access to jobs, file systems, queuing systems, account/allocation information, user authentication and HPC compute machines.

All portal framework use is on web servers and compute resources. KUIWeb back-end logic calls into framework APIs to access its services. As a representative framework use, for instance, to submit a job the first API call into the framework is to create a job object, then attach application files to the job object, extract the unique key from the job object, then pass the job unique key, and thus the job itself, around to applications that can operate on or view the job.

3.3 ERDC Vizify

KUIWeb's surface visualizer tool implements a VTK-based visualization capability developed by the ERDC visualization team. The implementation is based on ERDC's ezViz/Paraview/VTK stack with bash script add-on known as Vizify. Vizify uses the Kestrel Carpenter mesh manipulation tool for conversion to Ensight format prior to processing by the ezViz stack. Vizify supports static image and basic movie generation, which provides the user with the ability to preview geometry within the portal environment without having to download large data sets. It currently supports visualization of patch surfaces with and without solution overlay. The patch surface of the geometry is integrated into KUIWeb as shown in Figure 3. Solution overlay visualization will shortly be integrated into the job monitoring aspect of the job manager.

4. Significance to DoD

The HPC Portal is a vehicle for deploying capability for wide use across the DoD. The HPC Portal's Cloud-like deployment of Kestrel enables it to be used from any DoD Internet-connected desktop with a CAC-enabled browser. Powerful fixed-wing air vehicle simulation capabilities are now available without any local software installation or configuration. Any engineer in the DoD community, with proper authorization, has near-instant access to Kestrel outside of the traditional HPC batch environment. The Kestrel Web architecture can be applied to other DoD HPC software suites, enabling a larger class of tools to be widely available.

5. Future Work

Kestrel and the HPC Portal, being separate systems, will continue to grow features at different rates. KUIWeb presently supports Kestrel v2, a traditional single-mesh CFD method. Kestrel v3 and up are moving toward a multi-mesh paradigm to support relative body motion, and thus more extensive air vehicle calculations. KUIWeb will need to incorporate its multi-body hierarchical input method over the existing component tabular input scheme. In addition, KUIWeb will support more KUI features such mesh manipulation, system identification plotting and other post-processing features. In time, KUI and KUIWeb will merge into a single UI.

Future development within the HPC Portal will extend to other applications with a deeper emphasis on workflow. Among the areas of workflow improvements identified are case management, more interactive computation, and 3D web-available interactive graphics. It is possible today with the HPC Portal deployment of Kestrel to execute a job at a specific set of flight conditions. With built-in case management, the HPC Portal will enable submission of multiple jobs at a combinatorial set of input conditions (alpha sweeps, for instance) and supply well-organized output for analysis purposes. Today's HPC batch users employ interactive, non-batch, computations that is typically used for pre- and post-processing. Today's HPC Portal implementation supports mostly a batch, submit-queue-run, method of processing jobs. To support better pre- and post-processing, better interactive computational support is needed. Lastly, 3D browser-based graphics are feasible today, and will likely improve, given consumer computing demands. Leveraging such browser-based 3D graphics may permit services that currently can only be done on workstation desktops, such as interactive mesh manipulation and visualization.

6. Conclusion

CREATE-AV Kestrel has been successfully designed and deployed to the web, and is accessible securely from a browser via the new HPC Portal environment. We have demonstrated Kestrel job setup, given a previously-generated computational mesh. In addition, during job setup, we have shown visualization of patch surfaces of a computational mesh without network-intensive downloads. We have also shown how to easily submit and monitor jobs from a Kestrel setup, without the need to manually configure queue submission scripts. Additionally, we have reviewed the key building blocks and design choices in the Kestrel on the HPC Portal implementation, which is applicable across future Kestrel versions and other HPC applications. Lastly, we presented future emphasis areas with the aim to support DoD Science and Technology as well as acquisition engineering HPC workflows.

Acknowledgments

This work was funded by the CREATE Program Office and the MHPCC Portal Project (Doug Post and Dave infrastructure, key direction, and computational resources. We would like to thank Chris Atwood, Scott Morton, Jeff Houchard, and Scotty Swillie for their project leadership. We are grateful to Randall Hand and the ERDC Data Analysis and Assessment Center (DAAC) for support surrounding patch surface visualization. We would also like to thank the MHPCC administrative, security, and technical staff—too numerous to name—for their key support of this effort.

References

Morton, S., D. McDaniel, D. Sears, B. Tillman, and T. Tuckey, "Kestrel – A Fixed-Wing Virtual Aircraft Product of the CREATE Program." *DoD High Performance Computing Modernization Program 2009 Users Group Conference*, San Diego, CA.

Hand, R. and P. Adams, "ezViz – An Open-Source, Cross-Platform Visualization Tool." *DoD High Performance Computing Modernization Program 2005 Users Group Conference*, Nashville, TN.

Hirata J. and A. Cabello, "HPC Portal Framework Documentation Wiki." Maui High Performance Computing Center, https://portal.mhpcc.hpc.mil/portal/group/developer/wiki/-/wiki/Main/Portal+Framework+Wiki, May 7 2012.

Morton, S., T. Eymann, S. Lamberson, D. McDaniel, D. Sears, J. Utrilla, and T. Tuckey, "Relative Motion Simulations Using an Overset Multi-mesh Paradigm With Kestrel v3." *50th AIAA Aerospace Sciences Meeting and Exhibit*, Nashville, TN, January 2012.

Verification of a GPU Enabled 3D Solver for Hypersonic Flight Analysis

Evgeny Titov
Dynamics Research Corporation, High Performance Technologies Group, Fairborn, OH
etitov@drc.com

Jonathan Burt and Eswar Josyula
US Air Force Research Laboratory, Air Vehicles Directorate, Computational Aerodynamics Branch (AFRL/RBAC), Wright-Patterson AFB, OH
eswar.josyula@wpafb.af.mil,
jonathan.m.burt@nasa.gov

Abstract

In this paper, we discuss graphics processing unit (GPU) implementation and results for a computational fluid dynamics code designated as the US Air Force Research Laboratory perfect gas solver[1]. We also verify and validate the code by comparing its results with results of a similar tools including GASP[2]; experimentally-validated N-S solver whose results are provided by Nompelis[3], and a particle HAP-DSMC[4] code appropriate for rarefied flows. We find that for graphics processing units (GPU) implementation to be efficient, a sizable part of the code needs to be redesigned and only large computational cases with enough GPU load may be attempted.

1. AFRL Code GPU Implementation and Testing

1.1 AFRL GPU Implementation

After initial profiling of the baseline US Air Force Research Laboratory (AFRL) code, it was found that subroutines responsible for inviscid flux computations, driven by a major loop need to be targeted for graphics processing unit (GPU) acceleration. These subroutines whose names are: FSIROE, FSJROE, TRANSP, LMTRI, LMTRJ SOURCE, and L accounted for ~80% of the computational time.

After experimenting with various approaches, including the PGI accelerator, PGI CUDA FORTRAN, Nvidia CUDA framework, and OpenCL, two of them were identified as the most promising. These were the PGI Accelerator and OpenCL. Both approaches required refactoring and partly re-designing of the original AFRL code. The PGI implementation is now complete and reported here.

Upon initial implementation and testing, it was found that because of external calls from the major program cycles the implementation cannot be efficient. The original L subroutine that is the major driver for the computational loop was then replaced by two separate subroutines L1 and L2. Each of these subroutines currently contains independent loops over I and J procedure indexes. All the calls from FSIROE and FSJROE subroutines to the limiters (LMTRI and LMTRJ) and other subroutines and functions are in-lined, making the major loops applicable for GPU parallelization. After testing this alternative implementation with the PGI accelerator it appeared that more time was spent on data transfer between GPU and the main memory than on computations, which made the code inefficient. To avoid the data transfer overhead, the FSIROE and FSJROE subroutines were re-designed to copy all the required data into the GPU memory before the main cycles began, and transferring it back after the end of the computational loops. It was not possible to implement this approach fully because of partial inter-dependence of the resulting kernels on the data that is executed. The approach however worked to a degree to make the GPU implementation efficient.

For GPU implementation testing, a configuration involving flow in a long nozzle was chosen and the results are illustrated in Figure 1. The reason for selecting a long nozzle flow problem was as follows: for a GPU implementation to be effective, it is necessary to load the GPU processing units with a large number of computations per kernel invocation. In the case of computational fluid dynamics (CFD) codes this requirement calls for larger computational meshes. Unlike the blunted wedge cases, considered further in this paper, where relatively coarse meshes are sufficient for spatially-resolved solutions, long nozzle cases require larger meshes of up to 2,500×300 mesh points making them better candidates

for efficient GPU implementation. The solution presented in Figure 1 has an interesting feature that cannot be captured without high-mesh resolution and solution-adaptation. A weak shock-wave, which originates at the sharp corner of the nozzle throat, propagates through the nozzle and reflects several times from the nozzle walls and its centerline. The shockwave is observed in the LHS of Figure 1 where density gradient contour plots are presented. The shockwave also causes local spikes of flow parameters at several locations along the center line of the nozzle, as illustrated by the center line density plot presented in the RHS of Figure 1.

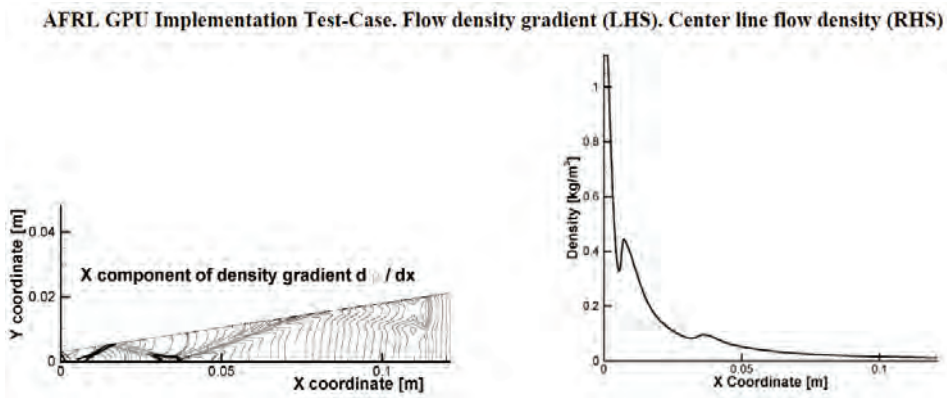


Figure 1. Long nozzle GPU test-case

1.2 PGI Accelerator Version Testing

A typical GPU version compilation result of the AFRL code using PGI accelerator is presented by the following listing:

```
Here is the compiler message:
713, Generating copyin(y(:,:))
     Generating copyin(x(:,:))
     Generating copy(fs1v(:,:,:))
     714, Generating compute capability 2.0 binary
     716, Loop is parallelizable
     718, Loop is parallelizable
          Accelerator kernel generated
          716, !$acc do parallel, vector(8) ! blockidx%y threadidx%y
          718, !$acc do parallel, vector(32) ! blockidx%x threadidx%x
               CC 2.0 : 7 registers; 8 shared, 48 constant, 0 local memory
bytes; 100% occupancy
     727, Loop is parallelizable
          Accelerator kernel generated
          727, !$acc do parallel, vector(256) ! blockidx%x threadidx%x
               Cached references to size [256x2] block of 'y'
               CC 2.0 : 34 registers; 4116 shared, 80 constant, 0 local memory
bytes; 50% occupancy
     763, Loop is parallelizable
          Accelerator kernel generated
          763  !$acc do parallel, vector(256) ! blockidx%x threadidx%x
               Cached references to size [256x2] block of 'y'
               CC 2.0 : 34 registers; 4116 shared, 84 constant, 0 local memory
bytes; 50% occupancy
     800, Loop is parallelizable
          Accelerator kernel generated
          800, !$acc do parallel, vector(256) ! blockidx%x threadidx%x
               Cached references to size [258x2] block of 'x'
               Cached references to size [258x3] block of 'y'
               CC 2.0 : 63 registers; 10324 shared, 196 constant, 0 local
memory bytes; 33% occupancy
```

As can be seen from the listing, the compiler reports that all attempted loops (in this case FSIROE loops) are vectorized with data moved into the GPU memory at the beginning of the program, and that kernels are created for each of the loops. The profiler timing results for 100 steps of a typical run report the following time usage:

```
Accelerator Kernel Timing data
/Network/Servers/xserve4.rb.rad-e.wpafb.af.mil/Volumes/RBAsan/Users2/titove/ PGI/NH5Air/NH5Air/main.f
  fsiroe
    714: region entered 680 times
      time(us): total=1027407 init=101 region=1027306
          kernels=240898 data=0
      w/o init: total=1027306 max=2709 min=1434 avg=1510
      718: kernel launched 680 times
        grid: [2x2]  block: [32x8]
        time(us): total=6193 max=28 min=9 avg=9
      727: kernel launched 680 times
        grid: [1]  block: [256]
        time(us): total=11090 max=21 min=16 avg=16
      763: kernel launched 680 times
        grid: [1]  block: [256]
        time(us): total=10327 max=20 min=13 avg=15
      800: kernel launched 680 times
        grid: [1]  block: [256]
        time(us): total=213288 max=325 min=311 avg=313
/Network/Servers/xserve4.rb.rad-e.wpafb.af.mil/Volumes/RBAsan/Users2/titove/ PGI/NH5Air/NH5Air/main.f
    fsiroe
      713: region entered 680 times
        time(us): total=1920696 init=584505 region=1336191
          data=124721
        w/o init: total=1336191 max=3601 min=1853 avg=1964
real    0m3.203s
user    0m0.629s
sys     0m2.409s
```

As can be seen in the profiler output, most of the GPU time is spent on computations, with minimum time spent on data transfer. However, when applied to a typical problem with a mesh size of some 200×100 points, the implementation yields a negative speed-up (~1.5 times slower as compared to an advanced Intel Xeon CPU computations). Further studies revealed the following cause: GPU implementation is correct, but because of the dataset size being small, kernels do not have enough work to utilize GPU processing elements power efficiently.

To verify this finding, a larger computational case was designed. This case, whose density gradient contour plots and center line density values are presented in Figure 1, was a long nozzle. To capture the effects, such as propagation of weak shock-waves originated at the nozzle throat, a larger computational mesh was required. Two computational meshes were created for the case. A coarse mesh consisted of 202×52 points and a larger one was 2,500×300, where the larger mesh dimension is along the nozzle axis. The results for 10 iterations over the larger mesh are as follows:

- On a GPU Tesla C2070 at 1147Mhz accelerator; wall-time was 17.2 sec.
- On Intel® Xeon® CPU 5160@3.00 Ghz wall-time 30.4 sec.

This constitutes an overall acceleration of 1.76 times.

Since the accelerated programs accounted for 80% of the original CPU time the actual acceleration of the subroutines that were modified was ~2.18.

For the coarse mesh of 202×52 points the speed-up was negative.

2. Selected Verification Cases

A series of two cases with increasing complexity were selected for verification and validation of the AFRL code. Two different inlet pressure conditions were identified that yield distinct high- and low-pressure flows that exhibit unique physical characteristics. Solution of GASP[2] and results provided by Ioannis Nompelis's[3] were used for verification of the AFRL code. Particle HAP-DSMC[4] solver was also used.

An N-S solver called GFSProto[5] was also evaluated, but not utilized for AFRL code verification. GFSProto, a Cartesian mesh-based N-S solver employs a newly implemented immersed boundary condition (IBM) that is of interest for the types of flows studied, and the results of this capability were evaluated as part of this work.

The two cases were designed to represent the limiting flow conditions with respect to applicability of the N-S and DSMC techniques. The case, referred to as "Low Pressure Case" or LPC, was designed to produce a substantial rarefaction for an easier DSMC solution; and the second case, identified as "High Pressure Case" (or HPC), was chosen to be of the same geometry as the LPC, but with the free-stream flow conditions at the limit of DSMC applicability in terms of practically available computational resources. The inlet flow conditions for the LPC and HPC cases are provided in Table 1.

Table 1. Test series free-stream flow properties

Series	LPC	HPC
Temperature [K]	191	191
Pressure [Pa]	20.13	402.67
Mach number	7.23	7.23
R	286.9	286.9
Gamma	1.4	1.4
Tw [K]	300	300

Figure 2 illustrates the solution for the HPC result obtained with GASP for a blunted-wedge configuration. In the LHS of the figure the temperature contour plots are presented, and in the RHS the gradient-based local Knudsen numbers are shown. To obtain the local Knudsen numbers, equations 1–3 were used, where: Kn - is the gradient-based Knudsen number; L - is characteristic length scale; λ - is mean free-path of molecules; μ - is viscosity; ρ - is density; m - is mass of molecule (averaged over species); k - is the Boltzmann constant and T - is temperature.

$$Kn = \frac{\lambda}{L} \tag{1}$$

$$\lambda = \frac{\mu}{\rho}\sqrt{\frac{\pi m}{2kT}} \tag{2}$$

$$L = \min\left\{\frac{\rho}{\nabla\rho};\frac{T}{\nabla T};\frac{a}{|\nabla V|}\right\} \tag{3}$$

As can be seen in the RHS of Figure 2, the HPC develops thin areas of high local Knudsen numbers in the shock-wave and in the boundary-layer. This effect is due to the large flow gradients in these areas. Rarefaction in the boundary-layer, reflected by the elevated local Knudsen numbers, may produce a velocity slip effect. This effect is studied separately in detail and will be reported in the following publications.

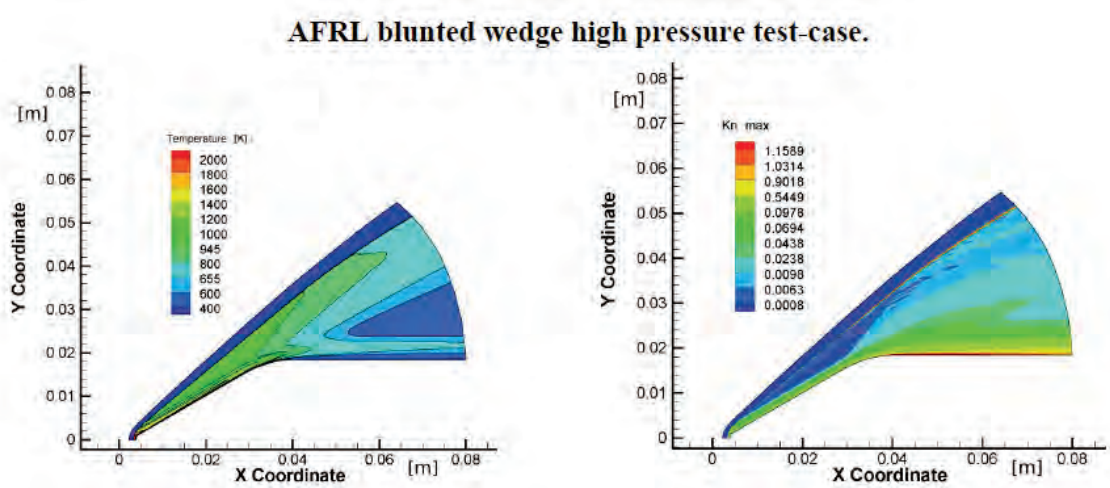


Figure 2. AFRL blunted-wedge high test-case; Temperature (LHS); Local Kn number (RHS)

2.1 Low-Pressure Solution Verification

Results of code verification for the blunted wedge LPC whose flow conditions are provided in Table 1, are presented in Figure 3 which also shows results of the following flow solvers: N-S GASP, GFSProto, and DSMC HAP. The DSMC computational domain was shortened to exclude the flow expansion region and molecular nitrogen was also used as the DSMC working fluid, whereas a perfect gas was assumed for all other cases. Figure 3 shows the surface pressure and heat flux obtained with all of the utilized solvers. The N-S solutions obtained with these different solvers are expected to coincide with mesh resolution sufficient for mesh convergence. Small deviations are anticipated in the case of the DSMC result due to molecular nitrogen modeling.

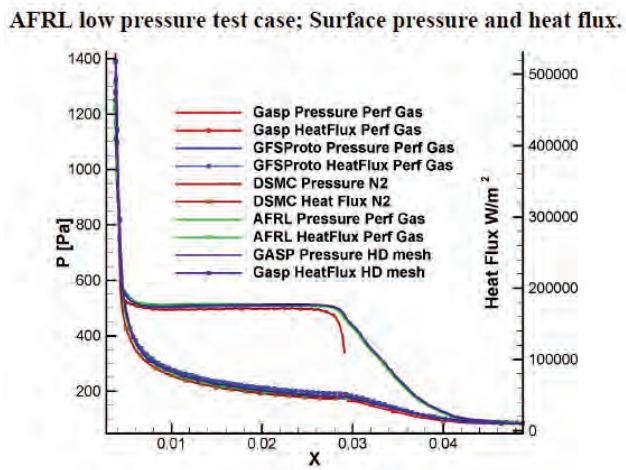


Figure 3. Low pressure verification test-case

In order to demonstrate mesh convergence for the N-S results, three computations with systematic mesh size variation were performed with both the AFRL and GASP codes. The first two solutions were obtained with AFRL and GASP using the same mesh with the cell-based Re number at the wall of near unity. The third case was solved with GASP using a much denser mesh giving the cell-based Re number at the wall of near 0.01. As can be seen in the figure, all three cases show identical pressure distributions along the surface while the surface heat-flux for the high-density mesh case (denoted as HD mesh) deviates only slightly from the coarse mesh solutions.

A GFSProto solution for this case was obtained with an automatically adapted Cartesian mesh that produced cell-based Re numbers at the wall of near-unity. The pressure distribution obtained with the GFSProto coincides with the results of the other two N-S solvers whereas the heat-flux is some 3–5% larger throughout the surface. This difference needs to be studied, and is probably due to the specifics of the immersed boundary condition implementation in GFSProto. Immersed boundary techniques are known to introduce assumptions in the near-wall region. Also a denser mesh case utilizing GFSProto needs to be run to further study the mesh convergence since the oct-tree-based meshing may pose different convergence requirements as compared to the body-fitted meshes used in the other N-S solvers.

Finally, a DSMC solution was obtained for the case and compared with the results of the N-S solvers. As can be seen in the figure, the DSMC solution differs somewhat due to nitrogen modeling from that of N-S solvers, but is qualitatively close and serves as additional verification of the AFRL results.

2.2 High Pressure Solution Verification

For the HPC blunted-wedge case whose flow conditions are presented in Table 1, solutions obtained by GASP, GFSProto, AFRL, and Ioannis Nompelis's provided data[3], are presented in the LHS of Figure 4. To test the mesh convergence a GASP case with a very dense, gradient-adapted mesh was used. The mesh guaranteed cell-based Re numbers on the order of 0.01 at the areas close to the surface as presented in the RHS of Figure 4. The dense mesh solution did not show significant difference as compared to a mesh with cell-based Re numbers at around 1.0. Therefore, convergence was obtained.

In a manner similar to the LPC, the results for the HPC are presented in the LHS of Figure 4 for surface heat-flux and pressure. All the codes produce close results verifying the AFRL code as accurate for sensitive, gradient-based, flow properties such as heat-flux. GFSProto case solved a smaller computational domain around the nose tip of the blunted-

wedge, and provides results in this area that are close to the other applied solvers. GFSProto results are not considered for verification in this paper; however, the Cartesian mesh implementation is an interesting alternative, and the results are presented for the interest of the reader and for verification of the GFSProto itself.

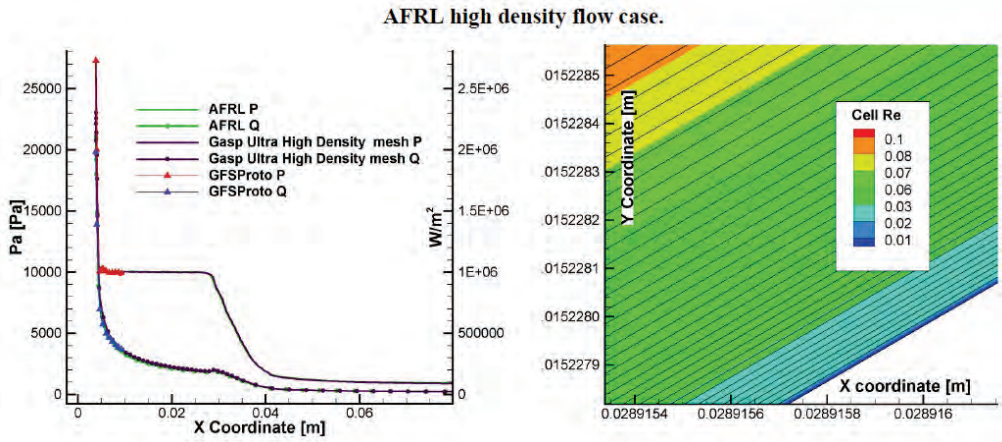


Figure 4. Surface pressure and heat-flux (LHS); Local Re number (RHS)

2.3 Double Cone Case

Figure 5 illustrates a solution for the second verification configuration, a double-cone flow. This case is more challenging due to a complex shock-wave and shock-wave—boundary-layer interactions, as well as shock-wave-induced boundary-layer separation and re-attachment. The basic flow structure is illustrated in the LHS of Figure 5 where the Mach number contour plots are presented and the flow features are described. The effects of strong non-equilibrium are illustrated in the RHS of Figure 5, where the local Knudsen number contour plots are presented. The shock-wave structure, as well as the shock wave-boundary layer interactions is visually articulated in the RHS of Figure 5 by the high (red)-levels of the local Knudsen number. This complex flow configuration makes a good verification case for the AFRL code when sensitive flow quantities such as wall heat-flux are compared with experimentally-validated results of Nompelis[3]. A Gasp solution for the case was also obtained and compared with the results of the other two codes. The inlet flow conditions for the studied case, as well as boundary conditions at the surface of the double-cone are presented in Table 2.

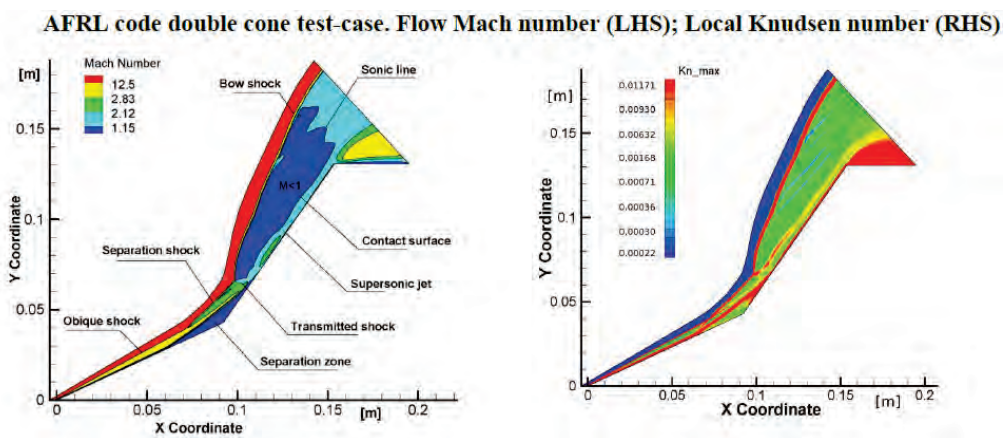


Figure 5. Complex geometry verification test-case

Table 2. Double cone flow conditions

Temperature [K]	98.3
Pressure [Pa]	16.49
Mach number	12.8
R	286.9
Gamma	1.4
Tw [K]	300

The shockwave-boundary-layer interaction developed by the flow produces peaks of the surface loads at the locations of such interactions. Figure 6 shows surface pressure (LHS) and heat-flux (RHS) computed with the three applied solvers. It can be seen in Figure 6 that the AFRL and Nompelis's data compare well for pressure values at all locations along the surface and the GASP solution deviates slightly in the downstream portion of the flow. More sensitive heat-flux results that depend on the temperature gradients at the surface also agree well between the AFRL and Nompelis in the more critical area of elevated heat-flux levels and deviate somewhat in the downstream area. The GASP results produce larger peaks for the heat-flux and also deviate insignificantly from the results of the other codes in the downstream area. It should be noted that for this complex case, the mesh convergence was not studied; however, the same gradient adapted mesh was used with all three solvers. While using the same mesh and the same (second) order of accuracy suggests that qualitatively similar results are obtained, the details of the numerical implementation may impact the agreement. Given that the mesh convergence needs to be studied, the level of agreement between the solutions is satisfactory and serves as a more complex verification case for the AFRL N-S solver.

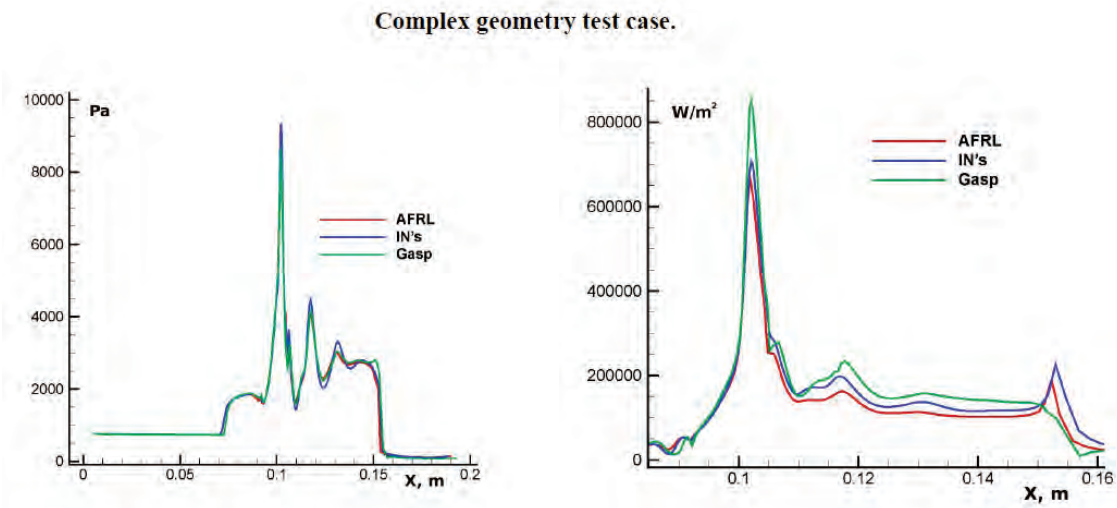


Figure 6. AFRL double cone test-case; Surface pressure (LHS); surface heat-flux (RHS)

3. Summary

The following results are obtained and discussed:

- Results of testing of the GPU implementation of the AFRL code for a long nozzle flow configuration that provides enough computational load for GPUs.
- Results of verification of the AFRL code by the results of two different N-S solvers and a DSMC solver for a blunted-wedge configuration typical for the AF applications.
- Results of verification of the AFRL code by results of the experimentally-validated Ioannis Nompelis's[5] N-S solver for the double-cone configuration.

4. Future Work

During the course of this work a subject of an interest has emerged. The main test cases studied in this report included blunted-cones and blunted-wedges and the flow regimes were such that the slip velocity was significant and is expected to alter the surface heat-fluxes. It was found that N-S solutions for such flows may not be accurate at the leading-edges, even for relatively high-pressure cases.

Sharp leading-edges, on the other hand, provide substantial aerodynamic advantages at high-supersonic and hypersonic speeds. The limitation to the sharpness of the edge comes from the heat-fluxes which the leading-edge material can sustain before failure. This is because the heat-flux increases with decrease of the leading-edge radius.

This makes an accurate prediction of the heat-fluxes into the surface essential for AF applications. Since thermal conductivity of typical hypersonic leading-edge materials is low suggesting low-heat removal by conduction, a radiative heat balance can be used to compute the local surface temperatures. Therefore, it is computationally possible to study the

accuracy of N-S based techniques in prediction of the leading-edge heat-fluxes for edges of practical radiuses. If it is shown that the N-S technique is not adequate for the leading-edge heat-flux prediction for leading-edge radiuses which the material can sustain, then more accurate kinetic approaches will be needed. This project has a potential of providing such assessment. This topic will be studied within the context of this project with available resources.

References

1. Josyula, E., "Computation of Hypersonic Flow past Blunt Body for Nonequilibrium Weakly Ionized Air." *AIAA 93-2995*, AIAA 24th Fluid Dynamics Conference, Orlando, FL, July 6–9, 1993.

2. *GASP Version 5.1 Technical Reference*, Aerosoft, Inc., 2011.

3. Nompelis, Ioannis, *Computational Study of Hypersonic Double-cone Experiments for Code Validation*, A doctoral theses, University of Minnesota, 2004; Results presented in this paper are provided through a private communication in year 2011.

4. Burt, J.M., E. Josyula, and I.D. Boyd, "A Novel Cartesian Implementation of the Direct Simulation Monte Carlo Method." *AIAA paper 2011-632*, 2011.

5. Arslanbekov, Robert. R., Vladimir I. Kolobov, and Anna A. Frolova, "Analysis of Compressible Viscous Flow Solvers with Adaptive Cartesian Mesh." *AIAA 2011-3381*, 20th AIAA Computational Fluid Dynamics Conference, Honolulu, HI, 27–30 June 2011.

6.2 Applied Research

Army

A 3D Studio Max Rendering Approach to Achieve Optimal Single-Dimension Geometry Exaggeration Necessary in Some Scientific Visualization Analysis Scenarios

Richard I. Walters and Mary A. Cialone
US Army Engineer Research and Development Center (ERDC), Vicksburg, MS
{richard.i.walters, mary.a.cialone}@usace.army.mil

Abstract

Some visualization requires stretching an X-, Y-, or Z-dimension of a dataset in order to detect small features that otherwise would go unnoticed. This is typical in oceanic datasets where square miles of water area are in view, but surface characteristics, based on a few feet, are the focus of the study. In these cases, the height of the surface geometry needs to be uniformly up-scaled to see these variations. Producing a very high-quality visualization typically follows a three-step approach: 1) extract geometry from data, 2) import that geometry into a rendering package, and 3) produce images for the animation. For scenarios where exaggeration is required, scaling is normally done during the first step; the data are extracted and exaggerated simultaneously. But this three-step approach has inherent problems; one has to scale by some amount, before seeing the final visualization playback. Intelligent guesses are used to help decide an amount to exaggerate, based on examining key dataset times when maximum and minimums occur. However, despite these best guesses, it is typical that a different amount of exaggeration could have been better after seeing the final visualization. Correction becomes a time-consuming, wasteful, and iterative process, where previous rendered images are discarded as a more effective exaggeration amount is used. This paper presents a fast, dynamic method for selecting an effective exaggeration amount by utilizing some image-processing capabilities of AutoDesk's 3D Studio Max rendering software. This technique uses a series of high-resolution height-mapped images rather than traditional fixed-polygonal geometry. Furthermore, this method allows playback preview for the amount of exaggeration (in wireframe) before any images are actually rendered. Refining the amount of exaggeration and previewing the results can be accomplished in seconds. In this paper, we demonstrate this technique using a Great Lakes Federal Emergency Management Agency (FEMA) dataset from a study that furthers the understanding of how meteorological events play a role in flooding.

1. Introduction

The US Federal Emergency Management Agency (FEMA) Region V has undertaken the task of updating the Great Lakes flood risk maps, and has contracted with the US Army Engineer Research and Development Center (ERDC) to develop wind, wave, and water-level climates to define the storm-event population that will be used to estimate the potential for coastal flooding in Lake Michigan. The techniques and procedures developed by ERDC will then be applied to the remaining Great Lakes by other FEMA contractors. The approach being taken is to simulate wave and storm surge conditions associated with 150 meteorological events via computer modeling. Storm surge and wave models, ADCIRC (Ceyhan, Basuchowdhuri, Judeh, Ous, Estrade, and Kosar, 2007) and WAM (Zhang, Dekate, Allen, Kelly, and Maclaren, 2007), respectively, were applied to the entire Lake Michigan/Lake Huron system for each event-specific wind, pressure, and ice fields to calculate storm-surge and wave information. These modeled storm water-level and wave data form the foundational database to be used in updating the Great Lakes flood risk maps.

2. Data

The Data Analysis and Assessment Center (DAAC) were requested to visualize the results of an ADCIRC storm event simulation. The simulation modeled an actual storm that lasted 12 days, from 25 November to 07 December 1990. The boundaries of the simulation were restricted to the coast lines of these two Great Lakes for simplicity, even though this storm event did produce flooding in the Green Bay area. (Future improvements will be made to model inland inundation.) A customized Visualization Toolkit (VTK) reader was used to translate the ADCIRC data into Legacy VTK format (Kitware, Inc., 2010), which enabled it to be read into ParaView, a free open-source visualization package from Kitware for processing. The mesh consisted of 409,264 nodes and 777,626 cells, and was greatly refined along the coastline edges of the Great Lakes as shown in Figures 1 and 2. Water-level solutions, as well as other data parameters, were recorded every 5 minutes over the 12-day event, yielding 3456 time-steps. The total size of the data was 46GB.

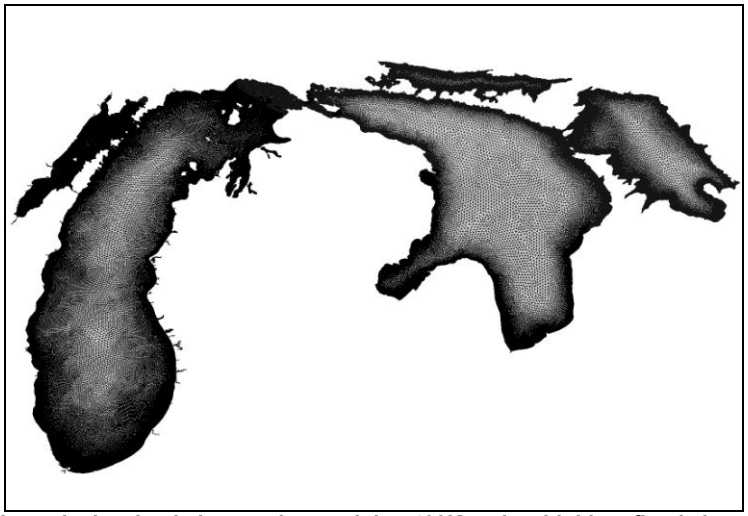


Figure 1. Shown is the simulation mesh containing 400K nodes, highly-refined along coastlines

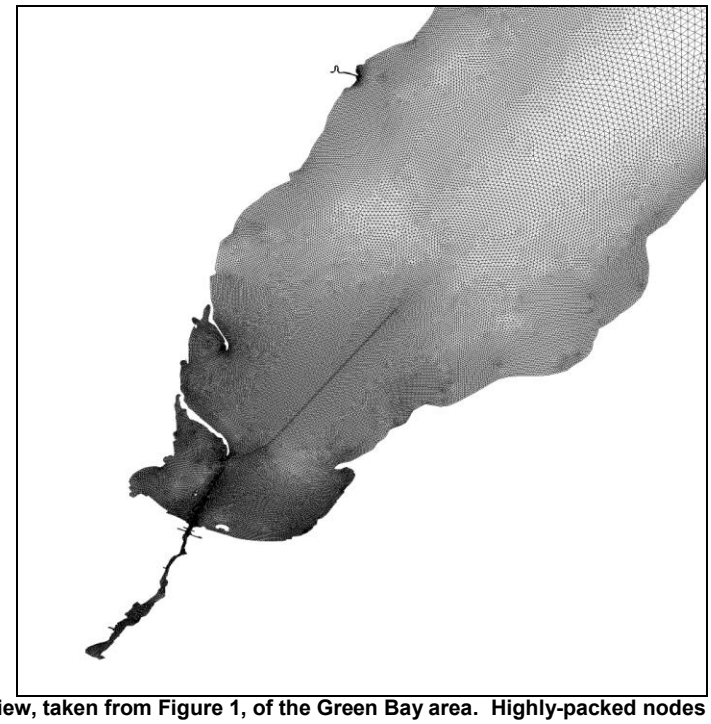


Figure 2. This is a detailed view, taken from Figure 1, of the Green Bay area. Highly-packed nodes capture details of water inlets and tributaries. The Fox River is seen extended from the bottom of the grid.

3. Visualization Goals

The goals of the visualizations were to relate insights gained from the simulation. This study focused on two main areas of interest. First, understanding how oscillatory modes can be induced by wind and atmospheric pressure gradients over the Great Lakes regions, including how those modes affect water-level fluctuations across these large lakes. A second point of interest included analyzing flood threats resulting from water-level fluctuations. Analyzing the first focus point required a view of all Lake Michigan and Lake Huron from a distant viewpoint. The second focus of the study needed visualizations isolated to the Green Bay area where maximum flooding occurred. A storyboard was developed that related these gained insights. It required four visualization playbacks, each from a different viewpoint, which can be seen in Figure 3. The first view incorporated the entire simulation, showing Lake Michigan and Lake Huron. The second viewpoint zoomed in to focus on the entire Lake Michigan area. The third playback stepped even closer, focusing on the Green Bay area, where surges demonstrated greatest water-level variations. The final playback used a perspective view of the Fox River area, showing where maximum flooding was experienced from the storm event.

A set of Google Earth Pro fly-through images were also rendered to contextualize the data geographically for the viewer. These connecting segments were used to tie each of these four simulation playbacks together in order to present a single, seamless visualization analysis. The fly-through incorporated the four viewpoints as key frames for its path. Furthermore, the fly-through paused momentarily at each key frame, where the four visualization playbacks were inserted. The results looked seamless and smooth due to the fact that the fly-through used a nice deceleration/acceleration between each stop. Additionally, an Adobe After Effects script was used to label day, hour, and minute of the event onto each frame.

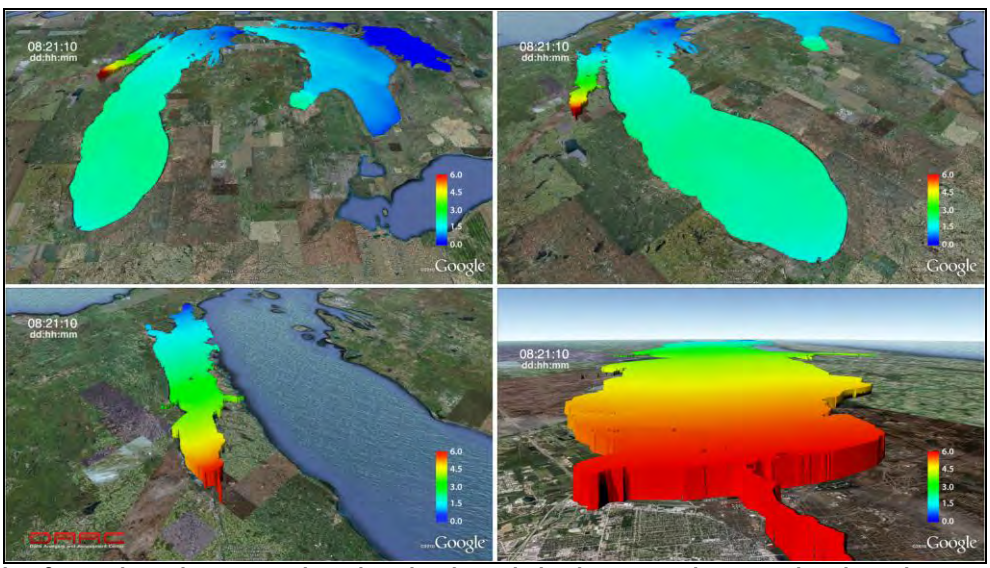


Figure 3. Four key frame viewpoints were selected to play through the dataset; each successive viewpoint zoomed in closer to study the most intense flooded area. The top left image shows simulation data overlaid onto Google Earth images with Lake Michigan and Lake Huron in view. Top right viewpoint focuses on Lake Michigan. Bottom left isolates a perspective view focusing on Green Bay. Bottom right image allows both color and exaggeration to convey storm-surge flooding threats.

Although it was not absolutely necessary to exaggerate the surface height of the water-levels, as differences could be observed by color cues alone, doing so greatly provided a more intuitive understanding of the oscillatory modes that were created by the storm. Watching the actual water-levels rise and fall made it much easier to understand the problem than simply watching color bands propagate through these Great Lakes.

4. Traditional Visualization Approach

The straightforward approach to produce very high-quality visualizations from these data would typically require three steps: extracting geometry from applications that can read ADCIRC data; importing that geometry into a high-quality rendering application such as 3D Studio Max; and encoding rendered images into animation clips. However, this straightforward process has inherent difficulties associated with guessing an optimum amount to scale the water height

before seeing the final animation clips. Intelligent guesses could be made based on experimentation using data times where maximum and minimum water-levels occur in the simulations. However, despite best guesses, the results can often look different in the software where geometry extraction and exaggeration are performed, such as in ParaView, than they do in software where the final frames are rendered, such as in 3D Studio Max. Additionally, viewing the results of a single time-step is sometimes misleading; results may look good for that one frame, yet not be pleasing over the entire 3,456 time-steps. Furthermore, the process of determining the correct amounts of exaggeration must be repeated for each of the four views, since they are vastly different. Factors such as these make it difficult to obtain an optimal amount of exaggeration on the first guess. Correction becomes a time-consuming and iterative process, where previously rendered images are discarded as a more effective exaggeration amount is used to repeat the process.

5. Optimized Visualization Approach

To overcome obstacles associated with guessing effective degrees of exaggeration, we present a fast, dynamic method for obtaining optimal exaggeration amounts. Instead of importing the simulation data into the rendering software as a series of time-based geometry files, we imported a series of height-mapped images instead; this has many advantages. 3D Studio Max is capable of deforming a flat surface based on a grayscale image; dark shades pull geometry vertices downward, while lighter shades pull geometry vertices upward; fifty-percent gray shades leave the geometry unchanged. This capability is accomplished in 3D Studio Max by selecting a flat piece of flat geometry and activating a *displacement modifier* to act on it. The geometry modifier works in combination with a grayscale image that is mapped against the geometry. Figure 4 shows a simple example of this process. Displacement modifiers can be controlled with a number of parameters. The amount of deformation can be controlled by entering a numeric value into its "strength" field; the higher the number, the more the geometry is deformed. 3D Studio Max is also capable of deforming the same flat geometry in a series of time-steps based on a series of grayscale images.

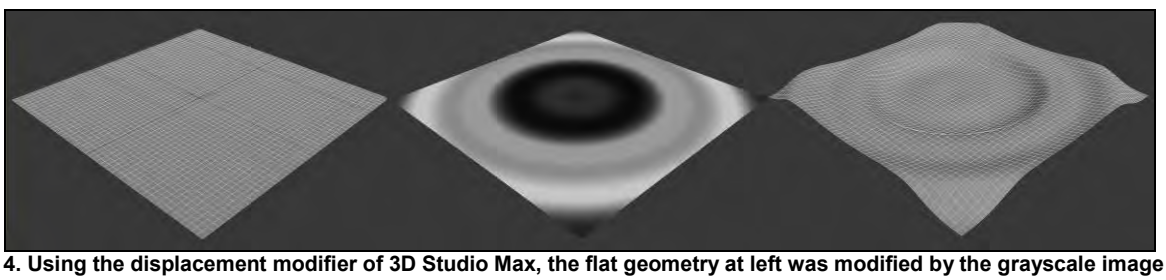


Figure 4. Using the displacement modifier of 3D Studio Max, the flat geometry at left was modified by the grayscale image in the center to produce the final geometry at the right. In the same way, grayscale water-level images can be used to exaggerate the surface geometry in the z-dimension to a customized amount.

This method of deforming geometry was applied to our dataset. A single-surface geometry file of the Great Lake simulation was flattened and imported into 3D Studio Max. This baseline geometry would be deformed to match the water-levels at each time-step of the simulation. Care was taken to align it in 3D Max. Once it was correctly aligned, 3D Studio Max's displacement modifier was applied to the geometry which deformed it based on a series of very high-resolution grayscale images that recorded the water-levels of each time-step of the simulation. ParaView was used to obtain these images; Figure 5 shows an example image from the overhead Green Bay viewpoint. Using high-resolutions ensured detailed mesh characteristics, such as tributaries, inlets, and outlets, which were retained during the deformation process. Additionally, a 12-bit grayscale resolution proved sufficient to produce smooth geometry. ParaView was also used to output a second set of color-mapped images using the standard blue to red color map. These images were applied as a texture to the final deformed geometry in 3D Studio Max as well.

There are many advantages for using this approach. First, the amount of exaggeration can be easily changed by entering a different numeric *strength* parameter for the displacement modifier. Changes are instantaneous, and no additional input files need to be processed. By the traditional method, which is based on importing fixed-geometry files, one would need to reprocess extracted geometry files and re-render images again. A second major advantage of using this approach is that one may quickly preview an exaggeration amount over the whole dataset. This is possible because 3D Studio Max offers wire frame scrubbing through the data by dragging the time slider. Therefore, a particular exaggeration factor can be better evaluated via playback mode as opposed to viewing single frames. Both of these advantages allow this approach to be dynamic. The process of changing an exaggeration amount is accomplished in seconds – before any frames are rendered.

Another advantage is that multiple viewpoints are possible without importing different input files to the renderer. Different views require different amounts of exaggeration to be used. The closer the viewpoint, the lesser the amount of exaggeration is required. This was the case with our data. The first view showing Lake Michigan and Lake Huron required displacement modifier strength of 100 to be used. However, in the fourth viewpoint showing a perspective view of the Fox River, a displacement modifier strength of 1 was used. In both cases, the same files could be processed for the different views. Using a traditional approach based on fixed-geometry files, a different set of geometry files needs to be imported to the rendering software for each view.

Figure 5. Shown are water-level data from a very high-resolution, 8,000×7,460 pixel, grayscale image. Pure white shows maximum water-levels; pure black shows minimum levels. Very high-resolution images ensure details in inlets and estuaries, which are preserved.

5. Conclusion

Although the deformation modifier capability of 3D Studio Max has been used artistically by conceptual animators (Murdock, 2011), it has proved successful for accurately processing scientific data with many time-saving advantages where exaggerating a single-dimension is necessary. Once the displacement modifier is applied to some base geometry, instantaneous scrubbing through the entire dataset in preview mode allows any necessary refinement to be made to the exaggerated amount quickly and easily before any portion of the sequence is rendered. Additionally, rendering additional viewpoints that require varying the degree of exaggeration does not require separate geometry-capturing passes of the original simulation data. The same set of grayscale images can be used for as many different viewpoints as a user wants to produce; each viewpoint may have its own customized exaggeration by varying the numeric "strength" parameter.

References

Ceyhan, E., Basuchowdhuri, P., Judeh, T., Ou, S., Estrade, B., and Kosar, T., "Towards a faster and improved ADCIRC (Advanced Multi-Dimensional CIRCulation)." *Journal of Coastal Research SI50 (2007) (Proceedings ICS2007)*, http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.135.9397.

Kitware, Inc., *File Formats for VTK (Version 4.2)*, 2010, http://www.vtk.org/VTK/img/file-formats.pdf, March 2, 2012.

Murdock, K., *3ds Max 2012 Bible*, John Wiley & Sons, Indianapolis, IN, 1688 pgs., 2011.

Zhang, C., Dekate, C., Allen, G., Kelly, I., and Maclaren, J., "An Application Portal for Collaborative Coastal Modeling." *Concurrency Computat.: Pract*, 2007.

Accuracy and Efficiency Enhancements for High-Speed Aero-Propulsive Flows with Combustion

A.C. Zambon, K. Brinckman, G. Feldman,
D.C. Kenzakowski, and S.M. Dash
Combustion Research and Flow Technology, Inc., Pipersville, PA
{azambon, kbrinckman, gfeldma, kenzakow, dash}@craft-tech.com

K.D. Kennedy
US Army Missile Research Development and Engineering Center (AMRDEC), Redstone Arsenal, AL
kevin.kennedy4@us.army.mil

Abstract

Large-scale, high-fidelity computational fluid dynamics (CFD) simulations of missile-related aero-propulsive problems with complex gas-phase/multi-phase combustion can be very challenging, both in terms of accuracy requirements and computational cost. The primary technical objective of the second year of this Department of Defense (DoD) high performance computing Challenge program is to improve the predictive capabilities and computational efficiency of multi-phase reacting flow CFD analyses, encompassing particle and droplet combustion and supporting the design and evaluation of missile system/components of interest to the Army. Effective approaches that improve the accuracy of particle/droplet combustion are being investigated. Possible applications of interest include missile fuel venting events, and metallic nano-particle and liquid hydrocarbon fuel injection in scramjets. Specifically, the emphasis of this paper is on the high-fidelity modeling of liquid fuel/oxidizer injection, which entails strong coupling of a number of complex phenomena, including the primary break-up of the liquid jets into droplets, followed by secondary break-up, evaporation, and ultimately combustion in the gas-phase. A demonstration of the advanced modeling capabilities required for this sequence of events is presented for an open-literature fundamental problem. These core multi-phase combustion modeling capabilities are being unified in CRAFT Tech's CFD codes, namely CRAFT CFD® and CRUNCH CFD®. Given the persistent high computational cost of the solution of the gas-phase finite-rate chemistry, in particular in the presence of a disparity between chemical time-scales and fluid-dynamic time-scales, recent advances in graphics processing unit (GPU)-enabled chemistry acceleration for CFD are also presented. The significance of this program to DoD is the improvement in accuracy and efficiency of CFD codes that are in widespread usage by DoD to support missile design and evaluation.

1. Introduction

The high performance computing (HPC) Challenge Project C4V entails improvements in solution accuracy, as well as computational efficiency, for CFD codes that are in widespread usage by the Department of Defense (DoD). These codes, namely CRAFT CFD® and CRUNCH CFD®, support a broad variety of missile-related aero-propulsive problems featuring complex gas-phase/multi-phase laminar and turbulent combustion. Specifically, high-fidelity predictive capabilities of the computational fluid dynamics (CFD) codes employed are critical in many applications supporting scramjet and missile system/component design of interest to the Army, and are enabled by the utilization of advanced chemistry/multi-phase/turbulent combustion models in large-scale CFD simulations. However, these models can render computations significantly longer and more resource-intensive to the extent that CFD simulations become impractical. The overall objective of this program is to address this limitation by applying to this class of problems innovative approaches that can accelerate computations, so that they can be completed within a reasonable turnaround time and yet provide the desired predictive accuracy.

In the first year of this DoD HPC Challenge Project, the focus was on high-speed aero-propulsive *gas-phase* reacting flows[1]. For this class of applications, it is recognized that an advanced modeling framework capable of accurately capturing complex combustion phenomena, like ignition and extinction, requires the use of advanced finite-rate chemistry models. Unlike many standard reaction mechanisms currently in use by the plume and scramjet community, these

advanced chemical kinetic models generally entail a larger number of chemical species and elementary reactions. These models feature a number of short-lived radical species characterized by very small chemical time-scales that need to be resolved. Therefore, the computational cost associated with the deployment of these higher-fidelity models in large-scale CFD simulations can be substantial. The methodologies investigated in Year 1 are summarized in Figure 1.

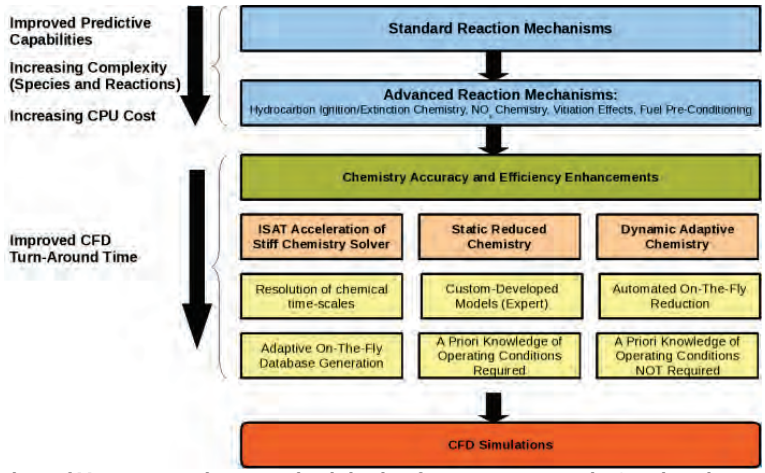


Figure 1. Schematic overview of Year 1 gas-phase Methodologies for Improvement is Combustion Predictive Capabilities and Computational Performance

The second year of this DoD HPC Challenge Project focuses on *multi-phase* combustion. While gas-phase combustion continues to play an important role in the overall modeling framework and the Year 1 methodologies are still being leveraged to reduce computational cost, the high-fidelity modeling of multi-phase combustion requires the inclusion of a number of additional strongly-coupled modeling components in order to capture the complexity of the physical and chemical phenomena involved. Depending on the nature of the heterogeneous phase fuel under consideration, kinetically- or diffusion-controlled particle burning or liquid jet break-up into droplets, followed by secondary break-up, vaporization, and ultimately gas-phase combustion are strongly-coupled aspects of the problem to be accounted for. Part of this HPC Challenge Project is devoted to exploring ways of enhancing the fidelity of multi-phase CFD simulations. Typical multi-phase applications of interest to the Army include missile fuel venting events, scramjet alternative fuels based on nano-sized metallic particle addition, as well as liquid hydrocarbon fuels, as an alternative to hydrogen fuel to reduce fuel volumetric requirements. Due to space limitations, the primary focus of this paper is centered on improving the accuracy of liquid fuel venting/injection with the application of a recently developed end-to-end modeling framework for liquid fuel combustion to a fundamental open-literature combustion problem. The relevance of this higher-fidelity modeling framework to aero-propulsive applications is discussed next, followed by an overview of the relevant multi-phase model components and a discussion of the simulation results. While gas-phase combustion was deliberately simplified in this demonstration, the predictive capabilities of the framework still rely on the accuracy of the chemical kinetics models. To complement the Year 1 approaches, recent advances in graphics processing unit (GPU) chemistry acceleration are also presented.

2. Relevance of High-Fidelity Liquid Fuel Injection/Venting Modeling

The venting of a gas from a missile at lower-altitudes can occur when pressure relief values are used to reduce propellant tank pressures due to cryogenic vaporization or as part of a controlled engine shutdown procedure during the missile-staging event. The vented gas may burn due to interactions with the missile aerodynamic flow, and may also enhance the afterburning of the plume. A similar scenario may arise, for instance, when the missile body is punctured, e.g., by a high-energy source. The ability of plume CFD codes to properly analyze these problems largely resides in the modeling of the chemical kinetics and of the liquid fuel jet phenomenology. CRAFT has analyzed this class of applications in the past[2,3]. As an illustration, a study was conducted on a generic missile body flying at 10 km, where the venting of gaseous methane was assumed to arise from a 6-inch diameter vent port in the center of the missile (refer to Figure 2)[2]. The methane fuel was assumed to be stored cold (300K), and at a pressure of 2 atm. However, to simplify the modeling requirements, the methane fuel was assumed to be vented already in the gas-phase and be readily oxidized in the

surrounding environment. As shown in Figure 3, burning was evident as the vented fuel mixed with the hot plume, and, there was a clear increase in temperature where the mixing occurred, along with the presence of product chemical species.

Figure 2. Missile liquid jet venting problem set-up

A concurrent study investigated the interaction of the liquid fuel jet vented into the gas environment[2]. Computationally, this type of problem involves a different set of challenges compared to gas-gas problem illustrated above, due to the large variation in fluid density and compressibility between the two phases. Figure 4 presents the liquid volume and fuel mass fraction computed from this earlier study. Furthermore, in a high-shear environment, the liquid stream can break-up into droplets, and based upon local thermodynamic conditions, could vaporize into a gas-phase. Thus, a gas-gas problem could ensue, similarly to the scenario above. In this case, however, the characteristics of the jet/free-stream interaction, such as jet penetration and gas mixing, could be significantly different, depending on the liquid jet break-up.

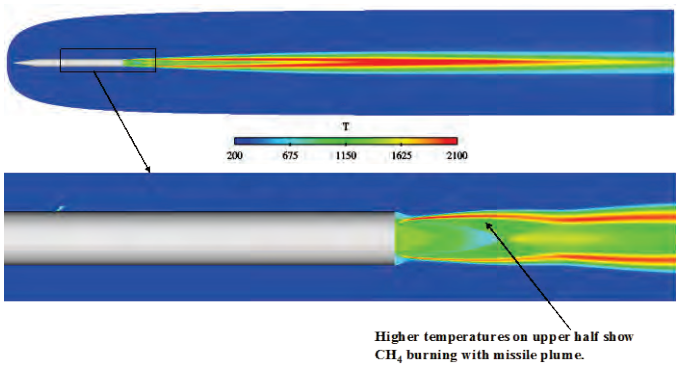


Figure 3. Temperature contours show vented fuel burning with hot plume (from 2)

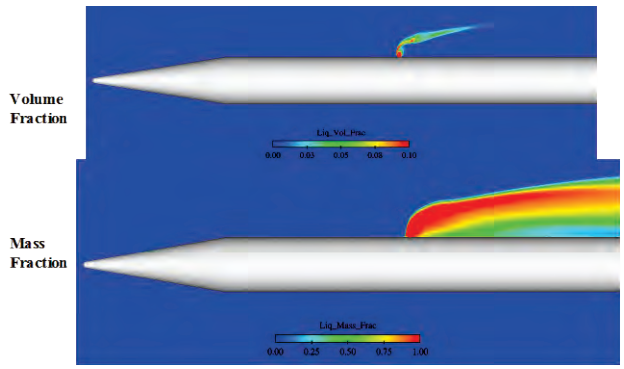


Figure 4. Liquid volume and mass fraction contours for a missile at 10 km (from 3)

A higher-fidelity modeling of the fuel venting problem could be attained by coupling all the relevant multi-phase models under a common framework. As anticipated, part of this HPC Challenge Project is devoted to improving these predictive capabilities by developing an end-to-end modeling framework capturing the entire sequence of events going from liquid fuel injection to gas-phase combustion. Due to the significant challenges of this effort, CRAFT Tech has focused on a fundamental problem, which contains all the elements of the ultimate missile/scramjet application in a more controlled setup amenable for validation, and implemented a methodology from the open published literature to model impingement break-up (IBU) of doublet injectors. This newly-developed impingement break-up methodology works within a Reynolds-averaged Navier-Stokes (RANS) framework to transfer material from a liquid sheet to dispersed-phase Eulerian droplets at the point where the sheet is predicted to break-up based on empirical correlations. In a combustor application, the fuel/oxidizer droplets formed by impingement atomization can undergo secondary break-up to smaller sizes while vaporizing into gaseous fuel and combusting. This sequence is shown schematically in Figure 5. CRAFT Tech has utilized the various individual models of the atomization/combustion sequence in previous research efforts, as in the case of the missile fuel venting problem, but had not coupled all of the models for application to an end-to-end injector/combustion simulation. CRAFT Tech has recently extended the IBU methodology to include droplet secondary-break-up, droplet vaporization, and combustion, providing an end-to-end fuel atomization/combustion framework for cryogenic doublet injector applications. The completion of this challenging demonstration represents a major milestone in the enhancement of missile/scramjet modeling.

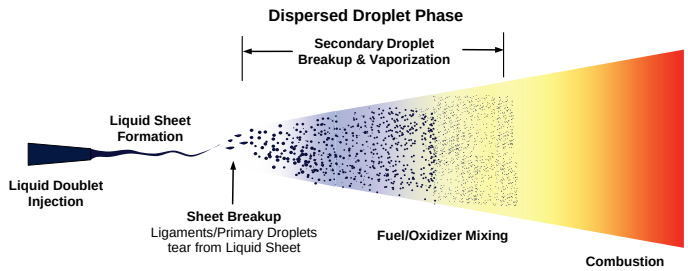


Figure 5. Schematic of the end-to-end injector/combustion sequence

3. Computational Framework

The foundation of CRAFT Tech's high-fidelity computational framework is represented by two advanced CFD flow solvers, namely CRAFT CFD®, a higher-order O(3-5) code that utilizes a structured grid methodology[4] and CRUNCH CFD®, an unstructured flow solver that is very computationally efficient for complex geometries[5]. Extensive generalized chemical kinetics and particle/droplet modeling capabilities are available in the two flow solvers, with both codes sharing a high-degree of commonality. Both codes solve the RANS equations for thermally-perfect gases, as well as for real fluids. Key gas-phase combustion modeling capabilities, including: i) targeted chemical reaction mechanism reduction, ii) tabulated chemistry approaches, and iii) innovative on-the-fly mechanism reduction methodologies, were presented in 1. Here, emphasis is given to those multi-phase modeling components that are an integral part of the end-to-end modeling framework. These include: i) gas/liquid continuum phase modeling, ii) primary break-up modeling, iii) dispersed-phase transport modeling, iv) droplet secondary-break-up, v) droplet vaporization, and iv) gas-phase combustion, as discussed next.

3.1 Gas/Liquid Continuum Phase Modeling

Various multi-phase problems involved in aero-propulsive applications require the prediction of a gas/liquid interface, which presents a challenge because the density of the two phases can vary by several orders-of-magnitude, causing numerical stiffness issues. Traditional methods for interface capturing employ either volume-of-fluid (VOF) and/ or level-set (LS) methods in which the complete Navier-Stokes equations are solved for both the liquid jet and gas flow, along with a surface-interface equation. Once the surface is captured, VOF and LS methods may seek to predict the surface wave structure and material stripping process leading to droplet formation. While there has been some success in applying surface capturing methods to resolve the surface waves and predict break-up on a research level, their application in engineering simulations is impractical, in that time-accurate and highly-resolved calculation of the interface evolution is required to predict and track the surface wave instability. An ensuing break-up of the liquid surface can yield a large number of arbitrarily shaped ligaments and droplets, whose size is well below the resolution of a practical computational mesh. Thus, the ability to resolve the liquid jet surface break-up and resulting droplets, using a surface capturing method, as part of a larger-scale problem, such as combustor liquid fuel injection or missile fuel venting, becomes very computationally-expensive. CRAFT Tech's approach to engineering simulations of liquid surface break-up and atomization in complex environments is accomplished through the use of a robust gas/liquid interface capturing methodology[6], where the interface cells contain a homogenous gas/liquid mixture with generalized fluid thermodynamics (i.e., pure liquid, pure gas or mixtures in between) coupled to an established surface break-up and atomization correlation using local gas/liquid properties. This method avoids the requirement of capturing a sharply-defined interface within a given computational cell, in order to retain the detailed structure of the surface waves, thus making large-scale problems computationally-practical.

In CRAFT Tech's gas/liquid formulation, the independent variables are transformed from a density and internal energy system to a pressure and enthalpy system. Mixtures of "generalized fluids" follow Amagat's law, and not Dalton's law. The original system of equations is given by:

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}+\frac{\partial G}{\partial z}=S+D_v \tag{1}$$

where

$$Q = [\rho, \rho u, \rho v, \rho w, \rho Y_G, e, \rho k, \rho \varepsilon]^T \tag{2}$$

Here, ρ represents the mixture density which is related to the liquid (L) and gas-phase (G) densities by their respective mass fraction (Y) in a given computational cell by:

$$\rho = \frac{1}{Y_L / \rho_L + Y_G / \rho_G} \tag{3}$$

The transformed system takes the form:

$$\Gamma \frac{\partial Q_v}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = S + D_v \tag{4}$$

where

$$Q_v = [u, v, w, Y_G, T, k, \varepsilon]^T \tag{5}$$

and Γ is a transformation matrix involving thermodynamic derivatives utilizing user-defined equations of state. For efficient operation over a wide-range of Mach numbers, this matrix is preconditioned to get better-matched eigenvalues that improve convergence and reduce round-off errors. For a comprehensive discussion on the gas/liquid formulation, solution procedure, and numeric, the reader is referred to Reference 8. Previous works[6–8] have shown that the upgraded formulation provides a robust methodology for capturing the gas/liquid interface for various applications, such as cavitation, liquid venting, and bulk liquid fly-out problems without limitations on system pressure or required liquid compressibility, and comparisons to data have been good.

3.2 Primary Break-up Modeling

Currently, the most challenging part of simulating a liquid discharge to combustion sequence is modeling the primary break-up of the liquid-phase into dispersed-phase droplets. The primary break-up mechanism involves the deformation of the gas-liquid interface and the formation of surface waves. The stripping of bulk liquid can be resolved by computing the merging of iso-surfaces of liquid mass fraction in a VOF type calculation, which may be an alternative in describing break-up in the Rayleigh regime, where the drops are of the order or greater than the original characteristic liquid length (e.g., jet diameter, sheet thickness). However, in more general situations, for atomizing flows, the by-products of primary break-up will be much smaller than the bulk, and the computational grids required to resolve the surface waves will be prohibitively large. To circumvent this difficulty, models are used to compute the rate of formation of a spray of droplets and characterize their size.

As an alternative to resolving the surface break-up numerically, CRAFT Tech has investigated the use of surface break-up models, in which correlations for droplet formation are applied based on local gas/liquid conditions, e.g., viscosity, density, and surface tension, at the gas/liquid interface. Correlations for aerodynamic break-up of isolated liquid jets, such as those involved in missile fuel vent problems, have been implemented in our CFD codes, as presented in 9, and most recently for impingement break-up (IBU) simulations, where multiple liquid jets impinge to form a liquid sheet which goes unstable and disintegrates into droplets, as shown in Figure 6.

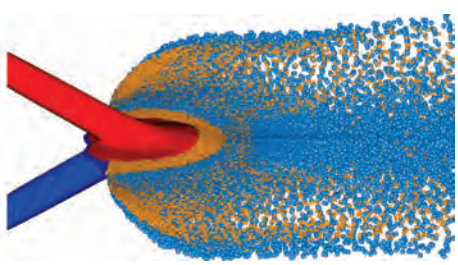


Figure 6. Jet impingement break-up

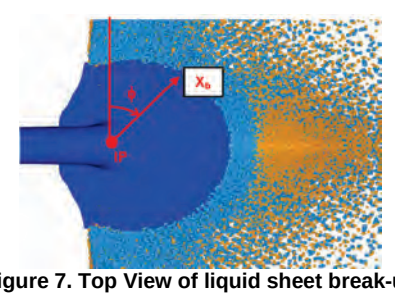


Figure 7. Top View of liquid sheet break-up

In the IBU methodology in CRUNCH CFD® , the liquid sheet resulting from impinging liquid injectors is predicted and a break-up length X_b is calculated for particular injector/combustor conditions to determine how far from the impingement point (designated as IP in Figure 7) the sheet will tear into ligaments and droplets. The break-up length is

prescribed based on the jet Weber number, and gas and liquid densities using an empirical correlation for doublet injectors, based on the work of Strakey and Talley[10], and is given by the relation

$$x_b/d_o = 6.47\left[We_j\left(\frac{\rho_g}{\rho_l}\right)^2\right]^{-0.192} \tag{6}$$

where We_j is the jet Weber number based on the liquid properties, and jet velocity and diameter, d_o is the jet nozzle diameter, and ρ represents the density of the gas (g)- and liquid (l)-phases.

The sizes of the droplets generated vary spatially as well as in time, and are a function of the local evolution of the gas/liquid interface and jet conditions. At the calculated break-up point, the mass, momentum, and enthalpy of the liquid sheet is transferred to a Eulerian droplet dispersed-phase, whose droplet diameter d_D is calculated from the LISA correlations for sheet break-up[11]. The LISA model provides analytically derived expressions for droplet diameter, which are applicable to the size at the sheet-edge where break-up occurs. Since the droplet pattern is extremely dense in atomizers around the break-up location, it is impossible to get accurate measurements of mean diameter with state-of-the-art experimental techniques; therefore the droplet diameter data used in correlations is typically taken many jet diameters downstream, and may not represent the true diameter at the break-up location required for CFD simulations. Therefore, CRAFT Tech has chosen to implement the LISA analytical expressions for droplet size in CRUNCH CFD®.

A sink term is imposed on the liquid-phase corresponding to, but opposite in sign to, the particle source term W_{p4}, where the subscript L denotes a property of the liquid-phase, $\dot{m}$ is the rate of mass of leaving the liquid-phase, h_L is the liquid enthalpy, and u, v, and w are the gas/liquid mixture velocities in the x, y, and z directions, respectively.

$$W_{p4} = \frac{1}{V_{cell}}\left[\dot{m}, \dot{m}u, \dot{m}v, \dot{m}w, \dot{m}h_L,\ 4\dot{m}/\rho_L d_{32}\right]^T \tag{7}$$

The sixth term in W_{p4} is related to the surface area per unit volume (S_p) which is used to calculate the Sauter-Mean Diameter (d_{32}) of the droplet group as discussed in the next section.

3.3 Dispersed Phase Transport Modeling

The droplet transport model employs a continuum type of approach to modeling a large number of droplets, by tracking a mass fraction of droplets and average particle surface area (Sp) in a given control volume. This along with the droplet density can be used to calculate the Sauter-Mean Diameter (SMD), i.e., the diameter of a droplet whose volume (Vp) to surface area ratio is the same as that for the entire continuum cloud, as

$$d_{32} = 6\frac{V_p}{S_p} \tag{8}$$

which experiments have shown is an appropriate particle-size metric to base vaporization and/or combustion rate calculations, where the droplet size changes throughout the domain. This approach avoids tracking a prohibitively large number of droplets, or using binning to cover a range of particle radii.

The liquid droplets are tracked within the computational domain using a Eulerian particle approach with second-order upwind flux numerics. Conservation equations for the Eulerian continuum cloud are solved for the mass, momentum, enthalpy, and droplet surface area within a computational cell. The Eulerian continuum-cloud particulate equations are written in generalized coordinates as follows:

$$\frac{\partial Q_p}{\partial t} + \frac{\partial E_p}{\partial \xi} + \frac{\partial F_p}{\partial \eta} + \frac{\partial G_p}{\partial \zeta} = W_{p1} + W_{p2} + W_{p3} + W_{p4} \tag{9}$$

where:

$$Q_p = \begin{bmatrix} \rho_p \\ \rho_p u_p \\ \rho_p v_p \\ \rho_p w_p \\ \rho_p h_p \\ S_p \end{bmatrix}; \qquad E_p = \begin{bmatrix} \rho_p \\ \rho_p u_p U_p \\ \rho_p v_p U_p \\ \rho_p w_p U_p \\ \rho_p h_p U_p \\ S_p U_p \end{bmatrix} \tag{10}$$

In this set of equations, ρ_p represents the particle cloud density, which can be expressed as follows:

$$\rho_p = N m_p \tag{11}$$

where N is the local number density of the droplets, and m_p is the averaged mass of the droplets. Assuming that the droplets are spherical in shape,

$$m_p = \frac{\pi}{6} d_p^3 \rho_s \tag{12}$$

where d_p is taken as the Sauter-Mean Diameter and ρ_s is the droplet material density.

The remaining variables in the above equation denote the particle velocity (u_p,v_p,w_p), enthalpy (h_p), surface area per unit volume (S_p) and the cell contravariant velocity, U_p. The fluxes F_p and G_p are defined similarly for the other computational directions. The source vector W_{p1} includes inter-phase drag and heat-transfer terms, W_{p2} denotes source terms related to the condensation and evaporation of the droplets with the surrounding gas-phase (discussed below), W_{p3} contains the gravitational-source term, and W_{p4} is related to primary break-up of the liquid-phase as described in the previous section.

3.4 Droplet Secondary Break-Up

The methodology to analyze how larger droplets (such as those produced in the primary break-up stage) break-up into smaller droplets then vaporize and burn, follows directly from that used in modeling spray combustion problems.
Secondary break-up, at a basic level, is directly related to the Weber number, and can be classified in terms of three basic mechanisms[12], namely:

$$We = 2\rho_g r \left| U_g - U_p \right|^2 / \sigma \tag{13}$$

Our most computationally-efficient droplet break-up/vaporization methodology makes use of a Eulerian framework, which tracks the droplet SMD in each computational cell. The secondary break-up methodology, applied at an engineering level, determines how many child-droplets a single parent-droplet breaks into, based on the droplet Weber number. A simplified secondary break-up model, based on the characteristics of the regimes in Table 1, has been incorporated into our CFD codes.

Table 1. Droplet break-up mechanisms

Weber Number	Break-up Mechanism	Description
6<We≤80	Bag Break-up	Thin bag forms behind droplet rim
80<We≤350	Stripping Break-up	Shear forces strip droplets from liquid ligaments
We>350	Catastrophic Break-up	Droplet immediately disintegrates

3.5 Droplet Vaporization

The secondary break-up methodology is coupled to a droplet vaporization model based on a modified form of the Hertz-Knudsen equation, which gives a mass transfer rate as the difference between incoming fluxes from the gas-phase, and evaporative fluxes from the droplet of radius r.

$$\dot{m}_p = N \quad 4\pi r^2 \left[\frac{P_{partial}}{\sqrt{2\pi RT}} - \frac{P_{sat_{\text{eff}}}}{\sqrt{2\pi RT_d}} \right] \tag{14}$$

$P_{partial}$ is the partial pressure of the droplet substance vapor in the gas-phase, T is the gas-phase temperature, T_d is the droplet temperature, and R is the gas constant for the material changing phase. An effective saturation pressure, P_{sateff}, given by the Kelvin-Helmholtz equation, which accounts for surface tension, σ, is used.

$$P_{sat_{\text{eff}}} = P_{sat} \exp\left[\frac{2\sigma}{\rho_d R T_d r} \right] \tag{15}$$

Here P_{sat} is the flat film saturation pressure at the droplet temperature. Surface tension is also expressed as a function of droplet temperature.

The particle source term W_{p2} is constructed to transfer mass between the gas and droplet phases. For the disperse phase, the source term becomes

$$W_{p2} = \frac{1}{V_{cell}} \begin{Bmatrix} \dot{m}_p \\ \dot{m}_p u_p \\ \dot{m}_p v_p \\ \dot{m}_p w_p \\ \dot{m}_p h^* \\ \dot{m}_p S_p / \rho_p \end{Bmatrix} \tag{16}$$

The value used for h^* in the above equation is dependent on whether condensation or evaporation is occurring within the given cell. For condensation (i.e., when $\dot{m}_p > 0$), the enthalpy of the condensable gas species within the mixture is utilized; for evaporation, the enthalpy of the droplet plus its heat of vaporization is used. A similar source term expression is used in the gas-phase equations but with opposite sign.

3.6 Gas-Phase Combustion Modeling

Current modeling capabilities for reacting systems encompass (i) detailed, (ii) skeletal, (iii) reduced, and (iv) global/semi-global kinetic models for a variety of fuels relevant to aero-propulsive applications. As detailed in 1, a number of methodologies can be applied to capture, reasonably well and efficiently, the combustion physics. In the demonstration presented in this paper, a representative global chemistry model is deployed. Global chemistry entails the use of global ad hoc finite-rate reaction steps, generally irreversible, to model the conversion of fuel and oxidizer into the final combustion products. One or a few reaction steps may be selected to represent the oxidation process. A key feature of global chemistry is that the rate parameters are not systematically derived from the rates of elementary reactions of detailed or reduced chemistry models. As a matter of fact, global chemistry rates are ad hoc and have to be derived for each application, combustion problem or narrow-range of conditions. For this demonstration, CRAFT Tech has developed a simplified CH_4/O_2 global chemistry model for computational efficiency, which has allowed for testing of the end-to-end framework with combustion. The single irreversible step for CH_4/O_2 combustion was modeled as

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O \tag{17}$$

with the global rate expressed in generalized Arrhenius form.

4. Results and Discussion

4.1 End-to-End Unlike Doublet Injector Simulation - Non-Reacting

A hypothetical LOX/CH_4 doublet single-injector configuration was analyzed to verify the functionality of the CRUNCH CFD® IBU/dispersed-phase/combustion CFD framework, and demonstrate a capability to model the complete sequence from liquid cryogenic fuel/oxidizer injection with impingement break-up through combustion. Because the fuel and oxidizer are different liquids with different material properties (density, viscosity, etc.) at different conditions (temperature, velocity, pressure), their respective sheets can break-up at different locations. The unlike doublet configuration is shown in Figure 8. The computational domain takes advantage of half-symmetry (one-half the width of the domain shown in Figure 8). The impingement half-angle is θ=30° and the jet boundary conditions listed in Table 2 provide momentum balance of the jets at the impingement point. The surrounding chamber is filled with nitrogen at 3.0 MPa. The initial domain temperature is set at 300K to facilitate droplet vaporization, and walls are added that surround the doublet to mimic a one-injector combustion chamber. The right-most boundary is a flow outlet.

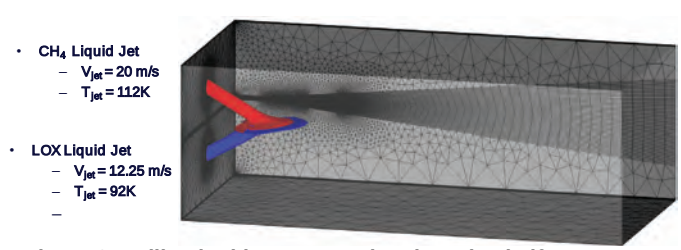


Figure 8. Unlike-doublet computational mesh – half-symmetry

Table 2. Unlike doublet jet boundary conditions

	Fuel Jet	Oxidizer Jet
Material	CH_4	LOX
Diameter (mm)	1.194	1.194
Velocity (m/s)	20.0	12.25
Temperature (K)	112	92
Pressure (MPa)	3.0	3.0

As a first step, a non-reacting configuration was analyzed with droplet formation from IBU, secondary break-up, and droplet vaporization, to verify that the hypothetical configuration would result in droplet vaporization and generate a fuel/oxidizer mixture to support combustion. For this effort, the chamber was initially filled with N_2 gas at a temperature of 300K and a pressure of 3.0 MPa (representative of combustion pressures). Despite much lower ambient temperatures than would be encountered in a combustion simulation, both the CH_4 and LOX droplets generate a significant amount of vapor as shown in Figure 9 and Figure 10. The CH_4 droplet diameter ranges from 1.0846E-04m to 5.07e-5m, and the LOX droplet diameter ranges from 1.4422E-04m to 3.249e-5m, with the smaller droplets at the outer periphery where the gas temperature is highest. The vapor clouds of the two species appeared to be sufficiently mixed to support ignition if the combustion model would be activated.

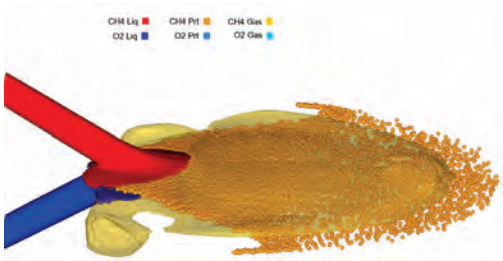


Figure 9. Non-reacting unlike doublet - CH_4 droplets and vapor

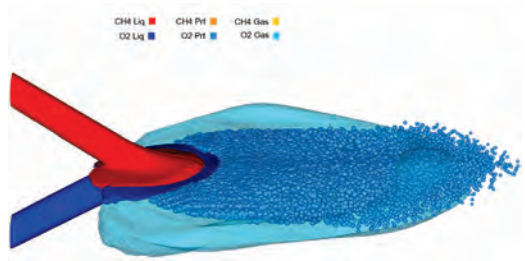
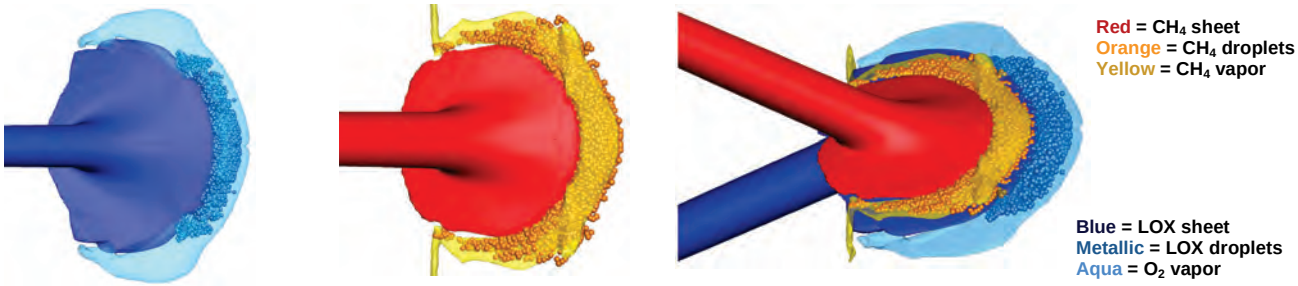


Figure 10. Non-reacting unlike doublet - O_2 droplets and vapor

4.2 Reacting End-to-End Unlike Doublet Injector Simulation

The CH_4/LOX unlike doublet was now simulated including combustion of the fuel/oxidizer species generated by droplet vaporization. The reacting doublet problem was not started from the non-reacting solution, but was started over from liquid-jet injection to assess how the problem would evolve with all models activated. In the following discussion, results are shown as the calculation proceeds; however, while it may have been interesting to see the solution evolving in time, to conserve resources the simulation was not run with sub-iterations to produce time-accurate results. Therefore, results here do not precisely correspond to snapshots in real-time, only stages of the calculation; however, it is expected that a time-accurate calculation would show similar behavior.

With a 300K free-stream gas temperature, the droplets vaporized readily and formed a combustible mixture of CH_4/O_2 gas. Figure 11 shows the sheet, droplets, and generated vapor at an early stage in the calculation. The Eulerian droplets are represented using the Largrangian droplet post-processor and provide an approximation of the droplet distribution. At this stage in the calculation, the CH_4 vapor and O_2 vapor are just beginning to mix and combustion is near onset. A 1,200K ignition source is applied around the sheet location to initiate burning when sufficient fuel/oxidizer mixing occurs.



(a) View from above LOX sheet; light blue surface represents O_2 vapor

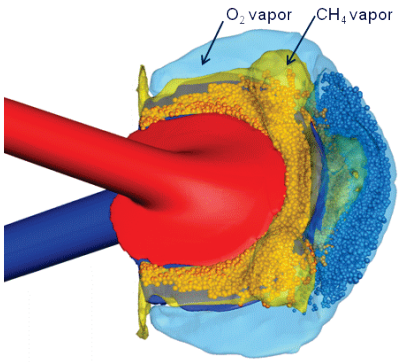
(b) View from below CH_4 sheet; yellow surface represents CH_4 vapor

c) Combined CH_4/O_2

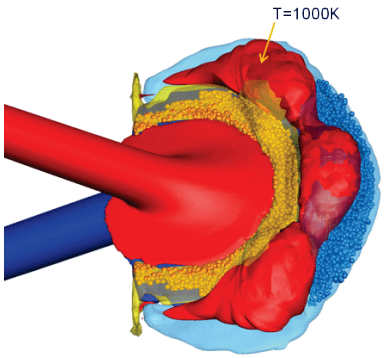
Figure 11. Unlike doublet - droplet/vapor formation early in the calculation

As the doublet calculation proceeds, the droplets continue to vaporize, and the resultant CH_4/O_2 gas mixes sufficiently to burn, with the ignition source initiating the combustion. Figure 12a shows the liquid sheets with CH_4/LOX droplets generated from IBU, and CH_4/O_2 gas from droplet vaporization. Figure 12b shows the same view with an iso-surface of gas temperature at 1,000K, revealing that combustion is taking place very near the sheet-edge, as the droplets have vaporized, and their respective gas-phases have mixed sufficiently to burn with the ignition source present. The color legend in Figure 11 applies. Figure 13 shows a similar view at a later time in the calculation, as an appreciable amount of fuel and oxidizer gas is present, with significant burning taking place as indicated by the iso-surface of gas temperature at 2,200K. The effect of the higher gas temperature on droplet vaporization is apparent in the reduction in droplet size away from the sheet-edge. The CH_4 droplet diameter ranges from 1.0846E-04m to 2.7E-5m, and the LOX droplet diameter ranges from 1.4422E-04m to 1.41E-5m.

Because this is a hypothetical doublet configuration, it is uncertain that combustion can be sustained with this simplified fuel/oxidizer doublet, even though all components of the end-to-end framework are operating properly in a coupled manner. However, the results shown demonstrate the functionality of the end-to-end IBU/Combustion framework on a 3D unlike doublet in modeling the sequence from liquid cryogenic fuel/oxidizer injection with impingement break-up through combustion, paving the way for future efforts which require modeling of the sequence from liquid jet break-up through combustion.

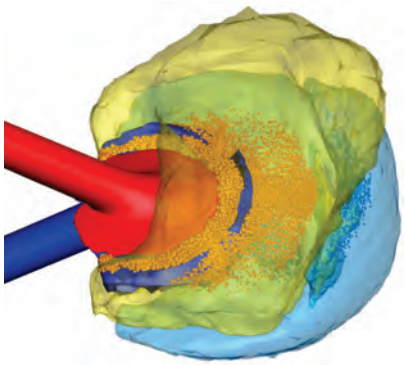


(a) Liquid sheets with CH_4/O_2 droplets & vapor

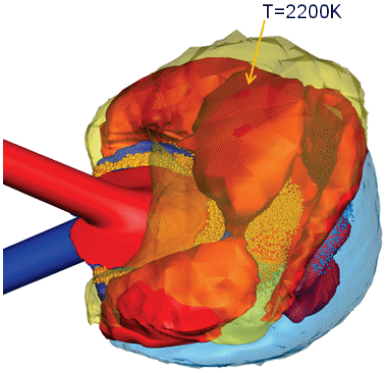


(b) Liquid sheets with CH_4/O_2droplets & vapor, & iso-surface of Temperature=1000K

Figure 12. Unlike doublet - droplet/vapor & temperature early after ignition (see Figure 11 for legend)



(a) Liquid sheets with CH_4/O_2 droplets & vapor



(b) Liquid sheets with CH_4/O_2droplets & vapor, & iso-surface of Temperature=2200K

Figure 13. Unlike doublet - droplet/vapor & temperature (see Figure 11 for legend)

4.3Gas-Phase Chemistry Acceleration with GPU

Since the flow field computed in this end-to-end framework demonstration is inherently unsteady, the gas-phase chemistry was intentionally simplified. As higher-fidelity models are deployed, the computational cost is expected to rise significantly. In addition to the Year I chemistry acceleration methodologies, recently CRAT Tech has leveraged on hardware acceleration of GPUs to speed-up chemistry evaluations. A few highlights from this work are presented next. Specifically, CRAFT Tech has ported to the GPU an *entire* implicit chemistry solution algorithm by rewriting in CUDA

FORTRAN a stand-alone baseline implicit chemistry solver package previously tested on a CPU. The originality of this approach is that all operations including reaction-rate evaluation, matrix construction and inversion are performed on the GPU by generating customized self-contained GPU-enabled source code. A comparison of computational performance between GPU and CPU solving the exact same chemistry time-integration problem featuring 10,000 grid-points to be processed is presented in Table 3. It can be observed that as the complexity of the reaction mechanism increases, both in terms of number of species and number of reactions, the computational speed-up attained with the GPU increases. For instance, a speed-up factor of approximately 2 with a one-step model is very modest compared to a speed-up factor of approximately 28 achieved on a relatively large reaction mechanism such as GRI Mech 3.0. The primary reason for this trend is that the computational load on the GPU increases with the size of the system, and with a large computational load, the GPU can offset much faster the overhead cost associated with GPU initialization and data transfer. All the benchmark comparisons were performed on a Linux workstation featuring Intel Xeon E5606 CPUs (2128 MHz, 12GB, 8 cores) and an NVIDIA GTX 570 GPU (1572 MHz, 1.4GB, 480 CUDA cores). Some difficulties were encountered with the compilation of the GRI Mech 3.0 implementation due to very long compilation times associated with the optimization of a very large kernel source code (of the order of tens-of-thousands of lines). The CPU and GPU codes for GRI Mech 3.0 were run successfully on a high-end HPC system featuring AMD Opteron 6,134 CPUs (2300 MHz, 32GB, 16 cores) and an NVIDIA M2050 (1572 MHz, 3.0GB, 448 CUDA cores with ECC). For GRI Mech 3.0, Table 3 also shows how dramatic the GPU speed-up is as the number of points processed increases. While the computational performance is comparable for about only 100 points, it increases to 28 for 10,000 points. A key finding from this benchmark study is that GPU performance increases as opportunities per massive data-parallelism also increase, i.e., the number of grid-points, which is the case for large-scale CFD simulations.

Table 3. GPU vs. CPU chemical kinetic timing comparisons for various reaction mechanisms

Reaction Mechanism Label	Reaction Mechanism Type	Number of Species	Number of Reactions	GPU Time [s]	CPU Time [s]	Speed-Up Factor	Number of Grid Points
ONESTEP	Global	5	1	4.9	9.8	~2	10,000
OHN	Detailed	9	20	6.9	73.7	~10.6	10,000
JET-A	Semi-Global	20	29	33.6	497.9	~14.8	10,000
C2H4	Skeletal	31	128	61.7	1,212.2	~20	10,000
GRI30	Detailed	53	325	54.5	11.3	~0.20	10
				75.8	111.6	~1.47	100
				86.0	1,114.9	~13	1,000
				386.5	11,042.3	~28.57	10,000

5. Conclusion

In this paper, a high-fidelity end-to-end framework for modeling liquid fuel venting/injection was presented and demonstrated for a fundamental open-literature impinging jet problem. By accounting for the sequence of events occurring from the moment the liquid fuel jet is introduced and combustion actually takes place in the gas-phase, a higher-level of simulation accuracy can be attained. Specifically, strongly-coupled physical phenomena, including primary and secondary break-up and droplet evaporation, are now being modeled. These enhanced capabilities will support high-speed aero-propulsive applications like missile-fuel venting and scramjet liquid fuel injection simulations. To complement Year 1 technical activities, recent advances in GPU-enabled chemistry acceleration algorithms were shown to have a significant impact in speeding-up chemistry evaluations targeted for CFD applications.

Acknowledgements

The support of the Army and all other agencies that over the years have contributed to advancing CRAFT Tech's code modeling capabilities is gratefully acknowledged. Computing time at the NAVY HPC facility was provided through a DoD HPC Challenge Project.

References

1. Zambon, A.C., Ott, J.D., Dash, S.M., and Kennedy, K.D., "Accuracy and Efficiency Enhancements for High-Speed Aero-Propulsive Flows with Combustion." DoD HPC User's Group Meeting, Portland, OR, June 23, 2011.

2. Chenoweth, J.D., Brinckman, K.W., York, B.J., Feldman, G., Calhoon Jr., W.H., and Dash, S.M., "Progress in Modeling Missile Fuel Venting and Plume Contrail Formation." *Paper No. AIAA-2007-1012*, 45th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 8–11, 2007.

3. Feldman, G., Hosangadi, A., Brinckman, K.W., and Dash, S.M., "Application of Improved Gas/Liquid Methodology to High Speed Venting and Bulk Flyout Problems." *Paper No. AIAA-2007-0922*, 45th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 8–11, 2007.

4. Sinha, N., Hosangadi, A., and Dash, S.M., "The CRAFT NS Code and Preliminary Applications to Steady/Unsteady Reacting, Multi-Phase Jet/Plume Flowfield Problems." *CPIA Pub. 568*, 19th JANNAF Exhaust Plume Technology Meeting, pp. 203–226, May 1991.

5. Hosangadi, A., Lee, R.A., York, B.J., Sinha, N. and Dash, S.M., "Upwind Unstructured Scheme for Three-Dimensional Combusting Flows." *Journal of Propulsion and Power*, Vol. 12, No. 3, pp. 494–503, May–June 1996.

6. Hosangadi, A., and Ahuja, V., "A New Unsteady Model for Dense, Cloud Cavitation in Cryogenic Fluids." *Paper No. AIAA-2005-5347*, 17th Computational Fluids Dynamic Conference, Toronto, Ontario, CA, Jun. 6–9, 2005.

7. Ahuja, V., Hosangadi, A., and Arunajatesan, S., "Simulations of Cavitating Flows Using Hybrid Unstructured Meshes." *Journal of Fluids Engineering*, May/June, 2001.

8. Hosangadi, A. and Ahuja, V., "Numerical Study of Cavitation In Cryogenic Fluids." *Journal of Fluids Engineering*, Vol. 127, pp. 267–281, March 2005.

9. Brinckman, K., Hosangadi, A., Ahuja, V., Dash, S.M. and Feldman, G., "A CFD Methodology for Liquid Jet Break-up and Vaporization Predictions in Compressible Flows." *Paper No. AIAA-2008-1023*, 46th Aerospace Sciences Meeting and Exhibit, Reno, NV, 7-10 Jan 2008.

10. Strakey, P.A. and Talley D.G., "Spray Characteristics of Impinging Jet Injectors at High Back-Pressure." Eight International Conference on Liquid Atomization and Spray Systems, Pasadena, CA, July 2000.

11. Senecal, P.K., Schmidt, D.P., Nouar, I., Rutland, C.J., and Reitz, R.D., "Modeling High Speed Viscous Liquid Sheet Atomization." *International Journal of Multiphase Flow*, Vol. 25 Issues 6-7, pp 1073–1097, September 1999.

12. Tanner, F.X., "Development and Validation of a Cascade Atomization and Drop Break-up Model for High-Velocity Dense Sprays." *Atomization & Sprays*, Vol. 14, Issue 3, 2004.

Computational Investigation and Fundamental Understanding of a Slowed UH-60A Rotor at High Advance Ratios

Mark Potsdam
US Army Aeroflightdynamics Directorate Research, Development, and Engineering Command (AMRDEC), Moffett Field, CA
mark.potsdam@us.army.mil

Anubhav Datta and Buvana Jayaraman
Science and Technology Corporation, Moffett Field, CA
datta@merlin.arc.nasa.gov,
buvana.jayaraman@us.army.mil

Abstract

This work carries out coupled Computational Fluid Dynamics and Comprehensive Analysis (Helios/RCAS) of a full-scale UH-60A rotor slowed down to 40% of nominal RPM, operating at advance ratios up to 1.0. The objectives of the work are to 1) validate the analysis with detailed aerodynamic measurements under loading conditions unique to this regime, and 2) to use the analysis to complement test measurements and expand fundamental understanding of slowed rotor and high advance ratio physics. Airload validation shows excellent agreement with normal force, but pitching moments are poorly predicted on both the advancing (negative lifting) and retreating (reversed-flow) blades. Fundamental understanding is gained of the physics of the reversed-flow region, wake interactions, advancing and retreating blade moment impulses, differential span loading, and blade deformations—as influenced by RPM and advance ratio. Flow field visualizations show in exceptional detail the unconventional wake patterns, the reversed flow dynamic stall vortex progression, blade-on-blade interactions, reversed-flow-edge vortex, and root vortices. Comparisons of rotor performance and trim controls are documented, showing reasonable trend agreement.

1. Introduction and Significance

Slowed rotors have long held the promise of efficient high-speed rotorcraft—both for edgewise rotors and tilt rotors. For an edgewise rotor, reductions on the order of 50% in RPM along with the intended increase in speed leads to high-advance ratios, μ~0.75 and beyond. The flow field in this regime, characterized by reverse-flow and its impact on blade dynamics, go beyond the nominal capabilities of conventional aerodynamic analysis. The objective of this paper is to bring to bear modern computational fluid dynamics (CFD)/CA analysis for the prediction, validation, and fundamental understanding of this regime.

The desired attributes for a future Department of Defense (DoD) Joint Multi-Role[1] vertical lift aircraft call for high-speed (170–300 kts), extended combat radius (424 km), high/hot performance (6K/95°F), high fuel efficiency, improved survivability, and shipboard compatibility. Conventional helicopter configurations cannot meet these needs, while lack of understanding regarding advanced configurations (compounds, coaxial rotors, lift-offset rotors, and advanced technology tilt rotors) necessitates fundamental research on high-speed, slowed-rotors.

For sixty years, the experiment of Meyer and Falabella[2] on a model rotor offered the only glimpse of reverse flow airloads at a high advance ratio (μ=1.0). There were no cyclic controls on the model; however, and hence the measurements were not representative of powered flight. Although numerous tests and calculations have been conducted in the ensuing years, the recent UH-60A slowed-rotor test[3] provides a new opportunity to gain a comprehensive insight into the details of airloads at a high advance ratio. This experimental investigation indicated unique mechanisms of airloading with direct impact on performance and blade loads—reverse-flow dynamic stall on the lower surface, large regions of negative lift, structural loads as large as a nominal rotor at conventional conditions, and high differential airloads across the span. The purpose of this paper is to provide a high-fidelity computational investigation of this regime—not only to validate the tools by comparing predictions with measurements but to complement and expand the fundamental understanding of this regime.

The earliest CFD predictions of high advance ratio flows were made by Strawn (up to μ=1.2) in 2003 and Theron (up to μ=1.05) in 2006. These unpublished works were later examined and documented by Harris[4], but both calculations were fundamental efforts to establish a baseline for accuracy using idealized test. A more complete historical background and perspective is available in Reference 5.

The present paper carries out a coupled CFD/CA analysis of a full-scale UH-60A rotor slowed down to 40% of nominal RPM, operating at high-speeds of up to 172 kts, and a maximum advance ratio of μ=1.0. This work is the first high-fidelity computational investigation of such a regime in forward flight and has two key goals. Firstly, to validate detailed pressures and airloads, as well as performance, by comparing measurements with analyses under loading conditions unique to this regime. Secondly, to use the validated analysis to complement test measurements and expand fundamental understanding of high advance ratio physics by calculating what is not available from test data: missing airloads at key stations which suffered significant transducer losses, elastic blade deformations (not immediately available), and patterns of dynamic stall and wake visualizations.

2. UH-60A Slowed-Rotor Test (SRT)

A full-scale UH-60A rotor (26.83 ft radius, 4-bladed, fully-articulated) was tested at the US Air Force National Full-Scale Aerodynamics Complex (NFAC) 40-by 80-ft wind tunnel at slowed rotational speeds—65% and 40% of nominal—with resulting advance ratios reaching up to 1.0[3,6]. The test was part of a broader UH-60A airload wind tunnel test program. The final phase of testing was the high advance ratio Slowed Rotor Test (SRT). The objective was to explore the aeromechanics of a conventional (edgewise) rotor in a non-conventional (high advance ratio) regime in anticipation of the next generation of high-speed, high efficiency, variable RPM rotors. It should be emphasized that the UH-60A rotor was not designed to operate at high advance ratio, and in many respects is not optimal for this flight regime, due to its high, nonlinear twist (-16 deg), for example. The SRT test was constructed as a parametric sweep with tip Mach number setting of 0.65 at 100% nominal RPM (NR)—258 RPM, 0.42 at 65% NR, and 0.26 at 40% NR. There are a total of 232 points carried out at the 3 RPMs and 3 shaft angles. The matrix of advance ratios and advancing tip Mach numbers produce a useful separation of compressibility and reverse flow physics. At each test point, the rotor was manually trimmed to minimize 1/rev root flapping. By the SRT phase, many blade pressure transducers had become inoperable. Therefore, integrated airloads are available only at 3 stations (22.5, 86.5, 92.0% span).

3. Helios Software

The HELIcopter Overset Simulations software (Helios) is a multi-disciplinary computational platform being developed by the DoD High Performance Computing Modernization Program (HPCMP) Computational Research and Engineering Acquisition Tools and Environments Program (CREATE-Air Vehicles) and the US Army. It includes modular software components for near-field and far-field CFD, off-body adaption, domain connectivity, rotorcraft comprehensive analysis, mesh motion and deformation, and an exact fluid-structure interface module[7].

The basis of the Helios aerodynamics solution procedure is a dual-mesh paradigm that consists of unstructured meshes in the near-body region and Cartesian meshes in the off-body region (Figure 1). The unstructured meshes allow ease of grid generation for complex configurations, while ensuring proper resolution of the wall-bounded viscous layer region. The Cartesian meshes enable the use of efficient high-order spatial discretizations and adaptive mesh refinement (AMR) to accurately resolve the off-body flow structures. The near-body CFD solver in Helios is the NSU3D code, which is an unsteady Reynolds-averaged Navier-Stokes (URANS) code for unstructured meshes. It utilizes a node-centered scheme that is 2nd-order-accurate in space and time. A backward Euler formulation along with a dual time-stepping scheme is employed for iterative convergence at each physical time-step. The Spalart-Allmaras turbulence model is used. The Cartesian off-body solver in Helios (SAMARC) is a combination of the block-structured meshing infrastructure SAMRAI and the ARC3DC solver, which solves the inviscid Euler equations. The off-body CFD solver uses a temporally 3rd-order explicit Runge-Kutta time integration scheme and a 5th-order central difference spatial scheme with scalar artificial dissipation. The fully-automated domain connectivity formulation is provided by the Parallel Unsteady Domain Information Transfer (PUNDIT) component. The Rotorcraft Comprehensive Analysis System (RCAS) has been used with Helios for CFD/CA coupled analyses. The procedure is the standard loose (delta) coupling. The conservative fluid-structure interface within Helios is a grid-topology-independent (for structured and unstructured mesh) and exact (total airloads/energy) formulation.

4. Computational Modeling – CFD & CSD

An unstructured mesh has been generated for the UH-60A configuration using GRIDGEN and AFLR3 grid generation software. Each blade mesh has 3.9 million nodes, with 61,000 points on the surface. The mesh includes clustering in the volume grid around the blade tip to help resolve the tip vortex. The outer boundary is located 0.7 chords from the blade surface. For the off-body Cartesian grids, 8 levels of resolution are employed, corresponding to a finest level of 5% chord spacing in the wake. A uniformly refined box surrounds the rotor. The complete mesh has 15.4M near-body nodes (36.5M cells) and 145 million active off-body points. The meshes are further detailed in Reference 5. Relevant Helios flow solver parameters are a 0.1 deg azimuthal time-step. Seven sub-steps are used for the explicit off-body solver. Twenty-five near-body sub-iterations are used with two levels of multi-grid. As further verification and comparison of the Helios analysis for the untested high advance ratio regime, coupled CFD/CA calculations were also performed using OVERFLOW 2/CAMRAD II at μ=1.0. The OVERFLOW near-body grids are similar in size to the Helios unstructured near-body grids with 14.4M points, although with double the chord-wise surface resolution. The off-body uses a 10% chord uniformly refined region, so that the total number of points is only 26.1M. However, the OVERFLOW results are 5th-order spatially-accurate in all the grids.

The baseline UH-60A RCAS model has been used for structural dynamics and trim. Convergence of airloads and controls was achieved in 2 to 3 rotor revolutions. Coupling was performed every 90-degrees for the 4-bladed rotor, with the first two couplings at 180-deg increments. The trim condition for low advance ratio, less than 0.8, is the experimental thrust and zero 1/rev flapping angle at the root. At higher advance ratio, it was not always possible to obtain the measured thrust, due to the shallow thrust vs. collective curve. In these cases, the collective has been fixed at the experimental value and only cyclic trim is used.

5. Results

The data points which have been run in CFD for this effort are indicated in Figure 2. Calculations are made at 0 deg shaft angle and 40% rotor RPM. The runs include an RPM sweep at C_T/σ=0.07 and collective sweeps at μ=0.4, 0.6, 0.8, 0.9, and 1.0. In the following sections, discussion will first focus on the effects of the slowed-rotor (RPM sweep), and subsequently on the effects of high advance ratio (μ sweep). A study of the differential aerodynamic loading, sectional airloads, wake visualizations, high structural load mechanisms, and reversed-flow physics follow. Additional details are in Reference 5. In the wake visualization plots throughout the paper, normalized Q criterion iso-surfaces are shown colored by vorticity magnitude. The advantage of the normalized quantity is that a constant value (0.5) can be used for all solutions, independent of flight condition, thrust level, or vorticity maximum.

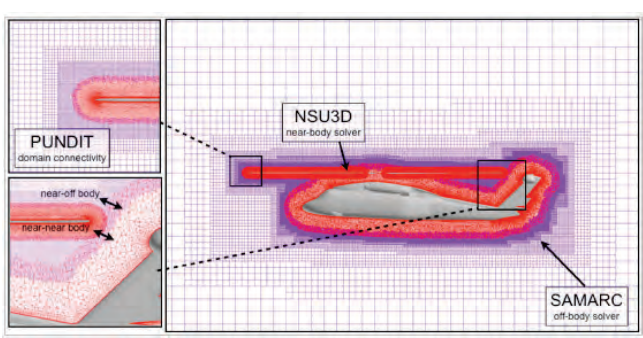


Figure 1. Helios dual-mesh overset CFD approach

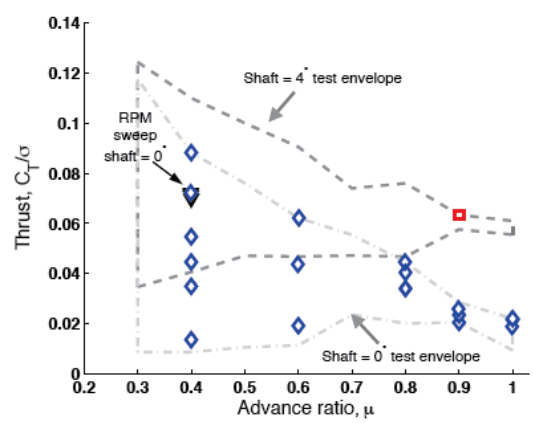


Figure 2. Wind tunnel test envelope and CFD runs

Slowed-Rotor Effects: RPM Sweep

An RPM sweep was performed at 100, 65, and 40% NR at C_T/σ=0.07 and μ=0.4. This investigation offers a particularly good opportunity to validate the CFD and CSD for all the associated effects of a slowed-rotor, while keeping the reversed-flow region fixed. This includes capturing compressibility and variations in structural dynamics behavior.

For the RPM sweep (points 6,912, 8,724, 9,125) the free-stream Mach ranges from 0.26 to 0.10 and the advancing tip Mach number reduces from transonic 0.91 to subsonic 0.36.

To validate the results, normal force (NF) and pitching moment (PM) airload comparisons are shown in Figure 3. In these figures the coefficients c_n and c_m, based on the local tangential velocity, are shown. Without compressibility, structural dynamics variations, and gravity, these coefficients should be the same for the different RPMs at constant μ and C_T/σ. Overall, the magnitude and trend comparisons are very good for NF and chord force (not shown), while only slightly worse for PM. Outboard of 75% span, PM for the 100% RPM case shows increased negative (nose-down) moments and reduced NF throughout the 2nd quadrant. In comparison for the lower RPMs, these 2nd quadrant effects are much reduced. The increased PM magnitude and transonic effects at high RPM and the resulting effect on NF is also influenced by the softer torsional stiffness of the high RPM configuration. In the reversed flow region (22.5% span), the trends are in reasonable agreement.

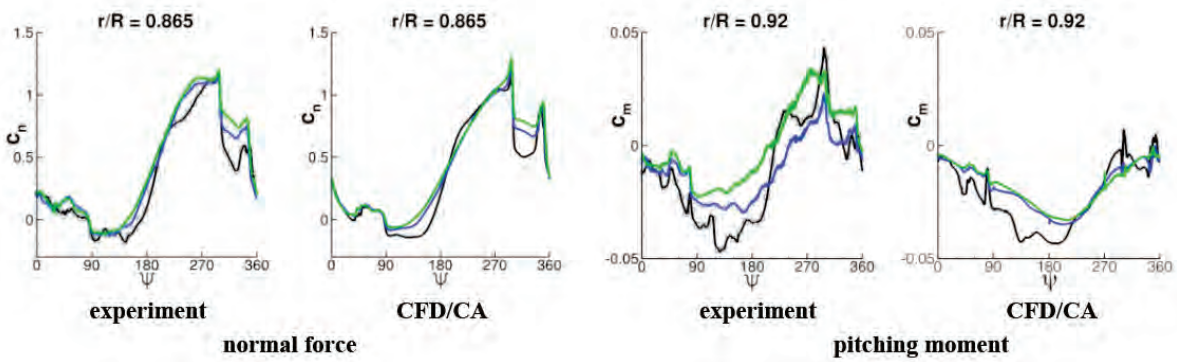


Figure 3. RPM sweep (μ=0.4, C_T/σ=0.07) aerodynamic coefficients

Key Aerodynamic Features

The key aerodynamic features of high advance ratio flight, which distinguish this regime from more conventional helicopter operating conditions, can be discerned in the global patterns of rotor disk normal force. Figure 4 shows the normal force distribution for low and high advance ratios. Figure 5 is the corresponding pitching moment and chord force distributions associated with the μ=1.0 condition (normal force in Figure 4b). Referring to the advance ratio 1.0 figures, several important phenomenons are identified and will be discussed in the following sections. In general, the reversed-flow region, and much of the retreating side of the rotor disk, carries little thrust although large impulsive positive pitching moments are associated with this region. Additionally, the airfoil sections in the reversed-flow region usually generate a dynamic stall vortex on the lower surface. On the other hand, the advancing side quadrants contain large positive and negative loading. Keeping in mind the lateral trim constraint, this must integrate to match the rolling moment generated by the retreating side of the rotor disk. Thus, the large resulting differential loading on the advancing side is understandable. This is in comparison to the smaller amount of negative loading typically associated with the lower advance ratios. The outer portion of the blade enters the advancing side negative lift region with an abrupt lift (and moment) impulse. On the aft portion of the rotor disk, around 0 deg azimuth, the blade interacts with discrete and coalesced root vortices.

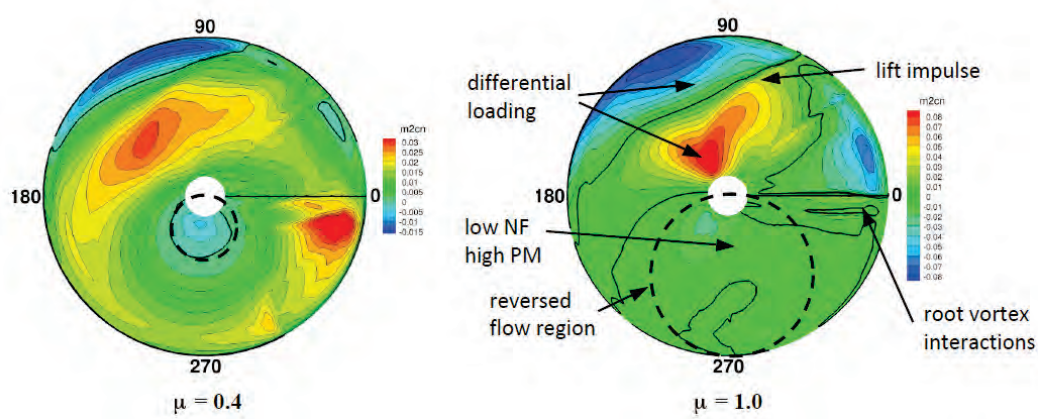


Figure 4. Normal force loading at varying advance ratio: a) 0.4 (9125), b) 1.0 with reversed-flow region

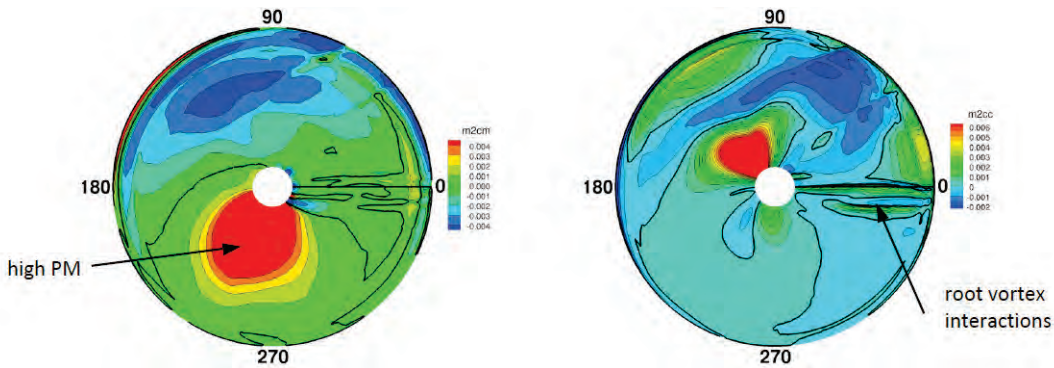


Figure 5. Point 9175, μ=1.0: section a) pitching moment, and b) chord force

Understanding of Airload Details

Detailed CFD airloads—normal force, pitching moments, and chord force (M^2c_n, M^2c_m, M^2c_c)—for advance ratios 0.4 and 1.0 are shown in Figures 6 and 7. In general, the NF details are exceptionally well-captured. The PM trends, peak-to-peak values and overall magnitudes are relatively poor given past correlation experience at nominal RPM. PM results seem particularly poor at the 92% span (μ=0.4). This may be attributed to slowed-rotor effects, but the ultimate cause is not known. In the reversed flow region the PM peak-to-peaks are uniformly over-predicted. The characteristic sharp increase followed by a more gradual decline seen in the test data is not mirrored in the analysis, which shows a wider, more symmetric distribution. Chord force (CF) at the inboard station appears accurate, while it suffers from known integration errors outboard[8]. At the higher advance ratio (Figure 7) the large advancing blade differential loading (positive inboard, negative outboard) is well predicted. The increased extent of the reversed-flow region at the outboard span station appears as a characteristically low normal force. For μ=1.0 airloads OVERFLOW 2/CAMRAD II and Helios/RCAS give similar results, which has been shown previously for conventional UH-60A flight conditions[7], and now is confirmed for a slowed rotor, high advance ratio condition. Taken collectively, across the advance ratio sweep, the trends and magnitudes of normal force are well-predicted, especially as relates to some minor flow-field details. The pitching moment trends are poor, and the peak magnitudes are uncharacteristically under-predicted outboard in the negative lift region, but over-predicted in the reversed flow region inboard. This over-prediction on the retreating side has important implications for pitch link loads.

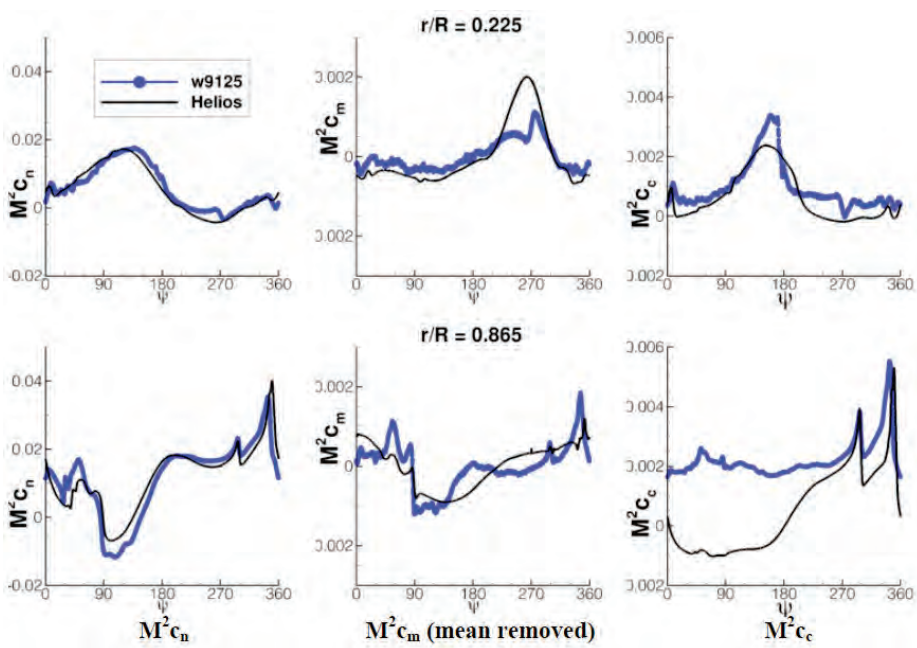


Figure 6. Point 9125, μ=0.4, C_T/σ=0.072

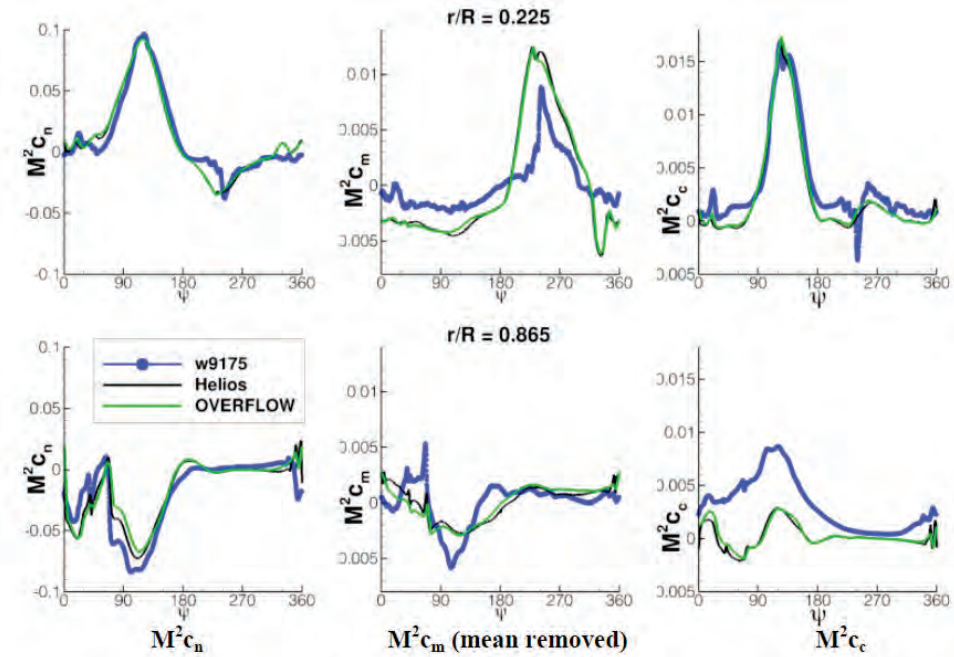


Figure 7. Point 9175, μ=1.0, CT/s=0.022

Advancing Side Lift Impulse

The slowed, high advance ratio rotor is well-suited for fundamental understanding of the advancing side lift and moment impulse that occurs at high μ. The impulse is important because it generates 3/rev lift, which is the dominant forcing for vibratory flap bending. For a nominal rotor RPM the primary mechanism of vibratory lift is the elastic twist driven by unsteady 3D transonic pitching moments near the tip. For a slowed-rotor, this transonic and torsional mechanism is absent or reduced, hence, a secondary wake mechanism is magnified. The strong normal force and pitching moment reduction around 90-deg (Figure 4b and 5a) is now due to the interaction of the blade with an opposite sense vortex created in the 2nd quadrant, due to the negatively loaded tip and lateral trim constraints. Figure 8 colors a wake visualization by the rotational sense of the vortex ($\omega \cdot (r x \Omega)$) with respect to the rotor rotation and a lifting blade (older wake ages and the hub region may be mischaracterized). The (blue) negative tip vortex is initiated in the 1st quadrant and reverts sign late in the 2nd quadrant when the blade tip is positively loaded again. Thus, the lift and moment impulse can be generated entirely in subsonic flow, as seen on the slowed-rotor. The strength of the advancing side impulse is a function of thrust. It is more marked at a lower thrust level and diminishes in strength as thrust increases, i.e., as the negative loading and negative sense tip vortex strength decreases, which is observed in both the experimental and CFD data. The vibratory airloads (3/rev and higher) and the magnitude of the 3/rev lift are consistent with the strength of the impulse. The trends are the same for measured and calculated values (now shown), but there is an under-prediction in 3/rev strength of about 25% at low thrust and 75% at high thrust. In comparison, the 4 and 5/rev components are extremely well-captured, probably due to exceptional agreement on the advancing side impulse.

Figure 8. Sense of vortex rotation at μ=0.4: red – positive tip lifting sense, blue – negative tip lift and positive root lifting sense

High Advance Ratio Wakes

Helios high-fidelity wake visualizations are shown in Figure 9 (top view) for advance ratios 0.4 and 1.0. These unique images show the distinct development and convection of the vortices from the blade tips, blade roots, swept tip break (high geometric twist region), and edge of the reversed-flow region. The root vortices appear to coalesce, and their interaction with the blade at 0 deg azimuth is seen in the previous airload plots. The lateral asymmetry of the wake geometry increases with advance ratio. Quadrant 3 at high advance ratio has almost no influence of the rotor wake in this view at the highest advance ratios.

Throughout the azimuth range of the reversed flow region (dashed circle), at its outboard extent, a vortex is generated from the trailing-edge. It convects over the top of the blade and downstream. At low advance ratio it interacts with the root vortex. Figure 10 shows more details of the development of this reversed-flow-edge vortex and the reversed-flow in general across a range of advance ratios (0.4, 0.8, 1.0). The view is of the bottom of the retreating blade. Flow is from the top. At μ=1.0, the reversed-flow region extends to the blade tip (Figure 10c). Interestingly, in this case vortices are generated from both the leading- and trailing-edges of the tip. The trailing-edge vortex is associated with the edge of the reversed flow. The leading-edge vortex is a "conventional" tip vortex generated due to circulation on the blade. At the higher advance ratios (μ>0.6) blade-vortex interactions can be self-induced on the retreating side, with the rolled-up tip vortex hitting the blade that generated it (blade-on-blade interaction) (Figure 10b). The 0.8 advance ratio image is taken at a high collective, and the pitch of the retreating blade is further increased by large longitudinal cyclic values, a total of 19.2 deg. In this case the reversed-flow region generates vorticity in the guise of a dynamic stall vortex along the underside of the blade. This dynamic stall vortex is not present at the μ=1.0 conditions when the retreating blade is at a lower pitch (1.0 deg) (Figure 10c). Overall, given the very complicated flow physics of the wake and in the reversed-flow region seen in these images, it seems unlikely that a free-wake model will be able to successfully capture the details of the physics.

Figure 9. Wake visualizations with advance ratio μ=0.4 (9125) and 1.0 (9175) with reversed-flow region (Top view)

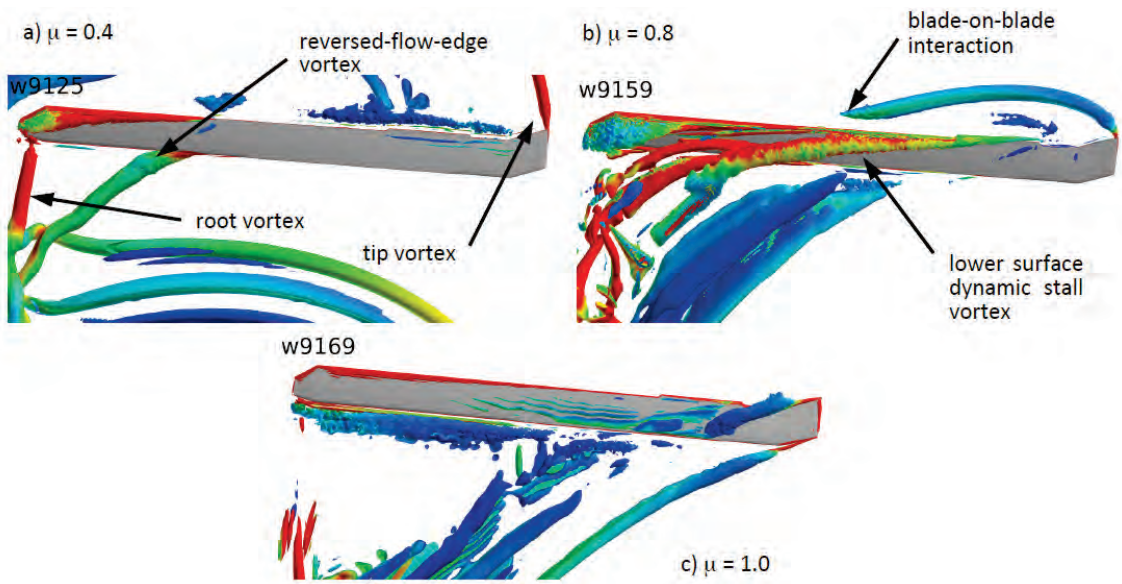


Figure 10. Wake visualizations with advance ratio a) 0.4 (9125), b) 0.8 (9159), and c) 1.0 (9169) – bottom view of retreating blade (Flow from top to bottom)

Reversed-Flow Physics

A major aerodynamic feature of the high advance ratio flight regime is the existence of the reversed flow region on a large portion of the blade. This phenomenon has a direct impact on rotor trim and structural loads in that it generates only a zero or small negative lift (Figure 4) but large pitching moments (Figure 5) and high-drag. The normal force distributions on the retreating side are associated with a trailing-edge down negative angle-of-attack for the reversed flow airfoil.

A qualitative and insightful picture of the retreating side airfoil section flow-field is visualized in Figure 11, which shows vorticity contours and corresponding surface pressures at 55% span at several retreating side azimuth angles. Follow along in both views: The aerodynamic leading-edge (geometric trailing-edge) vortex appears at 220 deg. The dynamic stall vortex progresses over the airfoil from 230 to 260 deg azimuth. At 240 deg azimuth the vortex appears centered over the airfoil mid-chord, as indicated by the characteristic pressure signature. Its strength initially increases as it moves aft on the airfoil. At 290 deg it drops off the aerodynamic trailing-edge, where large negative pressures on both the upper and lower surfaces are induced. By 310 degs the reversed-flow airfoil is in deep stall (fully-separated), and at 330 it returns to conventional operation. In Figure 11, the vortex appears overly diffused. Even though the grid and the azimuthal time-step for these calculations is reasonable by three-dimensional unstructured mesh rotor calculation standards, two-dimensional calculations have shown the need for much higher grid-density (perhaps even an order-of-magnitude) and even smaller time-steps and tighter temporal convergence in order to accurately capture the flow physics of dynamic stall[9].

A quantitative validation of the dynamic stall progression is shown in the lower surface pressure coefficients in Figure 12 at 40% span. The trends between the experiment and computation are similar. But the strength of the computational stall vortex seems to be greatly under-predicted and the computational vortex progression occurs at a slightly-delayed azimuth.

The large pitching moments on the retreating side (Figure 5a) are associated with the reversed-flow and the dynamic stall vortex. The reversed-flow causes movement of the center of pressure towards the quarter-chord of the trailing-edge forward airfoil, resulting in high-loads about the blade geometric ¼-chord.

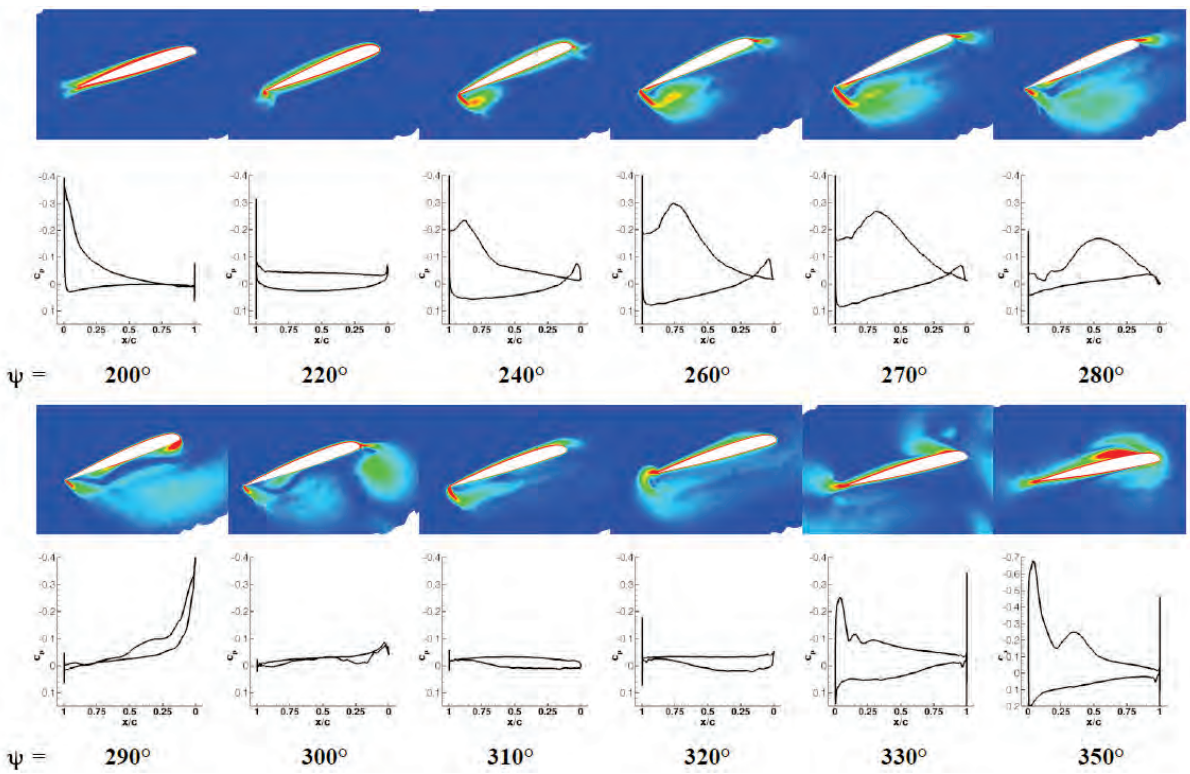


Figure 11. Azimuthal progression of vorticity contours and surface pressures at μ=0.8 on the retreating side reversed-flow, r/R=0.55

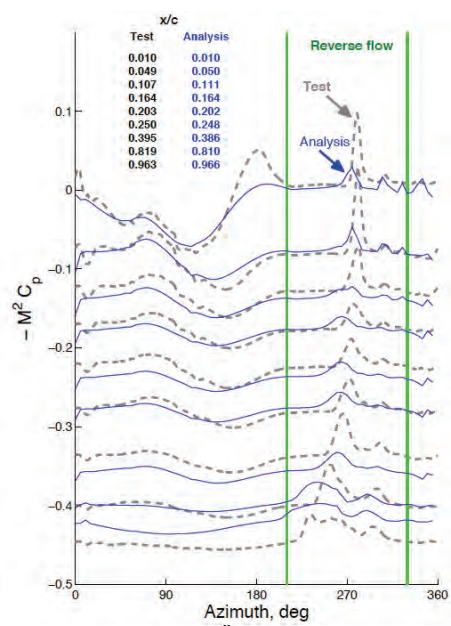


Figure 12. Lower surface pressure coefficient (M^2c_p) at μ=0.8 (9159), r/R=0.400, experiment vs. CFD

Performance

The control angles, cyclics and collective, predicted by the CFD/CA analyses for the range of advance ratios are compared with test data in Figures 13–15. Overall the agreement is favorable. Note that the fixed system control measurements, as used here, have an accuracy of 0.1–0.3 deg. For the low advance ratios it was possible to trim to thrust and the collective angles are in agreement to less than 1 deg. At higher advance ratios, it was not possible to trim to thrust

due to the lack of thrust sensitivity to collective control input, i.e., in the limiting case at μ~1.0, all collective angles give essentially the same thrust. These fixed collective runs at high μ show some significant differences. What is essentially a thrust offset discrepancy may be attributed to incorrect geometry modeling (hub, shank), erroneous lift predictions in the reversed flow region, or experimental issues. The lateral cyclic, θ_{1c}, (Figure 13) is especially well-predicted, except oddly at high thrust and low advance ratio. Figure 14 shows that the longitudinal cyclic, θ_{1s}, is very well-predicted at low advance ratios. At higher advance ratios the slopes are in good agreement, but the absolute values are shifted.

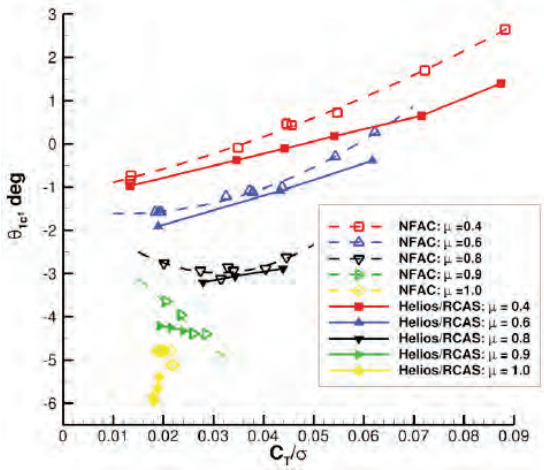


Figure 13. Lateral cyclic vs. thrust

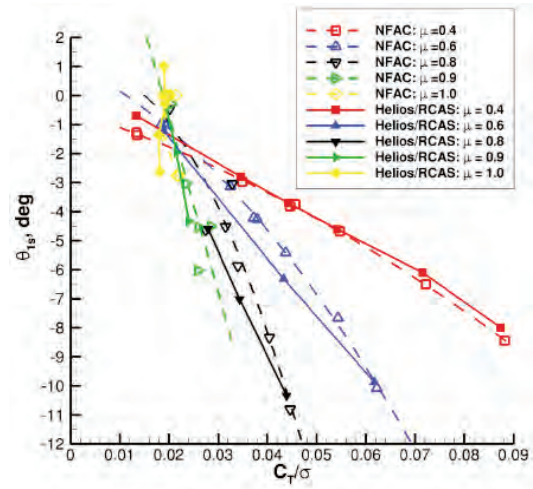


Figure 14. Longitudinal cyclic vs. thrust

Performance prediction is not a main objective of this work, but is included here for completeness and as an initial benchmark of CFD capability on realistic slowed-rotor and high μ configurations. For many reasons performance prediction, particularly for rotorcraft, has not been a strong-suit for CFD[5]. Additionally, these full-scale experimental datasets and the associated data reduction are extremely complex. Years of effort go into fully understanding the data and its limitations. The thrust vs. collective (Figure 15), shows excellent agreement in slope for all advance ratios. A very small thrust reversal is seen in the CFD results at μ=1.0 for collective greater than 1.0 deg. The CFD was even carried out to additional collective angles (up to 5 deg). The thrust reversal in the test data at μ=1.0 is denoted by a single data point at 1 deg collective. Drag predictions (Figure 16) show some trend agreement, although the absolute comparisons are quite poor. The agreement is best at low advance ratio and under-predicted by almost 50% at μ=1.0. Helios/RCAS and OVERFLOW/CAMRAD II are in good agreement with each other at μ=1.0. Figure 17 plots the zero-collective drag as a function of advance ratio. A shank drag increment based on an ad-hoc RCAS correction is shown to significantly improve the comparison. Finally, torque prediction is shown in Figure 18. The absolute comparisons are quite poor, while the increments and trends are only fair.

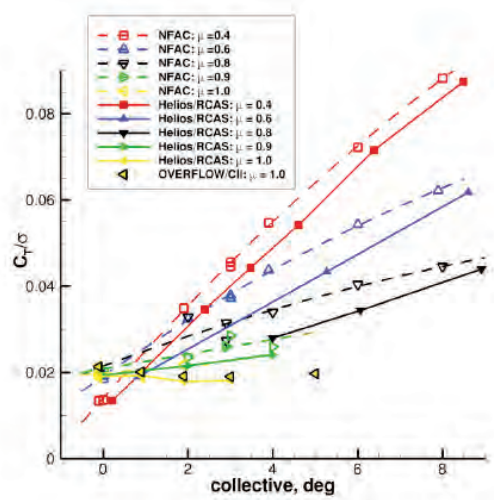


Figure 15. Thrust coefficient vs. collective

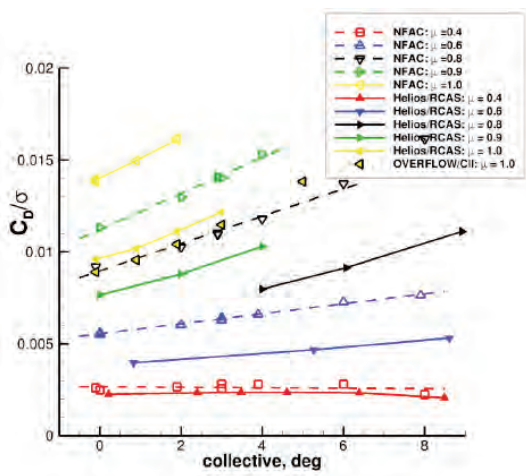


Figure 16. Rotor drag coefficient vs. collective

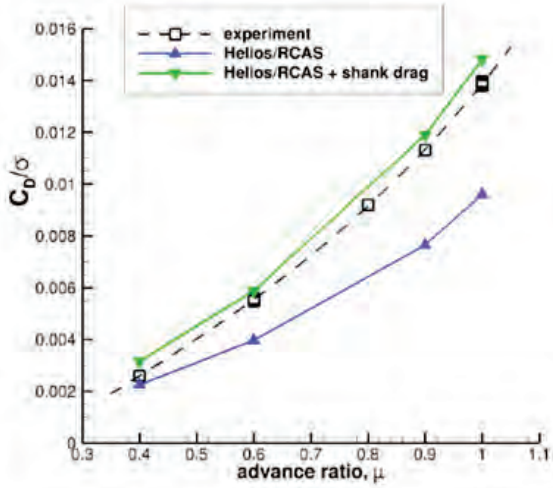


Figure 17. Rotor drag coefficient vs. advance ratio at zero-collective, including shank drag approximation

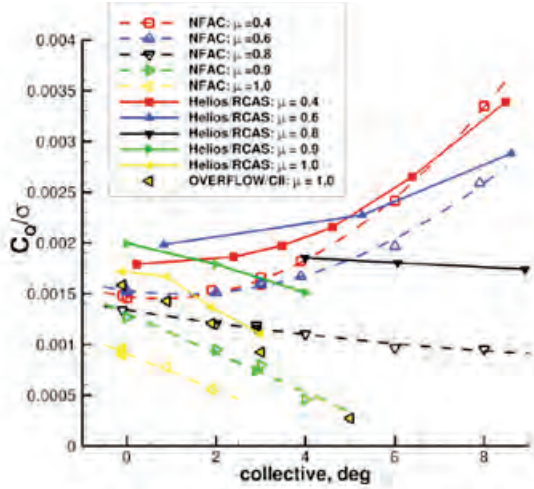


Figure 18. Torque coefficient vs. collective

Based on these results, CFD (Helios or OVERFLOW) would not currently be reliable enough for absolute performance prediction of a highly-twisted, high advance ratio, slowed-rotor. Further investigation of both the computational modeling and the experimental data is certainly warranted in this area.

6. Conclusion

High-fidelity CFD/CA calculations have been performed in order to validate, correlate, and investigate the slowed rotor and high advance ratio conditions from the National Full-Scale Aerodynamics Complex full-scale UH-60A slowed rotor test (SRT). An RPM sweep from 100 to 40% (tip Mach number from 0.65 to 0.26), as well as advance ratio sweeps from 0.4 to 1.0 with varying collective, were studied. The detailed CFD results, in conjunction with experimental measurements, are able to improve our fundamental understanding of the physics in this flight regime. From the results presented, the following conclusions are drawn: 1) Across the advance ratio range 0.4 to 1.0, the overwhelming effects of advancing side differential loading were seen in airloads, structural deformations and trim. 2) Airload comparisons were favorable. Normal force magnitudes, trends, and small aerodynamic details were very well-captured. For the slowed-rotor, negative pitching moment impulses due to wake interactions on the advancing side were under-predicted, while positive impulses on the retreating side due to reversed-flow were over-predicted. 3) The advancing side lift and moment impulse were shown to be caused by the negative sense tip vortex for the slowed-rotor. 4) Wake visualizations showed the flow-fields to be highly-complex and dominated by tip and root vortices, reversed-flow-edge vortices, and on-surface dynamic stall vortices. Numerous blade-vortex interactions were seen, including those involving the root vortex, the advancing side negative lift vortex, and blade-on-blade. 5) Details of the reversed-flow lower-surface dynamic stall vortex progression were shown from vortex initiation to deep stall. Compared with test data, the strength of the dynamic stall vortex was under-predicted due to CFD modeling deficiencies. 6) Performance prediction is not completely reliable for the conditions studied, although drag trends with collective and advance ratio are fairly well-captured.

Acknowledgements

Computing resources under a DoD HPCMP Challenge Project at the NAVY DSRC and the CREATE program are gratefully acknowledged. We thank the NASA Subsonic Rotary-Wing Program and the entire NFAC SRT Test Team for the data and their humongous post-processing efforts to date. Special assistance from AFDD and NASA colleagues (Drs. Robert Ormiston, Jimmy Ho, Thomas Norman, and Randall Petersen) is appreciated.

References

1. Chase, N., "Joint Multi-Role (JMR) Technology-Enabled Capability Demonstration, CTCD." AHS Redstone Chapter Meeting, June 9, 2012.

2. Meyer, J.R., and Falabella, G. Jr., "An Investigation of the Experimental Aerodynamic Loading on a Model Helicopter Rotor Blade." *NACA TN-2953*, May 1953.

3. Datta, A., Yeo, H., and Norman, T., "Experimental Investigation and Fundamental Understanding of a Slowed UH-60A Rotor at High Advance Ratios." *American Helicopter Society Int. Annual Forum 67*, Virginia Beach, VA, May 3–5, 2011; see also refined version at the European Rotorcraft Forum 37, Gallarate (Varese), Italy, September 2011.

4. Harris, F.D., "Rotor Performance at High Advance Ratio; Theory versus Test." *NASA CR 2008-215370*, October 2008.

5. Potsdam, M., Datta, A., and Jayaraman, B., "Computational Investigation and Fundamental Understanding of a Slowed UH-60A Rotor at High Advance Ratios." *American Helicopter Society 68th Annual Forum*, Fort Worth, TX, May 1–3, 2012.

6. Norman, T.R., Shinoda, P., Peterson, R.L., and Datta, A., "Full-Scale Wind Tunnel Test of the UH-60A Airloads Rotor." *American Helicopter Society 67th Annual Forum*, Virginia Beach, May 2011.

7. Sankaran, V., Potsdam, M., Wissink, A., Datta, A., Jayaraman, B., and Sitaraman, J., "Rotor Loads Prediction in Level and Maneuvering Flight Using Unstructured-Adaptive Cartesian CFD." *AHS 67th Annual Forum*, Virginia Beach, VA, May 2011.

8. Biedron, R.T. and Lee-Rausch, E., "An Examination of Unsteady Airloads on a UH-60A Rotor: Computation vs. Measurement." *American Helicopter Society 68th Annual Forum*, Fort Worth, TX, May 2012.

9. Richter, K., Le Pape, A., Knopp, T., Costes, M., Gleize, V., and Gardner, A., "Improved Two-Dimensional Dynamic Stall Prediction with Structured and Hybrid Numerical Methods." *Journal of the American Helicopter Society*, Vol. 56, No. 4, pp. 1–12, October 2011.

Construction of Accurate Reactive Potentials for Large Scale Molecular Dynamics Simulations of Materials Under Extreme Conditions

Edward F.C. Byrd and Neil Scott Weingarten
US Army Research Laboratory (RDRL-WML-B), Aberdeen Proving Ground, MD
{edward.f.byrd2, neil.s.weingarten}.civ@mail.mil

Abstract

Addressing the critical needs in the Department of Defense's (DoD's) Joint Vision 2020 to allow the US Armed Forces to fight and win by creating a dominant force across the full spectrum of military operations, the DoD has invested significant experimental and theoretical resources in the development of advanced materials designed to survive and optimally perform in the extreme conditions found in combat environments. In order to rapidly assess new emerging materials, this requires a fundamental understanding at the atomistic level of the physical and chemical properties of these materials. For large-scale atomistic descriptions of these materials, this requires the use of force field models (due to their low computational cost). However, the effectiveness of these models is completely reliant on the accuracy of the force field. A major limitation in the development of accurate, transferable force fields is the inability of the general user to generate force-fields without requiring the input of an expert developer for the entire process.

Our goal is to demonstrate the ability of the Multiple Objective Evolution Strategy (MOES) software to develop a reactive model of an explosive that will allow predictions of chemical reaction rates and mechanisms for thermodynamic conditions relevant to understanding energetic materials initiation and detonation phenomena, directly supporting the core research programs at the Army Research Laboratory. Determination of reactive force field (specifically, ReaxFF) parameters for carbon, oxygen, nitrogen, and hydrogen through the use of the MOES software will be used in large-scale molecular dynamics simulations of a conventional explosive (RDX).

In the course of our studies, it became apparent that the fundamental ReaxFF equations did not include the appropriate physics to properly account for the systems under consideration in this Challenge proposal. We have successfully incorporated the recently developed (Liu, 2011) long-range dispersion correction terms to the ReaxFF formalism and have modified the MOES software package to take these terms into account. Initial fits of the original ReaxFF parameters are under analysis. We are in the process of generating hundreds of potential candidate force fields using the evolutionary strategies utilized in the MOES for force fields both with and without the additional long-range terms in order to determine the exact effects of said terms. Once we have generated these optimal solutions, we will further test the resultant force fields using molecular dynamics calculations.

1. Introduction

A rapidly expanding area of research within the Department of Defense (DoD) involves the study of materials subjected to extreme conditions, such as shock and thermal loading. Under these extreme conditions, materials often experience radical physical and chemical changes that will influence their performance and survivability. It is, therefore, imperative that a fundamental understanding of material failure, phase transitions, and reactive chemistry be gained in order to design and develop optimally-performing weaponry or protective equipment. Classical molecular dynamics (MD) can be utilized to obtain such basic knowledge. In principle, MD is a straightforward simulation technique that can provide a complete fundamental description of dynamic response of a material to insult. However, the quality of the description is completely dependent upon the interactions within the system, typically defined by an interatomic potential or force field. While it is not difficult to develop reasonably accurate interatomic potentials for simple materials, or for materials for which the MD simulations will explore a small volume of phase space, these potentials cannot correctly describe the complex chemical and physical behavior of materials subjected to shock or thermal loading. More elaborate models are required to capture complexities such as the making and breaking of chemical bonds. Without such "reactive" force fields,

the modeling of materials of interest to the DoD will be limited to low-energy, low-strain conditions, and will not be able to address high-energy phenomenon such as rapid heating, detonation, shear events or any condition where chemistry can occur.

ReaxFF is the most well-developed general reactive force field for use in classical MD simulations of condensed phase material (van Duin, 2001). The goal of the developers of ReaxFF was to create a reactive force field that could be used to describe any chemical system composed of any element in the periodic table. The ReaxFF formalism is based on a bond-order/bond-distance relationship, where contributions to the sigma, pi and double-pi bond order terms are computed from the interatomic distances. The bond order is subsequently used to compute various partial energy contributions to the overall system energy

$$E_{Reax} = E_{bond} + E_{lp} + E_{over} + E_{under} + E_{val} + E_{pen} + E_{coa} + E_{C_2} + E_{tors} + E_{conj} + E_{Hbond} + E_{vdW} + E_{Coul} \quad (1)$$

where the partial energy terms are, respectively, bond energy, lone-pair energy, corrections for atom over-coordination and under-coordination, valence angle energy (including a penalty function E_{pen}), angle conjugation energy, C_2 correction energy, torsion angle energy, torsion conjugation energy, hydrogen bond energy, van der Waals energy and Coulomb energy. More recently, Liu, et al. (2011) published a correction to ReaxFF improving the description of London dispersion through low-gradient corrections (ReaxFF-lg). In this formalism, the total system energy has been modified such that the long-range dispersion correction terms are expressed as

$$E_{Reax\text{-}lg} = E_{Reax} + E_{lg} \quad (2)$$

The second term in Equation 2, the long-range dispersion correction terms, are expressed as

$$E_{lg} = -\sum_{ij,i<j}^{N} \frac{C_{lg,ij}}{r_{ij}^6 + dR_{eij}^6} \quad (3)$$

where r_{ij} is the separation distance between atoms i and j, R_{eij} is the equilibrium van der Walls distance between atoms i and j, d is a scaling factor that is set to 1.0, and $C_{lg,ij}$ is the dispersion energy correction parameter.

Within each of the energy terms in Equations 1 and 2 are a large number of adjustable parameters arranged of seven main types, as seen in Figure 1. ReaxFF is parameterized using a training set of relevant structural and reaction path data that encapsulates the various environments each atom in the chemical system will experience. To date, ReaxFF has been parameterized for a number of systems (for a brief review, see Byrd and Weingarten, 2011). In order to enable a non-expert user to re-parameterize ReaxFF (or any other force field) without reliance on the original force field developers, the Multiscale Reactive Modeling of Insensitive Munitions Software Applications Institute (MSRM) (administered by the US Army Research Laboratory (ARL) and the US Army Armament Research, Development and Engineering Center (ARDEC), under the High Performance Computing Modernization Program (HPCMP) support) created the first massively-parallel Multiple Objective Evolution Strategies (MOES) program for fitting force field parameters (including those of ReaxFF). While the software was first applied to C-H-N-O explosives, MOES was written to address any chemical system. Thus, the impact of the MOES software extends far beyond the scope of Multiscale Reactive Modeling Software Application Institute (MSRM) goals; it could potentially impact any material science project within the DoD involving condensed-phase chemistry.

Our goal has been to use MOES to develop a force field suitable for MD simulations of RDX, but with outstanding transferability to other C-H-N-O type energetic materials. In the next section, we will describe the general computational approach we employ when running MOES to fit force field parameters for the ReaxFF potential, including a description of the training set data. In Section 3, we present results of our fitting routines where only parameters related to E_{vdW}, E_{Coul} and E_{lg} are allowed to vary (all other parameters are held fixed). We focus on the effect of the new long-range dispersion correction term. This is followed by a brief conclusion in Section 4.

2. Computational Approach

2.1 Overview

The process of optimizing a reactive potential requires various components of the MOES software package to communicate with each other in a parallel manner. The overall workflow of the software is illustrated in Figure 2. The software contains three main components: the potential function evaluation engine, the training set data, and the MOES optimization software. The potential function was described in the previous section, the training set data will be described

below, and in previous work (Byrd and Weingarten (2011)) we have provided a complete description of the MOES algorithm.

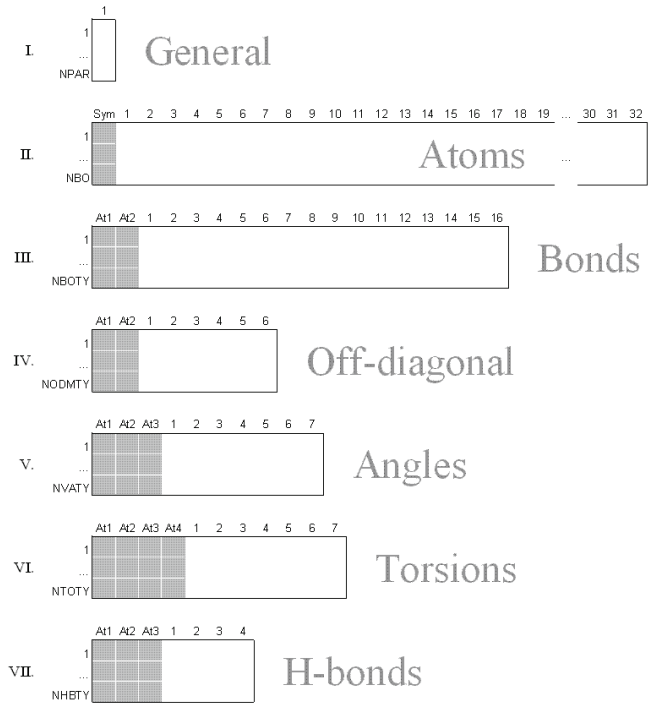


Figure 1. The seven blocks of the ReaxFF force field

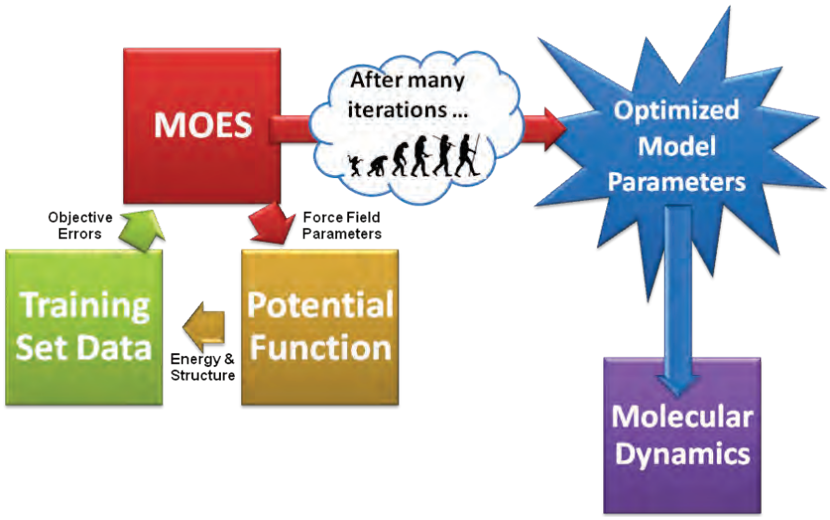


Figure 2. The relation between MOES, the potential, the training set data and the molecular dynamics simulations

In general, the process is initiated by providing MOES with an initial parameterization of the potential function. MOES manipulates the initial parameterization according to a set of strategy parameters and creates the first generation of parent and child parameterizations. These parameterizations are each passed to the potential function evaluation engine to compute the energetic and structural properties of numerous chemical structures. The energetic and structural properties are compared with the reference data contained within the training set and the objective errors are computed and returned to MOES. MOES interprets the objective errors and determines how to best manipulate the parameterizations in creating a new generation of parameterizations. The process is repeated until convergence is obtained. Once the optimized parameterizations are obtained, molecular dynamics simulations are conducted to assess the quality of each candidate force field.

2.2 Description of Training Set

The ultimate goal of this work is to extend the transferability of the potential by supplementing the original training set with additional data. Of particular interest to the MSRM is the development of ReaxFF parameters for carbon, oxygen, nitrogen, and hydrogen for high-energy explosives subjected to a variety of thermodynamic and non-equilibrium extreme conditions. To accomplish this, the original training set was augmented with highly-accurate quantum mechanical information of C-H-N-O explosives, which included ~1,150 single-point energy calculations of the RDX dimers and ~950 FOX7 dimers using the novel symmetry-adapted perturbation theory based on the density-functional SAPT (DFT) description (Podeszwa, et al., 2009).

The figure of merit that is used to determine the fitness of a force field is defined by a list of equations in the training set. In all, the training set consists of nearly 3,600 equations, requiring the evaluation of ~3,600 independent chemical structures and configurations. Associated with each equation is a corresponding weight and corresponding reference data point that is typically obtained via quantum calculations. Each equation lies within a grouping of similar equations. The training set used in this work is divided into 5 groupings that are sectioned according to their physical significance (i.e., atomic partial charges, geometric information such as bonds, angles and torsions, and configuration energies). The 5 objectives contained within the training set and used in this study are as follows:

1. Charges – consists of atomic partial charges of various structures
2. Geometry – consists of bond lengths, angles and torsions of various structures
3. Heat of Formation – Formation enthalpies of various chemical species
4. Energy (Chemistry) – consists of energies of chemical species and configurations
5. Energy (RDX/FOX7) – consists of energies of various RDX and FOX7 configurations

The objective values for each equation in the training set are computed. All of the objective values (errors) of a given group are combined according to

$$Error = \sum_{i=1}^{N_{eqn}} \left(\frac{calc(i) - ref(i)}{weight(i)} \right)^2 \tag{4}$$

MOES identifies each group as a separate objective, even when two groups have the same type (e.g., the Energy objectives listed above). This allows some energies to be optimized independently from others, and the corresponding objective equation is given as

$$Obj(j) = \sum_{i=1}^{N_{eqn}(j)} \left(\frac{calc(i) - ref(i)}{weight(i)} \right)^2 \tag{5}$$

2.3 Screening the Pareto Efficient Force Fields with Molecular Dynamics

Upon MOES generation of the Pareto efficient solutions (i.e., families of ReaxFF parameter sets), the quality of each parameterization (force field) is assessed by performing reactive molecular dynamics (MD) simulations of a 2×2×3 supercell of crystalline RDX at room temperature and pressure using the LAMMPS simulation software (Plimpton, 1995). The MD screening calculations are conducted in two phases. The first screening phase consists of 2,500 time-steps in the NVE ensemble, followed by 2,500 time-steps in the NVT ensemble and 15,000 time-steps in the anisotropic-NPT ensemble (1 time-step=0.25 femtoseconds). The final lattice constants at the 20,000th time-step are compared to the experimental RDX lattice constants. The best force fields are determined by examining the percent volume error and the unit cell vector lengths as compared to experiment.

Once the best force fields are determined, a second screening phase is conducted in the isothermal-isostress (NsT) ensemble for 100,000 time-steps to allow volumetric and cell vector fluctuations of the simulation cell during the course of the simulation. In the NsT ensemble, the shape of the unit cell (i.e., the cell α, β, and γ) will change to maintain constant stress. Thermodynamic and crystallographic properties are calculated and accumulated for averaging over the duration of the trajectory integration, during which atomic configurations are recorded at periodic intervals. From these, thermally-averaged molecular orientation information (i.e., center-of-mass fractional coordinates and Euler angles) for each molecule is generated. The Euler angles describe the orientations of molecules within the unit cells and are determined using a rotation matrix that transforms the principal axes of inertia of each molecule to the space-fixed coordinate system of the molecule (i.e., the x, y, and z Cartesian axes). This allows direct comparison with experimental analogs to assess the quality of the force field at ambient conditions. Additionally, molecular structural parameters, such as bond lengths, bond angles, and ring conformations, for each of the eight symmetry-equivalent molecules in the RDX unit cell are averaged.

3. Results and Discussion

3.1 Overview

Multiple objective optimizations are performed to optimize the ReaxFF parameters. Four separate case studies are conducted to examine the relationship between the long-range parameters (van der Waals, EEM, and dispersion parameters) and the predicted crystal properties:

- Case 1: Reax formalism varying the E_{vdW} and E_{Coul} parameters (46 parameters).
- Case 2: Reax formalism varying the E_{vdW} parameters (34 parameters).
- Case 3: Reax-lg formalism varying the E_{vdW} and E_{Coul} parameters (46 parameters).
- Case 4: Reax-lg formalism varying the E_{vdW}, E_{Coul} and E_{lg} parameters (54 parameters)

The "Reax formalism" refers to Equation 1, and values in the initial parameter set are those provided by Strachan, et al. (2005). The "Reax-lg formalism" includes the long-range dispersion correction described in Equation 2, and the values in the initial parameter set are those provided by Liu, et al. (2011). In all cases, the number in parentheses indicates the number of parameters that are varied during the fitting routine.

3.2 Reax Formalism

3.2.1 Case 1

In the first case study, MOES was conducted to produce a total of 19 converged evolutions, giving a total of 1,900 elite solutions. The Pareto analysis showed that 604 of those 1,900 elite solutions were found to lie on the Pareto frontier. These 604 Pareto-efficient solutions were screened by conducting molecular dynamics simulations with LAMMPS, as described in Section 2.3. The final lattice constants at the 20,000th time-step of the first screening phase were compared to the experimental RDX lattice constants. All force fields that produced crystal structures deviating by less than 2.5% volume error and less than 0.5 Angstrom difference in the *a*, *b*, and *c* unit cell vectors from the experimental values were considered as viable parameter sets. Only eight Pareto-efficient force fields satisfied the criteria and were considered viable solutions. These force fields were further screened by conducting a 100,000 time-step simulation in the NsT ensemble. The average lattice constants and cell vectors from the NsT-MD simulation are compared with experiment and the initial force field in Table 1.

Based on the behavior of the lattice constants, and the molecular structure and orientation, the two best force fields from this case study are FF-0134 and FF-0183. The lattice constants and cell angles are plotted as a function of simulation time in Figure 3, and compared with the initial force field. Both force fields yield crystal structures that are in better agreement with experiment than the initial force field. It should also be noted that it is important to examine the full set of resultant data in these plots rather than just the average value. Without such a detailed analysis, it is difficult to determine whether the cell constants fluctuate about a mean value, or if they exhibit large deviations from experiment that average out over the course of the simulation. Examining the full set of data provides a more complete understanding of how well the force field behaves.

Table 1. Average cell constants for the best performing force fields of Case Study 1

ForceField	*a*	*b*	c	*α*	*β*	*γ*
0048	13.28	11.47	11.19	90.01	89.72	90.68
0134	13.48	11.44	10.61	90.06	90.28	90.03
0183	13.17	11.43	11.14	89.53	89.87	89.76
0251	13.35	11.49	11.02	89.29	90.04	89.62
0366	13.74	11.81	10.67	91.42	89.95	89.10
0421	13.54	11.33	10.54	90.23	89.33	89.69
0440	14.26	12.09	10.85	89.51	89.68	91.36
0572	13.06	11.40	11.17	90.73	90.39	89.36
Initial	13.62	11.97	12.44	90.14	90.39	89.89
Experiment	13.18	11.57	10.71	90.00	90.00	90.00

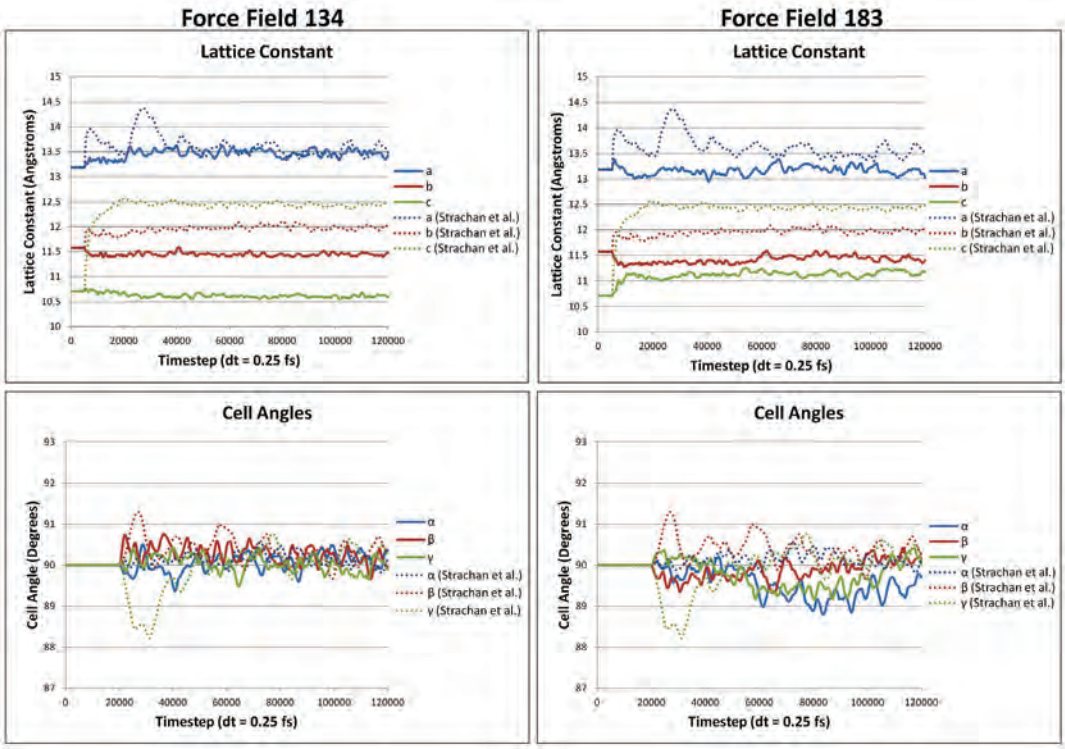


Figure 3. The lattice constants and cell angles as a function of simulation time (solid lines) compared with the initial force field (dotted lines). Simulations are conducted in the NVE (2,500 timesteps), NVT (2,500 time-steps), NPT (15,000 time-steps) and NsT (100,000 time-steps) ensembles at 300K. Note that the simulations commenced at the experimental lattice constants, thus the first 5,000 time-steps correspond to the experimental volume.

3.2.2 Case 2

For Case 2, MOES was conducted to produce a total of 20 converged evolutions, giving a total of 2,000 elite solutions. The Pareto analysis was conducted and 582 elite solutions were found to lie on the Pareto frontier. Of those Pareto-efficient force fields, only three Pareto-efficient force fields produced crystal structures that deviated by <2.5% volume error and <1.0 Angstrom difference in the unit cell vector lengths compared with experiment values. The average lattice constants and cell vectors from the NsT-MD simulation are compared with experiment and the initial force field in Table 2.

Table 2. Average cell constants for the "best" force fields of Case Study 2

ForceField	*a*	b	c	*α*	*β*	*γ*
0069	13.01	10.94	11.70	89.93	89.98	90.26
0503	13.10	11.01	11.72	90.00	89.84	90.28
0574	13.05	10.90	11.65	89.82	90.01	89.82
Initial	13.62	11.97	12.44	90.14	90.39	89.89
Experiment	13.18	11.57	10.71	90.00	90.00	90.00

The lattice constants and cell angles as a function of time were analyzed for the 3 viable force fields. Each force field improved the cell constants as compared to the original parameterization; however, none were particularly good in comparison with experiment, as the *c* lattice vector deviates from experiment by ~1 Angstrom. This suggests that it is important to allow flexibility in varying the Coulomb parameters in conjunction with the van der Walls parameters in order to obtain better agreement with experiment. The remainder of the case studies includes the Coulomb parameters in the optimization.

3.3 Reax-lg Formalism

3.3.1 Case 3

We begin by noting that the only difference between Strachan, et al.'s (2005) initial parameterization and that reported in Liu, et al. (2011) is the inclusion of the dispersion correction parameters. In this case, MOES was conducted to produce a total of 21 converged evolutions, giving a total of 2100 elite solutions. The Pareto analysis was conducted and 705 elite solutions were found to lie on the Pareto frontier. Of those, 10 Pareto-efficient force fields produced crystal structures that deviated by <2.5% volume error and <0.22 Angstrom difference in the unit cell vectors' lengths.

In examining the crystal structure fluctuation over the course of the simulation, FF-0252 was determined to be the best, as compared to the initial parameterization by improving the Geometry objective as well as the lattice constants. The lattice constants and cell angles are plotted as a function of simulation time for this force field in Figure 4 and compared to Liu, et al. (2011). It is observed that FF-0252 provides an improved crystal structure in the *b* and *c* lattice constants than the original force field in comparison with experiment.

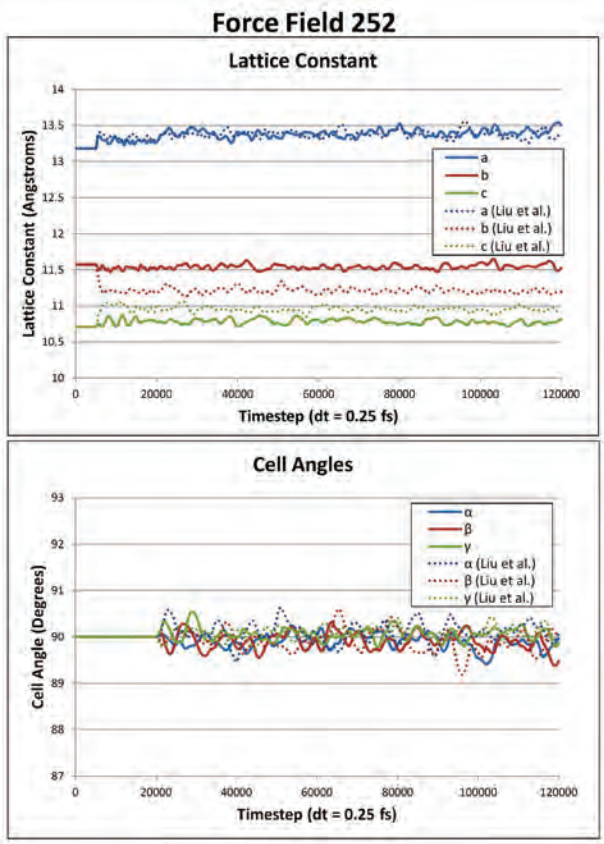


Figure 4. Same as Figure 3, the lattice constants and cell angles as a function of simulation time (solid lines) compared with the initial force field (dotted lines)

3.3.2 Case 4

In the final case, MOES was conducted to produce a total of 17 converged evolutions, giving a total of 1,700 elite solutions. The Pareto analysis was conducted and 576 elite solutions were found to lie on the Pareto frontier. According to the screening protocol previously described, 11 Pareto-efficient force fields produced crystal structures that deviated by <2.5% volume error and <0.22 Angstrom difference in the unit cell vectors' lengths. In examining the crystal structure fluctuation over the course of the simulation, FF-0069 was determined to be the best as compared to the initial parameterization by improving the heat of formation objective as well as the lattice constants, which are plotted along with the cell angles as a function of simulation time in Figure 5.

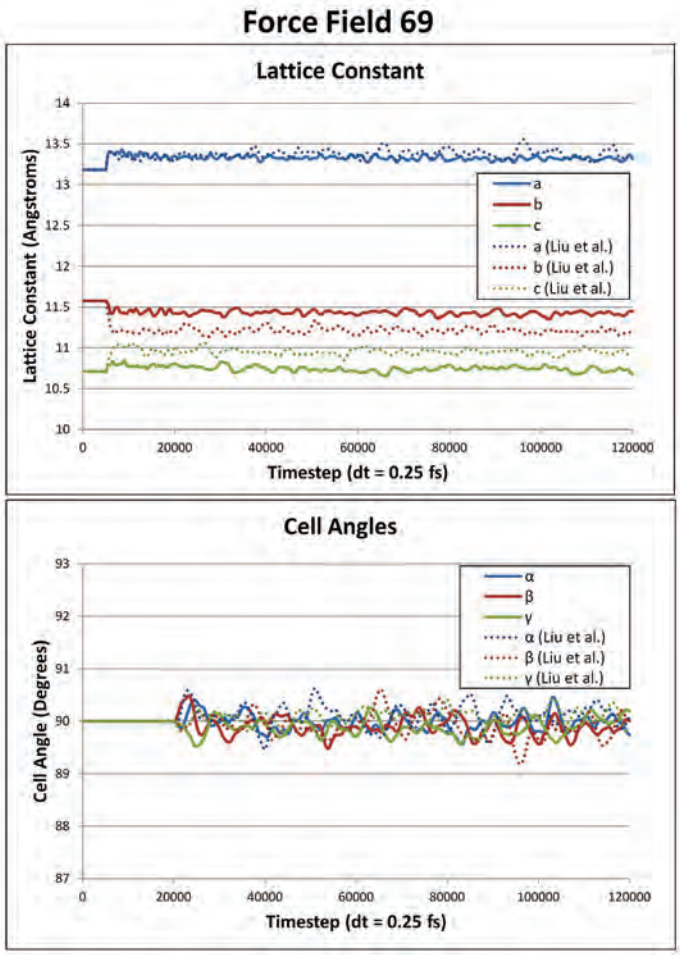


Figure 5. Same as Figures 3 and 4, the lattice constants and cell angles as a function of simulation time (solid lines) compared with the initial force field (dotted lines)

4. Conclusion

We have used MOES, a novel means of fitting interatomic potentials based on evolutionary strategies, to develop ReaxFF force fields for use in classical MD simulations of C-H-N-O energetic materials. The results presented here focus on the force fields generated when only a small subset of the ReaxFF parameters is allowed to vary. In Cases 1 and 2, MOES calculations resulted in improved force fields compared to the original (Strachan, et al. (2005)) when varying only the parameters associated with the van der Walls and Coulomb energies, although the improvement was modest. Moreover, we observed that varying only the van der Waals parameters was not sufficient in order to fully describe the molecular energetic crystal under study here. Cases 3 and 4 examined the effect of the newly added long-range dispersion correction to the ReaxFF formalism. We observed vast improvement in lattice constants compared to Cases 1 and 2, but still only modest improvement compared to the original force field (Liu, et al. (2011)).

So far, we have only tested our force fields on a single energetic material, namely RDX. Future work will involve analyzing the transferability of the force fields generated by MOES, in addition to increasing the number of parameters that we vary during a fitting routine. We are in the midst of generating multiple evolutions of potential candidate force fields. We expect that our simulations will be finished before the end of the fiscal year, and that we will soon have some significant results. Following the successful completion and evaluation of our large-scale evolutions, we will perform further testing to determine how well the generated force fields model *reactive* chemistry.

Acknowledgements

The DoD HPC Modernization Program supported this project by supplying supercomputer time under the Computing Challenge Project C4W. This time was made available at the DoD Supercomputing Resource Centers at the Army Research Laboratory SGI Altix, Harold, the US Air Force Research Laboratory Cray XE6, Raptor, and the Engineer Research and Development Center SGI Altix, Diamond. We would like to thank Betsy M. Rice (ARL), as well as PETTT contractors James V. Lill (AFRL) and James Larentzos (ARL) for their invaluable aid in this project, including data-generation and analysis, and maintaining, updating and modifying the MOES and LAMMPS software packages.

References

Byrd, E.F.C., and N.S. Weingarten, "Construction of Accurate Reactive Potentials for Large-Scale Molecular Dynamics Simulations of Materials Under Extreme Conditions." *Proceedings of the DoD HPCMP Users Group Conference 2011*, pp. 227–233, Portland, OR, June 20–23, 2011.

Liu, L., Y. Liu, S.V. Zybin, H. Sun, and W.A. Goddard, III, "ReaxFF-lg: Correction of the ReaxFF Reactive Force Field for London Dispersion, with Applications to the Equations of State for Energetic Materials." *Journal of Physical Chemistry A*, 115, pp. 11016–11022, 2011.

Plimpton, S., "Fast Parallel Algorithms for Short-Range Molecular Dynamics." *Journal of Computational Physics*, 117, pp. 1–19, (1995). (http://lammps.sandia.gov)

Podeszwa, R., B.M. Rice, and K. Szalewicz, "Crystal structure prediction for cyclotrimethylene trinitramine (RDX) from first principles." *Phys. Chem. Chem. Phys.*, 11, pp. 5512–5518, 2009.

Strachan, A., E.M. Kober, A.C.T. van Duin, J. Oxgaard, and W.A. Goddard, "Thermal decomposition of RDX from reactive molecular dynamics." *J. Chem. Phys.*, 122, pp. 054502, 2005.

van Duin, A.C.T., S. Dasgupta, F. Lorant, and W.A. Goddard III, "ReaxFF: A Reactive Force Field for Hydrocarbons." *Journal of Physical Chemistry A*, 105 (41), pp. 9396–9409, 2001.

Emulation of Forward-Looking Radar Technology for Threat Detection in Rough Terrain Environments: A Scattering and Imaging Study

DaHan Liao, Traian Dogaru, and Andy Sullivan
US Army Research Laboratory (ARL), Adelphi, MD
{dahan.liao, traian.v.dogaru, anders.j.sullivan}.civ@mail.mil

Abstract

Large-scale, full-wave modeling of multi-static target imaging in a rough-ground environment is described. The emulation methodology employs a parallelized three-dimensional "near-field" finite-difference time-domain algorithm in characterizing the electromagnetic scattering from the ground-surface and buried and on-surface targets in the form of landmines and unexploded ordnances; subsequent focusing of the scattered fields into an image is obtained with the time-reversal technique. The emphasis of this study is on investigating the detectability of discrete ground targets in the presence of distributed variable ground clutter as relevant to performance prediction for ultra-wideband forward-looking radar applications.

1. Introduction

Electromagnetic-oriented detection is expected to be an integral component of the US Army's multi-sensor-based strategy for force protection and route clearance. Over the last decade, a myriad of research avenues have been directed toward developing ground-based, ultra-wideband (UWB) radars with the capability to simultaneously penetrate the ground, and image concealed landmines and improvised explosive devices (IEDs) from a standoff distance (Kositsky, et al., 2002; Bradley, et al., 2003; Wang, et al., 2005; Ressler, et al., 2007; Wang, et al., 2008). Despite the fact that experimentation over the last several years has demonstrated the potential of the technology against certain classes of targets, the amorphous nature of the encountered threats has complicated the development and fielding of the technology. One promising system—developed at the US Army Research Laboratory (ARL)—is the low-frequency, UWB synchronous impulse reconstruction (SIRE), imaging ground-penetrating radar (GPR) (Figure 1), which is a vehicle-based platform operating at the nominal frequency band of 500 MHz–2500 MHz with the forward-looking coverage angle spanning approximately 5°–20°, with respect to the horizon. As opposed to conventional downward-looking GPR, a forward-looking GPR such as the SIRE system offers the advantage of having a reduced ground-bounce return due to the use of a sensing geometry at an off-normal—or near-grazing—angle. One of the disadvantages of the forward-looking geometry is realized by first noting that the overall backscattered signal power response of a target, either on-surface or buried, decreases as a function of the illumination-angle approximately proportional to the form of $|(1+R(\theta_{illu}))|^4$, where $R(\theta_{illu})$ is the reflection coefficient of the ground-surface. (Here, it is assumed the predominant interactions are from plane-wave far-field effects—with negligible surface-wave contributions). This behavior of the scattering response makes the standoff detection of ground targets a challenge, especially when the targets are non-metallic and constraints may be imposed on the maximum power that the radar transmitter can radiate.

In general, as shown by Liao (2011)—a study focused mainly on landmines and metallic shells, the radar responses from shallow-buried targets are—depending on the polarization—at least 8 to 20 dB weaker than those from their on-surface counterparts and, hence, are much more vulnerable to interference scattering effects arising from interface roughness and subsurface clutter. As such, the understanding of the scattering responses of distributed and discrete ground clutter (e.g., surface unevenness, soil inhomogeneities, plant roots, and ground debris) is expected to play an important role in assessing the detectability limit of the radar for sensing buried targets. The scattering interactions within a radar scene, inherently, are intricate functions of the radar geometry (i.e., transmitter/receiver-to-target distance, array configuration), transmission and reception polarizations, frequency, as well as the electrical and physical properties of the targets and the terrain environment. From the electromagnetic perspective, in analyzing a low-to-ground system such as the SIRE radar, as the propagation paths

defining the radiowave interactions between the radar transceiver and the targets adhere to the ground, existing analytical and heuristic ground-surface and target-scattering models often prove to be insufficient in characterizing various backscattering phenomena of interest. Accordingly, from the system development perspective, full-wave electromagnetic simulations are desired—and necessary—for an accurate prediction of the performance of the forward-looking radar.

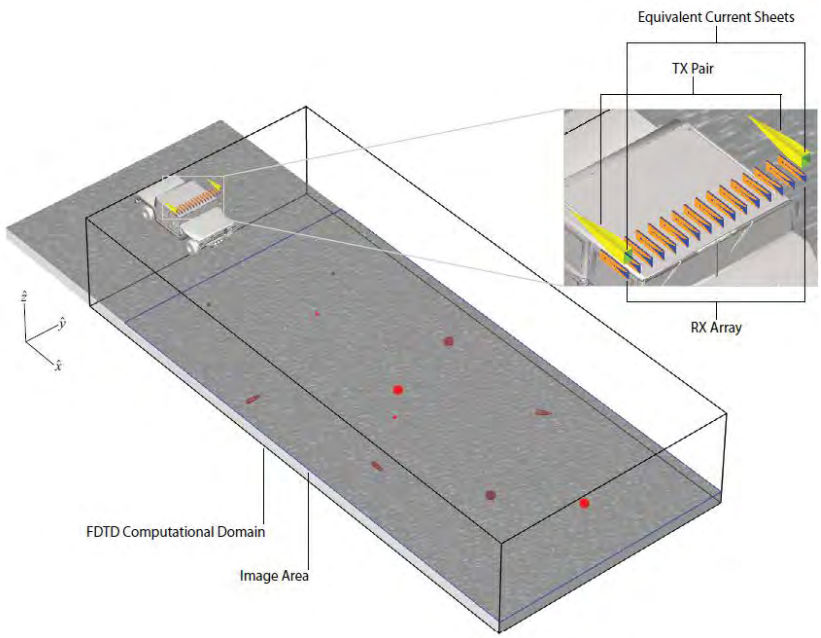


Figure 1. Simulation domain: forward-looking sensing of a terrain scene. Note that the vehicle, transmitting and receiving antennas are not included in the FDTD grid—the transmitting antennas are replaced with equivalent aperture current distributions, and the electric fields are simply sampled at the locations of the receiving antennas for image generation.

Previous efforts in systematically modeling the operation of radio sensing technology for the detection of ground targets include the studies outlined by Bourgeois and Smith (1996); Bourgeois and Smith (1998); Teixeira, et al. (1998); Gürel and Oğuz (2000); Gürel and Oğuz (2001); Lampe and Hooliger (2001); Uduwawala, et al. (2005)—albeit the focus of these works is mostly on the downward-looking sensing geometry, and thus the simulation domains considered therein are relatively small. In the work by Teixeira, et al. (1998), a finite-difference time-domain (FDTD) method is utilized for computing the scattering from metallic and plastic pipes buried in a ground with dispersive soil. To mitigate the direct signal coupling between the transmitter and receiver, in the work by Gürel and Oğuz (2000), the transmitting and receiving elements are housed in a shielding structure coated with absorbers; surface roughness and soil heterogeneities are also included in the model by populating random scatterers within the FDTD computational grid. In both studies above, the transmitters and receivers are assumed to be electric dipoles; UWB radiators more akin to those used in actual GPR systems are analyzed by Bourgeois and Smith (1996), Lampe and Hooliger (2001), and Uduwawala, et al. (2005), where bow-tie antennas are exploited as sources of illumination. In all the aforementioned studies, the emphasis is on characterizing the ground and target-scattering responses, and therefore no imaging results are included. Note that in addition to FDTD techniques, many method-of-moments (MoM)-based approaches also have been proposed (He, et al., 2000; El-Shenawee, et al., 2001a; El-Shenawee, et al., 2001b; Johnson and Burkholder, 2001; El-Shenawee and Rappaport, 2002; Johnson, 2002; Wang, et al., 2003; Altuncu, et al., 2006; Guan, et al., 2009) to facilitate GPR-related numerical investigations—including cases where the ground-surface roughness needs to be taken into account; sensing and imaging at grazing-angles, however, is not discussed in detail within these works.

An effort by the authors to characterize the detection operation of the SIRE radar using a full-wave electromagnetic solver is documented in previous works (Liao, et al., 2010; Liao and Dogaru, 2011a; Liao and Dogaru, 2011b; Liao and Dogaru, 2012), in which only far-field interactions are studied—that is, the excitation of the target and the environment is supplied by pure plane waves, and the receivers of the scattered signal are placed at infinity. In the current work, to achieve a more accurate emulation of the sensing operation employed by the SIRE radar, a "near-field" FDTD approach is proposed to provide system-level analysis, and for tracking the propagation and scattering of the radio-wave energy through the computational domain. This study takes advantage of the computing resources received through the three-year High Performance Computing Modernization Program Challenge Project award entitled "Advanced Computational Modeling of

Forward-Looking Radar for the Detection of Landmines and IEDs." This article provides a summary of the works performed during the first year of the project and is organized as follows: Section 2, the framework of the full-wave-based radar emulator is detailed; Section 3, an investigation of the validity of the "near-field" FDTD algorithm is put forth; and Section 4, a demonstration of the effects of ground clutter on target imaging generated by the time-reversal technique is presented with the simulation of a large terrain scene consisting of targets in a rough ground environment. In sum, this study illustrates the practicality of the chosen full-wave simulation strategy for the emulation of forward-looking radar operation and imaging.

2. Electromagnetic Simulation and Analysis Framework

In demonstrating the modeling framework put forth in this study, electromagnetic simulation and coherent imaging of the scene depicted in Figure 1 is considered. Note that during in-field operations, the ARL SIRE radar integrates scattering responses gathered at multiple vehicle locations to form the final image for a scene. As this is the first phase of the numerical investigation, in order to establish a baseline for validation and identify any potential fundamental limitations of the solver, the radar transceiver is assumed to be fixed in position, with the image synthesized from return signals captured at only a single aperture. With these considerations, the components of the forward-looking radar emulation engine are outlined below.

2.1 Terrain Characterization

The ground is modeled as a non-dispersive, non-permeable homogeneous medium with effective relative dielectric constant ϵ_r and conductivity σ_d. The electrical properties of the ground are dependent upon the soil composition and moisture content; ϵ_r and σ_d can be either determined using direct dielectric measurement techniques or approximated with empirical-based formulations (Dobson, et al., 1985; Peplilnski, et al., 1995). The surface statistics of the ground, in general, are described by two functions: the probability density function of the height variations and the surface autocorrelation function. Within the scope of this study, a two-dimensional zero-mean surface profile obeying Gaussian statistics $\{h_{rms}, l_c\}$—where h_{rms} is the *rms* height and l_c the correlation length—is assumed and generated from its randomized power spectral density function by following the procedure prescribed by Thorsos (1988).

2.2 Ground Targets

As illustrated in Figure 1, targets in the form of landmines (either anti-tank or anti-personnel, metallic or plastic, buried or on-surface) and 155-mm shells (metallic, buried) are considered. Specifically, these targets—the shapes, dimensions, orientations, and locations of which are also indicated by the white outlines in Figure 4(a)—include the following: 1) Buried metallic anti-personnel landmine; 2) Buried plastic (ϵ_r=3.1, σ_d=2 *mS*/*m*) anti-personnel landmine; 3) On-surface plastic (ϵ_r=3.1, σ_d=2 *mS*/*m*) anti-personnel landmine; 4) Buried metallic 155-mm shell; 5) Buried metallic anti-tank landmine; 6) On-surface metallic anti-tank landmine; 7) Buried metallic 155-mm shell; 8) Buried metallic 155-mm shell; 9) On-surface metallic anti-personnel landmine; 10) Buried plastic (ϵ_r=3.1, σ_d=2 *mS*/*m*) anti-tank landmine; and 11) On-surface plastic (ϵ_r=3.1, σ_d=2 *mS*/*m*) anti-tank landmine. Buried targets are positioned at 3 cm below the surface.

2.3 Transmitting and Receiving Antenna Treatment

The SIRE radar employs two TEM horn antennas as transmitters and 16 Vivaldi antennas distributed over a 2 m-wide aperture as receivers. In the current implementation of the algorithm, the TEM horn transmitting antennas are simply replaced by equivalent current distributions; that is, the following electric and magnetic current sheets are positioned at the horn apertures (Figure 1): $\vec{J} = \hat{x} \times \vec{H}$; $\vec{M} = \vec{E} \times \hat{x}$, where the electric and magnetic fields are assumed to be constant over the apertures. The Vivaldi antennas are not included in the FDTD model: the electric fields are sampled at the locations of the 16 antennas for image generation.

2.4 Three-Dimensional "Near-Field" FDTD Algorithm

The composite scattering from the targets and the rough ground is calculated numerically with the FDTD method (Taflove and Hagness, 2005), which directly solves the differential forms of Maxwell's equations in the spatial and temporal domains. In essence, the three-dimensional computational domain is decomposed into small cubic cells, with each cell assigned the material dielectric properties of the corresponding region in space. Electric and magnetic field components are sampled along the cells' edges and faces, and the field spatial distribution is then propagated in time according to a time-marching scheme that implements a discretized version of Maxwell's equations. The entire computational grid is terminated by a perfectly-matched layer (PML) that emulates reflection-less propagation of the outgoing waves in infinite space. In the current application, an infinite half-space (air/ground layered media) is considered as the background environment in which the targets are embedded; the rough ground-surface is introduced as a finite, random perturbation of the flat interface between the two media (the approach is similar to that described by Liao and Dogaru [2012]). The overall FDTD code has been fully-parallelized and runs on large distributed computer systems using the Message Passing Interface (MPI) framework: the computational domain is decomposed into rectangular sub-domains and the FDTD equations are solved separately for each sub-domain within one MPI process. Since only the electric and magnetic field samples in the boundary-layer between two adjacent sub-domains need to be exchanged between the respective processes, the code is highly-scalable, even when the number of MPI processes is in the hundreds.

The field excitation sources in the algorithm are implemented as electric and magnetic current distributions in space ("soft sources"), which are added to the regular FDTD field update equations at the desired sample points. Note that by choosing this approach, direct, full-scale modeling of the transmitting antenna inside the FDTD code is avoided; instead, the antenna is replaced with electric and/or magnetic current distributions by employing the equivalence principle. Computation of these current distributions is performed externally, and can be implemented in various ways—based on approximate analytical formulations or exact numerical analysis. The frequency domain equivalent currents replacing the transmitting antenna are subsequently modulated by the spectrum of an UWB pulse (in this case, a fourth-order Rayleigh pulse), and their time-domain versions are used as excitation sources in the FDTD time-marching scheme. In the numerical examples presented in Sections 3 and 4, the sources are implemented either as an infinitesimal electric dipole, or as uniform electric and magnetic current sheets distributed on a rectangular aperture (the latter approximate the currents produced by a TEM horn antenna, as explained in Section 2.3). During the simulation run, the electric fields at the receiving antenna locations are recorded for image generation; specifically, the frequency domain version of the received electric field components is obtained by deconvolving the excitation pulse from the FDTD simulation results.

For sensing applications at low-depression-angles, one salient feature of field interactions in the vicinity of the air/ground interface is the near cancellation of the direct and reflected waves—as it occurs during the illumination of the target by the transmitter, as well as during the propagation of the backscattered target signal to the receiver. Consequently, the direct fields that propagate between closely-spaced transmitting and receiving points can be many orders-of-magnitude larger than those received via scattering from the target. As such, to properly characterize the complete field behavior, stringent requirements must be placed on the precision and the dynamic range of the solver to ensure very weak scattering responses are captured. Given these considerations, the only solution for obtaining accurate numerical values of the fields scattered by a target with the "near-field" code is to run the FDTD time-marching process twice—once for the configuration that includes the target and once without the target—and then subtract the received fields from the two cases. The final result is the "perturbation" in the total field introduced by the presence of the target, which is exactly the definition of the scattered field. Note that some types of errors associated with the direct (or "unperturbed") field are completely cancelled by this procedure. However, other errors, such as numerical dispersion artifacts, cannot be eliminated this way and must be kept under control by choosing a sufficiently-fine grid discretization.

2.5 Time-Reversal Imaging With Asymptotics

A coherent field integration technique is employed to process the FDTD-computed scattered signals to obtain an image of the illuminated scene. According to the time-reversal, or phase-conjugation, method, since the received scattered field can be written as

$$E_s = \left(\vec{r}_R, \vec{r}_T, \omega\right) = Y\left(\omega\right) G\left(\vec{r}_s, \vec{r}_T, \omega\right) G\left(\vec{r}_R, \vec{r}_s, \omega\right), \tag{1}$$

an approximate objective (image) function can be formed with the following expression (Borcea, et al., 2003; Sarabandi, et al., 2004):

$$O=(\vec{r})=\sum_{\omega}\sum_{R}\sum_{T}E_s^*(\vec{r}_R,\vec{r}_T,\omega)G(\vec{r},\vec{r}_T,\omega)G(\vec{r}_R,\vec{r},\omega); \tag{2}$$

where the asterisk notation denotes the phase conjugation operation; $G(\vec{r},\vec{r}',\omega)$ is the Green's function of the environment ($\vec{r}$ is the position of the observation point; $\vec{r}'$ is the position of the source point); $\vec{r}_T,\vec{r}_R$, and $\vec{r}_s$ are the locations of the transmitter, receiver, and scatterer, respectively; $Y(\omega)$ is the "radiation admittance" of the scatterer—a measure of how easily currents can be induced throughout the object. The cross- and down-range resolution of the image are determined by the sensor aperture and system bandwidth, respectively, of the coherent summation performed in Equation 2. For sensing in the presence of a randomly-varying ground interface, as the exact propagation Green's function is not known, $G(\vec{r},\vec{r}',\omega)$ is approximated by the half-space Green's function; as such, the target image is corrupted by ground clutter.

For the computation of $G(\vec{r},\vec{r}',\omega)$—or for the characterization of radiation from current sources in the presence of a half-space, note that exact closed-form expressions can be formulated in terms of integrals of the Sommerfeld type; however, these expressions are difficult and time-consuming to evaluate numerically (especially when the source and observation points are far apart). To completely avoid the use of Sommerfeld integrals, the transformation of these integrals into asymptotic forms can be carried out by applying the method of steepest descents (Liao and Sarabandi, 2005). The generalized asymptotic expressions—simplified to the half-space propagation scenario—from the work of Liao (2009) are employed here in Equation 2 to achieve a more efficient imaging process.

3. Validation of FDTD Solver

The accuracy of the simulation approach outlined in Section 2 is determined by comparing FDTD-derived results with those obtained using another full-wave technique—namely, MoM, as implemented in the commercially-available electromagnetic software FEKO. Only relatively simple canonical problems are considered in the validation process here, since more complicated scenarios become intractable in FEKO.

The images generated from FDTD-and MoM-computed scattering responses for an on-surface target are compared in Figures 2 and 3: a small perfectly-conducting cube (side length=88 mm) is considered in Figure 2, while a perfectly-conducting cylinder (radius=height=15 cm) is considered in Figure 3. The antenna source is an infinitesimal electric dipole (vertically-polarized) located above the ground at 2 m height; 16 evenly-spaced receiving points are distributed over a 2 m-wide aperture parallel to the y-axis; this receiving aperture is centered immediately below the dipole, at 1.9 m height. The targets are positioned at a down-range distance of either 3.24 m or 18.24 m from the source.

As conveyed in the figures, the FDTD results are in good agreement with those from MoM in characterizing the primary return from the target for all test cases analyzed. Unfortunately, the FDTD response contains a late-time return (secondary "echo") that does not appear in the MoM solution. It is conjectured here that this late-time "echo", which lies 23 to 30 dB below the main response, is purely an FDTD simulation artifact—a byproduct of numerical dispersion inherent in FDTD and of the imperfectness of the absorbing boundary condition. It is also observed that the magnitude of this error can be reduced by decreasing the grid cell size—this has to be done, of course, at the expense of increasing the CPU and memory usage.

4. Large-Scale Simulation and Imaging Results

Ful1-wave characterization of the scene shown in Figure 1 is performed with the "near-field" FDTD algorithm described in Section 2. The computational domain has dimensions 20 m×10 m×4 m and is discretized into 2.6 billion FDTD cells. Parallelized simulations are performed after partitioning the scene into 128 sub-domains (i.e., a total of 128 processors are employed in the MPI network). Approximately 1,000 CPU-hours are needed for each simulation run. As mentioned in Section 2.4, the computation of the scattered field essentially requires two runs: one for the "unperturbed" problem—the source in the presence of the flat ground only, without targets—and one for the "perturbed" problem—with targets and any ground-surface roughness. In view of the fact that the SIRE system employs two transmitters, two separate sets of scattering responses (corresponding to two sources of excitation) are synthesized together to form the image for the scene—that is, the transmitters are excited sequentially, not simultaneously, for each simulation.

The effects of a rough ground-surface on focusing are demonstrated with the simulation results displayed in Figures 4 and 5, which show the co-polarization images of the scene for different degrees of surface roughness: a) flat ground; b) h_{rms}=1.2 cm, l_c=14.93 cm; (c) h_{rms}=1.6 cm, l_c=14.93 cm. It should be emphasized that the incident (direct) field amplitude at the location of each image pixel, as well as the propagation distance from the pixel to the receiver have been taken into account in the imaging process. In effect, each derived image can be interpreted to be a map containing the spatial variation

of the average bi-static radar cross section (RCS) $\bar{\sigma}(\vec{r}_s)$ for the scene, in which the averaging operation is carried over multiple transmitter/receiver locations and a range of frequencies. Consequently, consistent with this RCS interpretation of the observed image intensities, a particular scatterer would appear brighter the closer it is to the radar—due to illumination at a larger depression angle (i.e., away from grazing-angle) that occurs in the near-range. (Depending on the "normalization" method utilized, the image can also be construed as a map of the electric surface and volumetric current densities induced throughout the scene by the source; this is realized by noting that the term $Y(\omega)G(\vec{r}_s,\vec{r}_T,\omega)$ in Equation 1 is basically a transformation of the incident fields into currents.) From the analysis of the point-spread function (Appendix), note that the sensing resolution for the current setup is $c/2\Delta f$ in the down-range direction and $\lambda_c r/L_{eff}$ in the cross-range direction, where c denotes the speed of light; Δf is the frequency bandwidth; λ_c is the wavelength at the center frequency; r is the range distance; and L_{eff} is the effective aperture length—in this case, it is the sum of the lengths of the transmitting and receiving apertures.

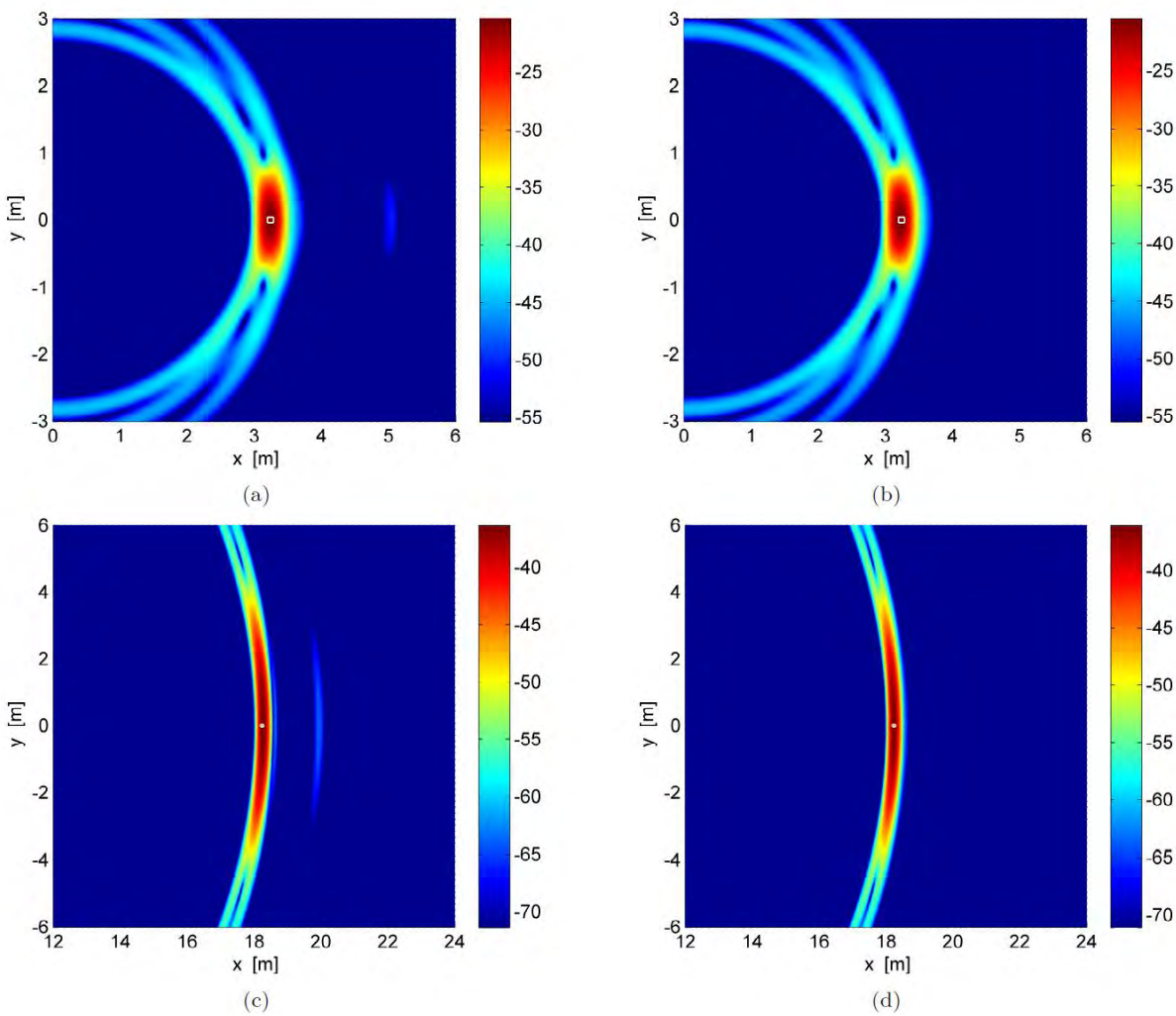


Figure 2. Comparison of FDTD- and MoM-simulated imaging results for an on-surface, perfectly-conducting cube at two ranges: (a) FDTD: range=3.24 m; (b) MoM: range=3.24 m; (c) FDTD: range=18.24 m; (d) MoM: range=18.24 m. Ground properties: ϵ_r=6, σ_d=10 *mS/m*. Frequency span: 0.1–1.4 GHz.

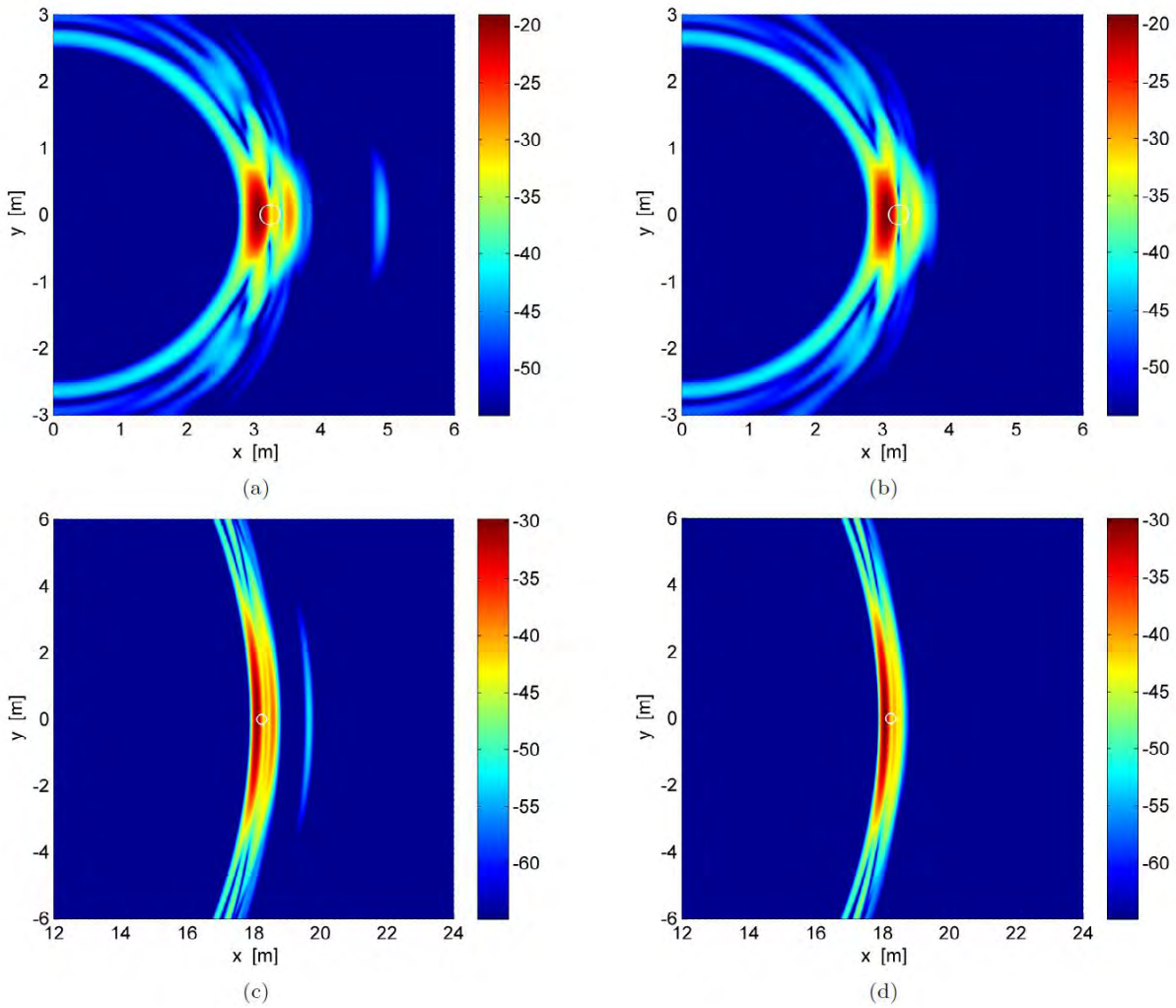


Figure 3. Comparison of FDTD- and MoM-simulated imaging results for an on-surface, perfectly-conducting cylinder at two ranges: (a) FDTD: range=3.24 m; (b) MoM: range=3.24 m; (c) FDTD: range=18.24 m; (d) MoM: range=18.24 m. Ground properties: ϵ_r=6, σ_d=10 *mS/m*. Frequency span: 0.1–1.4 GHz.

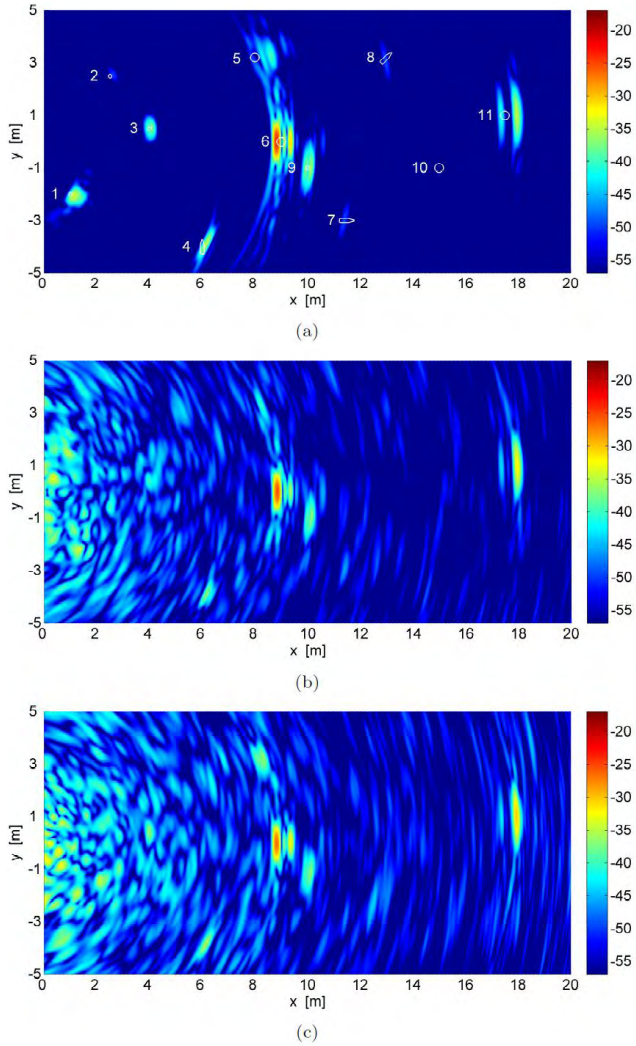


Figure 4. Simulated *vv*-polarized images for 20 m×10 m scene: (a) Ground with flat surface; (b) Ground with randomly rough surface, h_{rms}=1.2 cm, l_c=14.93 cm; (c) Ground with randomly rough surface, h_{rms}=1.6 cm, l_c=14.93 cm. Ground electrical properties: ϵ_r=6, σ_d=10 *mS/m*. Frequency span: 0.3–1.5 GHz.

As seen in the figures, buried targets are generally more difficult to discern from the background as compared to on-surface targets. The response from the larger targets (anti-tank landmines) in the backscattering directions is primarily due to edge-scattering from the front- and back-rims of the top of the cylindrical structure; the scattering centers corresponding to these two edges are resolved in the images—for both on-surface and buried cases. For the smaller targets (anti-personnel landmines), the two scattering-edges tend to merge into one, since the imaging configuration as chosen here is unable to provide sufficient resolution to separate the two phase centers. In the presence of a rough ground interface, these scattering effects are less apparent in the images for the buried targets due to interference from surface scattering. A buried plastic target (either large or small landmine) is especially hard to detect, owing to the limited dielectric contrast between the target and the soil background. (The FDTD artifact discussed in Section 3 can also be seen in these images, when its intensity falls within the dynamic range shown—which is true for targets 5 and 6 at *vv*, for example.) The backscattering return from the ground-surface itself is stronger at *vv* than *hh*, and the response at both polarizations tends to decrease as the incidence angle approaches grazing. These observations are consistent with previously-reported results derived from far-field simulations (Liao and Dogaru, 2011b) and scatterometer measurements (Oh, et al., 1992).

Comparing all target responses across the two polarizations, on average, *vv* provides the stronger return, and it exceeds *hh* by about 5 to 10 dB. As a vertically-polarized signal experiences less transmission loss at the air-ground interface and suffers less propagation loss in general, the above observation supports the notion that *vv* is the more desirable configuration for detecting targets either on top of or buried under a flat ground—at the low-frequency region studied (<1.5 GHz). In the

presence of ground-surface roughness, detection performance must be evaluated with a consideration of the target-to-background clutter ratio. As such, based on the results of these figures, given that the surface response appears as a stronger interference component at *vv*, it is not apparent that sensing with vertically-polarized waves provides a definitive advantage—for the set of parameters examined herein. The statistical distribution of the ground polarimetric scattering response as a function of the surface properties can also be estimated to facilitate a more comprehensive analysis of the target detection performance.

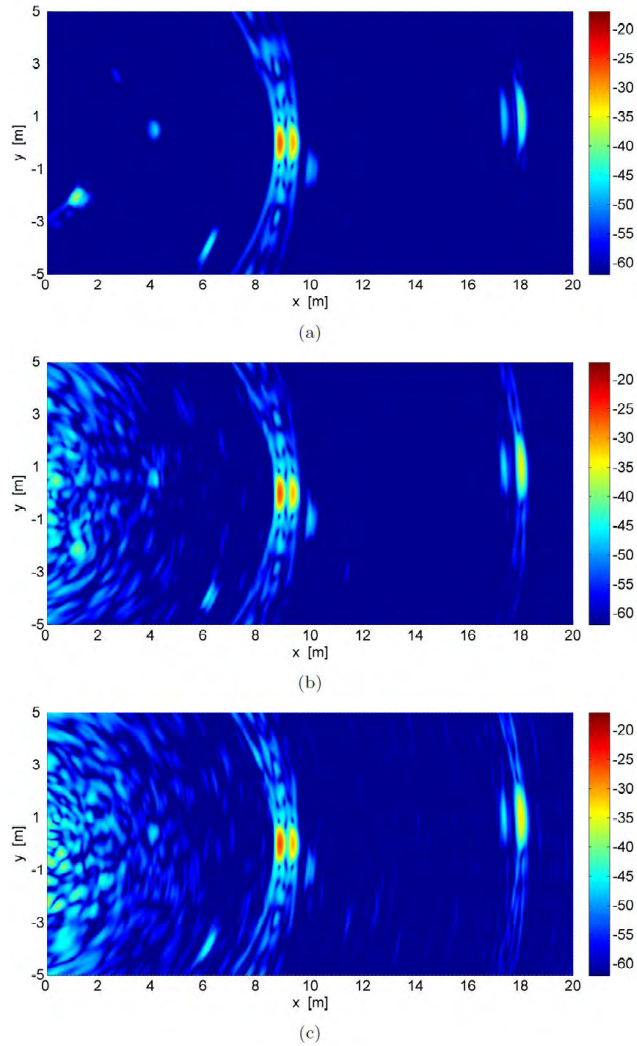


Figure 5. Simulated *hh*-polarized images for 20 m×10 m scene: (a) Ground with flat surface; (b) Ground with randomly rough surface, h_{rms} =1.2 cm, l_c=14.93 cm; (c) Ground with randomly rough surface, h_{rms}=1.6 cm, l_c=14.93 cm. Ground electrical properties: ϵ_r=6, σ_d=10 *mS/m*. Frequency span: 0.3–1.5 GHz.

5. Conclusion

To predict the performance of forward-looking radar technology deployed in complex environments, large-scale, full-wave analysis of multi-static target-imaging in a rough ground environment is described. The emulation methodology employs a parallelized three-dimensional "near-field" FDTD algorithm in characterizing the electromagnetic scattering from the ground-surface and buried and on-surface targets. The transmitting antennas are represented by equivalent current sheets placed at the real antenna aperture locations. Subsequent focusing of the scattered fields at the receiver locations into an image is obtained by exploiting the time-reversal technique, with the aid of asymptotics. The overall simulator is expected to be a useful tool for studying the effects of ground properties— both physical and electrical—on the signatures of various targets.

As evident from the simulation-based validation studies, the primary scattering returns of the targets are accurately characterized by the proposed FDTD approach; however, numerical artifacts—which manifest as "echoes" of the main imaging response—are also observed. Future work includes investigating the fundamental causes of this error component as well as exploring means to mitigate its effects without compromising the computational efficiency of the solver. Efforts in extending the algorithm to accommodate the full models of the transmitting and receiving antennas will also be undertaken. In addition, multi-aperture scattering and imaging simulations will be performed in the next phase of the study to provide a true emulation of the forward-motion-based operational capability of the SIRE system.

Appendix: Derivation of the Imaging Point-Spread Function

Consider an array of vertically-polarized point radiators and a point-scatterer directly on the ground at $\vec{r}_s = (r_s, \theta_s, \phi_s)$, approximating the Green's function for the interactions of the co-polarized field component as (Liao, 2009)

$$G(\vec{r},\vec{r}',k) \propto e^{-ik|\vec{r}-\vec{r}'|}\left\{\frac{1}{|\vec{\rho}|}+\frac{R(\theta_{illu})}{|\vec{\rho}|}+\frac{C_1}{|\vec{\rho}|^2}+\frac{C_2}{|\vec{\rho}|^3}+\cdots\right\} = e^{-ik|\vec{r}-\vec{r}'|}F(\vec{r},\vec{r}'), \tag{3}$$

and assuming observation with an array of isotropic point receivers, the point-spread function (PSF) can be constructed from Equation 2 as

$$PSF(\vec{r},\vec{r}_s) \approx \sum_k \sum_{L_R} \sum_{L_T} e^{ik|\vec{r}_s-\vec{r}_T|}F(\vec{r}_s,\vec{r}_T)e^{ik|\vec{r}_R-\vec{r}_s|}F(\vec{r}_R,\vec{r}_s)e^{-ik|\vec{r}-\vec{r}_T|}F(\vec{r},\vec{r}_T)e^{-ik|\vec{r}_R-\vec{r}|}F(\vec{r}_R,\vec{r}). \tag{4}$$

Using far-field expressions for the distance vectors, the $F(\cdot)$ terms can be moved outside the summation operation, and orienting the transmitting and receiving arrays parallel to the y-axis (x_R=x_T=0), the function in Equation 4—on the xy-plane—becomes

$$\begin{aligned} PSF(\vec{r},\vec{r}_s) \propto\ & F(\vec{r}_s,\vec{r}_T)F(\vec{r}_R,\vec{r}_s)F(\vec{r},\vec{r}_T)F(\vec{r}_R,\vec{r}). \\ & \sum_{n_k}\sum_{n_R}\sum_{n_T} e^{2in_k\delta k(r_s-r)}e^{i(n_k\delta k+k_c)n_R\delta y_R(\sin\phi-\sin\phi_s)}e^{i(n_k\delta k+k_c)n_T\delta y_T(\sin\phi-\sin\phi_s)}, \end{aligned} \tag{5}$$

where δk, δy_R, and δy_T are the k-space sampling interval, the receiver spacing, and the transmitter spacing, respectively; k_c is the center wavenumber; the wavenumber bandwidth is then Δk=$N_k\delta k$, and the aperture sizes are L_R=$N_R\delta y_R$, and L_T=$N_T\delta y_T$. Applying a stationary-phase-like approach to evaluate the summation in k-space and noting the relation

$$\left|\sum_{n=0}^{N-1} e^{-inu}\right| = \left|\frac{\sin\left(\frac{Nu}{2}\right)}{\sin\left(\frac{u}{2}\right)}\right|, \tag{6}$$

the amplitude of Equation 5 approximately simplifies to

$$\begin{aligned} \left|PSF(\vec{r},\vec{r}_s)\right| \propto\ & \left|F(\vec{r}_s,\vec{r}_T)F(\vec{r}_R,\vec{r}_s)F(\vec{r},\vec{r}_T)F(\vec{r}_R,\vec{r})\right|. \\ & \left|\frac{\sin(\Delta k(r_s-r))}{\sin(\delta k(r_s-r))}\cdot\frac{\sin\left(\frac{L_R k_c(\sin\phi-\sin\phi_s)}{2}\right)}{\sin\left(\frac{\delta y_R k_c(\sin\phi-\sin\phi_s)}{2}\right)}\cdot\cos\left(\frac{L_T k_c(\sin\phi-\sin\phi_s)}{2}\right)\right|, \end{aligned} \tag{7}$$

where it is assumed that there is a transmitter at each end of the aperture L_T. The first sin $(\cdot)$ / sin $(\cdot)$ term in Equation 7 dictates the behavior of the PSF in range: it can be shown that the first null occurs at $c/2\Delta f$—which is the resolution in range, δr, independent of r_s and ϕ_s. The periodicity of this term determines the frequency sampling interval needed: for maximum-range $r_{s,max}$, the condition δf<$c/2r_{s,max}$ must be met. Similarly, the cross-range properties of the PSF are controlled by the last two terms in Equation 7. For the current two-transmitter system (and L=L_T=L_R), based on the location of the first null of these two terms, it can be readily seen that the angular resolution depends on the angular position of the scatterer as

$$\delta\phi = \phi_s - \sin^{-1}\left(\sin\phi_s - \frac{\lambda_c}{2L}\right), \quad 0 \le \phi_s < \frac{\pi}{2}. \tag{8}$$

At ϕ_s=0, $\delta\phi \approx \lambda_c/2L$; the angular resolution degrades as the scatterer moves away from the broadside direction of the aperture. In general, the spatial sampling should be less than $\lambda_{min}/2$; however, for an UWB system, it has been shown (Zhuge and Yarovoy, 2011) that high cross-range resolution can be achieved with coarser sampling than the stated criterion.

Acknowledgments

The DoD HPC Modernization Program supported this project by supplying supercomputer time under the Computing Project C5C. The simulations were performed at the US Army Research Laboratory DoD Supercomputing Resource Center at Aberdeen Proving Ground, Maryland.

References

Altuncu, Y., A. Yapar, and I. Akduman, "On the Scattering of Electromagnetic Waves by Bodies Buried in a Half-Space with Locally Rough Interface." *IEEE Trans. Geosci. Remote Sensing*, 44, 6, pp. 1435–1443, 2006.

Borcea, L., G. Papanicolaou, and C. Tsogka, "A Resolution Study for Imaging and Time Reversal in Random Media." *Cont. Mathem.*, 313, pp. 63–67, 2003.

Bourgeois, J.M. and G.S. Smith, "A Fully Three-Dimensional Simulation of a Ground-Penetrating Radar: FDTD Theory Compared with Experiment." *IEEE Trans. Geosci. Remote Sensing*, 34, 1, pp. 36–44, 1996.

_____, "A Complete Electromagnetic Simulation of the Separated-Aperture Sensor for Detecting Buried Land Mines." *IEEE Trans. Antennas Propagat.*, 40, 10, pp. 1419–1426, 1998.

Bradley, M., T. Witten, M. Duncan, and B. McCummins, "Mine Detection with a Forward-Looking Ground-Penetrating Synthetic Aperture Radar." *Proc. SPIE*, 2003.

Dobson, M.C., F.T. Ulaby, M.T. Hallikaiken, and M.A. El-Rayes, "Microwave Dielectric Behavior of Wet Soil—Part II: Dielectric Mixing Models." *IEEE Trans. Geosci. Remote Sensing*, 23, pp. 35–46, 1985.

El-Shenawee, M. and C. Rappaport, "Monte Carlo Simulations for Clutter Statistics in Minefields: AP-Mine-Like-Target Buried Near a Dielectric Object Beneath 2-D Random Rough Ground-Surfaces." *IEEE Trans. Geosci. Remote Sensing*, 40, 6, pp. 1416–1426, 2002.

El-Shenawee, M., C. Rappaport, E.L. Miller, and M.B. Silevitch, "Three-Dimensional Subsurface Analysis of Electromagnetic Scattering From Penetrable/PEC Objects Buried Under Rough Surfaces: Use of the Steepest Descent Fast Multipole Method." *IEEE Trans. Geosci. Remote Sensing*, 39, 6, pp. 1174–1182, 2001a.

El-Shenawee, M., C. Rappaport, and M. Silevitch, "Monte Carlo Simulations of Electromagnetic Wave Scattering from a Random Rough Surface with Three-Dimensional Penetrable Buried Object: Mine Detection Application Using the Steepest-Descent Fast Multipole Method*." J. Opt. Soc. Am. A*, 18, pp. 3077–3084, 2001b.

Guan, B., J.F. Zhang, X.Y. Zhou, and T.J. Cui, "Electromagnetic Scattering From Objects Above a Rough Surface Using the Method of Moments with Half-Space Green's Function." *IEEE Trans. Geosci. Remote Sensing*, 47, 10, pp. 3399–3405, 2009.

Gürel, L. and U. Oğuz, "Three-Dimensional FDTD Modeling of a Ground-Penetrating Radar." *IEEE Trans. Geosci. Remote Sensing*, 38, 4, pp. 1513–1521, 2000.

_____, "Simulations of Ground-Penetrating Radars Over Lossy and Heterogeneous Grounds." *IEEE Trans. Geosci. Remote Sensing*, 39, 6, pp. 1190–1197, 2001.

He, J., T. Yu, N. Geng, and L. Carin,"Method of Moments Analysis of Electromagnetic Scattering From a General Three-Dimensional Dielectric Target Embedded in a Multilayered Medium." *Radio Sci.*, 35, 2, pp. 305–313, 2000.

Johnson, J.T., "A Numerical Study of Scattering from an Object Above a Rough Surface." *IEEE Trans. Antennas Propagat.*, 50, 10, pp. 1361–1367, 2002.

Johnson, J.T. and R.J. Burkholder, "Coupled Canonical Grid/Discrete Dipole Approach for Computing Scattering From Objects Above or Below a Rough Interface." *IEEE Trans. Geosci. Remote Sensing*, 39, 6, pp. 1214–1220, 2001.

Kositsky, J., R. Cosgrove, C. Amazeen, and P. Milanfar, "Results from a Forward-Looking GPR Mine Detection System." *Proc. SPIE*, 2002.

Lampe, B. and K. Holliger, "Numerical Modeling of a Complete Ground-Penetrating Radar System." *Proc. SPIE*, 4491, 2001.

Liao, D.H., *Physics-Based Near-Earth Radiowave Propagation Modeling and Simulation*, Ph.D. Thesis, The University of Michigan, 2009.

_____, "Full-Wave Characterization of Rough Terrain Surface Effects for Forward-Looking Radar Applications: A Scattering and Imaging Study From the Electromagnetic Perspective." *ARL-TR-5758*, U.S. Army Research Laboratory, Adelphi, MD, 2011.

Liao, D.H. and T. Dogaru, "Full-Wave-Based Emulation of Forward-Looking Radar Target Imaging in Rough Terrain Environment." *Proc. IEEE Antennas and Propagat. Soc. Sym.*, 2011a.

_____, "A Comparison of Surface Integral Equation and FDTD Methods for Modeling Backscattering From Dielectric Rough Surfaces at Near-Grazing-Angles." *Proc. IEEE Antennas and Propagat. Soc. Sym.*, 2011b.

_____, "Full-Wave Characterization of Rough Terrain Surface Scattering for Forward-Looking Radar Applications." *IEEE Trans. Antennas Propagat.*, 60, 8, pp. 3853–3866, 2012.

Liao, D.H., T. Dogaru, and A. Sullivan, "Characterization of Rough Surface Clutter for Forward-Looking Radar Applications." *Proc. 27th Army Science Conf.*, 2010.

Liao, D.H. and K. Sarabandi, "Near-Earth Wave Propagation Characteristics of Electric Dipole in Presence of Vegetation or Snow Layer." *IEEE Trans. Antennas Propagat.*, 53, 11, pp. 3747–3756, 2005.

Oh, Y., K. Sarabandi, and F.T. Ulaby, "An Empirical Model and an Inversion Technique for Radar Scattering from Bare Soil Surfaces." *IEEE Trans. Geosci. Remote Sensing*, 30, pp. 370–381, 1992.

Peplilnski, N.R., F.T. Ulaby, and M.C. Dobson, "Dielectric Properties of Soils in the 0.3–1.3 GHz Range*." IEEE Trans. Geosci. Remote Sensing*, 33, pp. 803–807, 1995.

Ressler, M., L. Nguyen, F. Koenig, D. Wong, and G. Smith, "The ARL Synchronous Impulse Reconstruction (SIRE) Forward-Looking Radar." *Proc. SPIE*, 2007.

Sarabandi, K., I. Koh, and M. Casciato, "Demonstration of Time Reversal Methods in a Multi-Path Environment." *Proc. IEEE Antennas and Propagat. Soc. Sym.*, 2004.

Taflove, A. and S.C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Third Edition. Artech House: Boston, MA. 2005.

Teixeira, F.L., W.C. Chew, M. Straka, M.L. Oristaglio, and T. Wang, "Finite-Difference Time-Domain Simulation of Ground Penetrating Radar on Dispersive, Inhomogeneous, and Conductive Soils." *IEEE Trans. Antennas Propagat.*, 36, 6, pp. 1928–1937, 1998.

Thorsos, E.I., "The Validity of the Kirchhoff Approximation for Rough Surface Scattering Using a Gaussian Roughness Spectrum." *J. Acoust. Soc. Am.*, 83, pp. 78–92, 1988.

Uduwawala, D., M. Norgren, and P. Fuks, "A Complete FDTD Simulation of a Real GPR Antenna System Operating Above Lossy and Dispersive Grounds." *Progress in EM Res.*, 50, pp. 209–229, 2005.

Wang, J., Y. Li, Z. Zhou, T. Jin, Y. Yang, and Y. Wang, "Image Formation Techniques for Vehicle-Mounted Forward-Looking Ground-Penetrating SAR." *Proc. SPIE*, 2008.

Wang, T., O. Sjahputera, and J.M. Keller, "Landmine Detection Using Forward-Looking GPR With Object Tracking." *Proc. SPIE*, 2005.

Wang, X., C.-F. Wang, and Y.-B. Gan, "Electromagnetic Scattering From A Circular Target Above Or Below Rough Surface." *Progress in EM Res.*, 40, pp. 207–227, 2003.

Zhuge, X. and A.G. Yarovoy, "A Sparse Aperture MIMO-SAR-Based UWB Imaging System for Concealed Weapon Detection." *IEEE Trans. Geosci. Remote Sensing*, 49, pp. 509–518, 2011.

High Performance Java

Jordan J. Ruloff and James A. Ross
Dynamics Research Corp., Andover, MA
{jruloff, jaross}@drc.com

David A. Richie
Brown Deer Technology, Forest Hill, MD
drichie@browndeertechnology.com

Song J. Park, Dale R. Shires, and Brian J. Henz
US Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
{song.j.park.civ, dale.r.shires.civ, brian.j.henz.civ}@mail.mil

Abstract

At this point in time, it is apparent that future programming paradigms will be based around many-core processors and heterogeneous computing. Diversity in new processor architectures has led to a large variety of processors, which were designed to address different issues found in past architectures while, unfortunately and unintentionally, burdening programmers to use these new architectures effectively. As more programming libraries and languages are developed, programmers will be able to design algorithms for these different architectures to maximize their code efficiency, whether to maximize performance or minimize power usage. Unfortunately, not all code can scale efficiently in many-core architectures nor can all code efficiently utilize heterogeneous architectures. Sometimes, a programmer may even have to deal with a task that is inherently serial in nature. Even if the task is trivially parallel, a programmer may even find that, due to limiting constraints of a particular architecture, like memory or interconnect speed, an algorithm best-suited for a particular problem may not be the most desirable to maximize performance. In order to efficiently utilize the computing hardware, programmers must have a basic understanding of the fundamental differences between the various architectures and how to best utilize them. This paper covers the methods that were employed for addressing task and data parallelism within the Java language to maximize performance of the World Wind Java Ballistic Interface code, Java 7's fork/join framework and AMD's Aparapi Java bindings, as well as the importance of parallel execution time and how to map it to the various execution frameworks.

1. Introduction

The future of high performance computing will be based around many-core processors and heterogeneous computing. While Moore's law has continued to demonstrate itself ever since the invention of the integrated circuit in 1958, Dennard scaling has failed to keep up with the scaling of transistor size (Bohr, 2007). At the moment, the only attainable means for performance growth in a serial application is advancements in processor architecture; hence, it is best to transition from a serial programming paradigm towards a parallel programming paradigm. The task at hand is a difficult one for programmers, especially if one wants to maximize efficiency, but programming in parallel will continue to get easier as more programming libraries and languages are developed. In addition to the task of programming with parallelism, since processor architecture is now playing a larger role in determining performance of a program, a developer must carefully consider the requirements of their program and how to maximize the efficiency of their code on a particular architecture of choice. A quick glance at the plethora of processing architectures available, each designed with a different goal in mind promising high-performance, should make it apparent that this is no easy task. Regardless, tools and application programming interfaces for multi-core and heterogeneous computing are maturing and making it easier to maximize performance of one's program.

The Ballistic Trajectory Field Calculator (BTFC) Interface is a user interface to put the power of a supercomputer in the hands of the US Army Warfighters. The interface allows for the creation, loading, and saving of scenarios, which involve the placements of threats and watch-points on a map. A warfighter can create a scenario, as shown in Figure 1, and submit it to a high performance computer connected to the same network and get back safe positions where they could observe the

watch-points without being detected by the threats, shown in Figure 2. It is critical that the interface maximize performance in order to minimize energy use on a portable device and to keep the interface responsive. The interface was coded in Java on top of NASA World Wind and makes use of the latest parallel programming additions in the Java language as well as Aparapi, a Java library that maps Java code to OpenCL™ code.

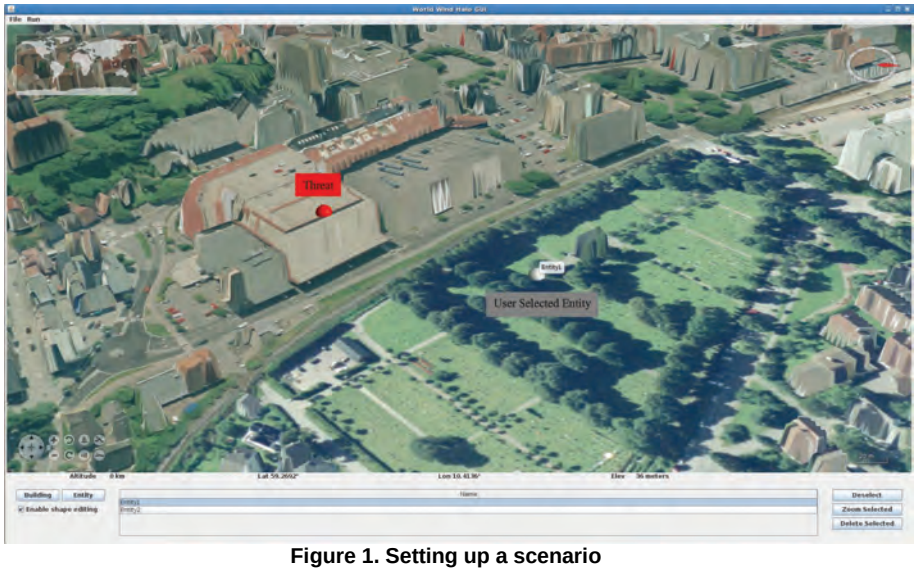


Figure 1. Setting up a scenario

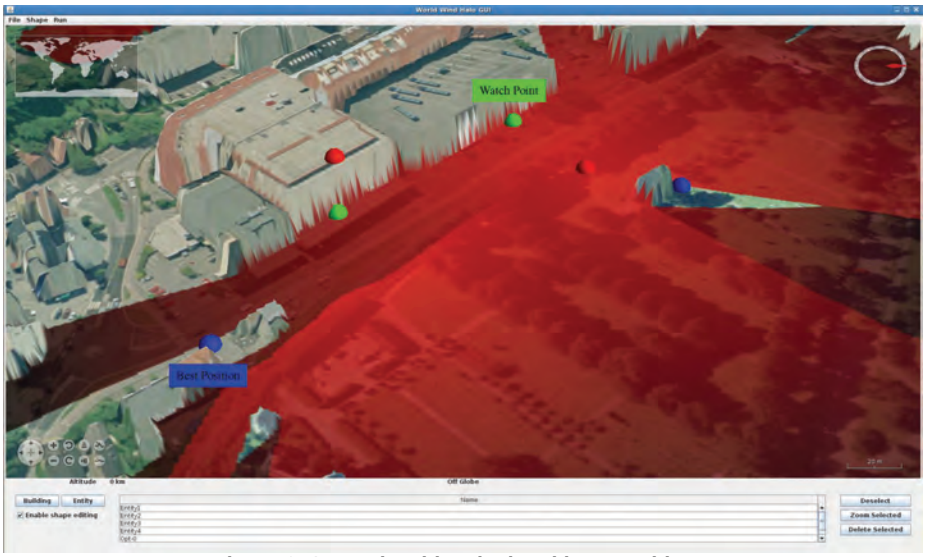


Figure 2. Scenario with calculated best positions

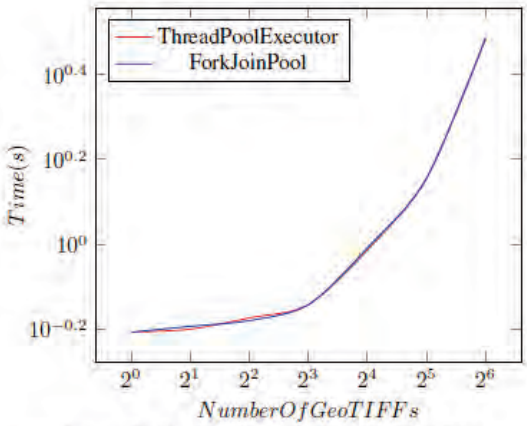
2. Java Parallelism

One of the features that the BTFC Interface supports is the loading of GeoTIFF files for viewing landscapes with higher resolution than what is currently provided by the NASA World Wind server for use in the scenario. The process of loading GeoTIFFs can occur in parallel, except for a small portion which adds the generated data from the files to the interface. It is important to execute the process in the least amount of time possible so that the task does not inhibit the ability of the warfighter to interface with the program. In order to accelerate this process, the BTFC Interface was programmed to efficiently utilize an abstract number of processors through Java thread pooling.

In order to ease multi-threaded programming in Java, the ThreadPoolExecutor class was added in Java 5 (Rajan, Kautz, and Rowcliffe, 2010). It was capable of executing tasks asynchronously and could return values from the other task upon completion and, while it was very efficient and useful, it still failed to handle all the possible cases of parallelism that may

arise in a Java program with maximum efficiency. In Java 7, ForkJoinPool was added to ease multi-threaded programming in Java under conditions that ThreadPoolExecutor could not efficiently handle: unbalanced workloads and recursively subdividing tasks (Lea, 2000). It is important to note that ForkJoinPool does not replace ThreadPoolExecutor, but provides an alternative to ThreadPoolExecutor. In order to evaluate the difference in performance between ThreadPoolExecutor and ForkJoinPool, a benchmark was created which loads a specified list of GeoTIFF files multiple times, and the benchmark is timed in order to find which Java thread pool more efficiently utilizes its processors. The benchmark was run on two Intel® Xeon® X5675 processors.

In the first case, which is shown in Figure 3, one GeoTIFF is loaded multiple times. Since the workload for each task is equivalent, all the tasks execute in an equal amount of time, and no thread spends a long time idling while another thread is executing work. The graph shows that the two thread pools have similar performance and that there is no advantage to using a ForkJoinPool.



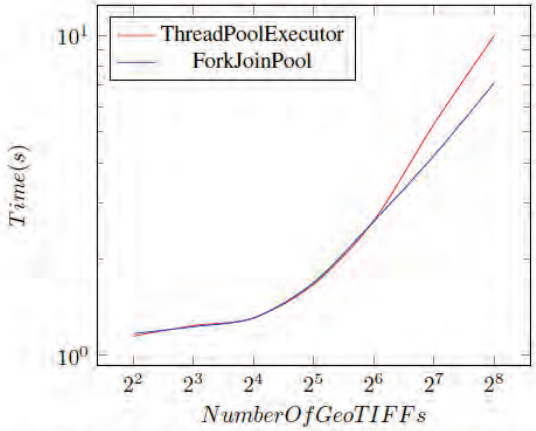
(a) Execution Time For Loading Same GeoTIFF Multiple Times Graph

GeoTIFFs	ThreadPoolExecutor(s)	ForkJoinPool(s)
1	0.62	0.62
2	0.63	0.64
4	0.67	0.66
8	0.72	0.72
16	0.97	0.98
32	1.43	1.43
64	3.07	3.06

(b) Execution Time For Loading Same GeoTIFF Multiple Times Table

Figure 3. Execution time for loading same GeoTIFF multiple times

In the second case, which is shown in Figure 4, four GeoTIFFs are loaded multiple times. The GeoTIFFs vary in both type and file size, so the workload assigned across all the threads is unbalanced. If there is an enough of an imbalance, the idle threads in the ForkJoinPool can steal work from the queue of the busier threads, which more effectively utilizes the thread pool. In unbalanced workloads, it is more effective to use ForkJoinPools due to work stealing.



(a) Execution Time For Loading Multiple GeoTIFFs Multiple Times Graph

GeoTIFFs	ThreadPoolExecutor(s)	ForkJoinPool(s)
4	1.15	1.17
8	1.24	1.23
16	1.31	1.31
32	1.67	1.69
64	2.64	2.63
128	5.30	4.22
256	9.99	7.12

(b) Execution Time For Loading Multiple GeoTIFFs Multiple Times Table

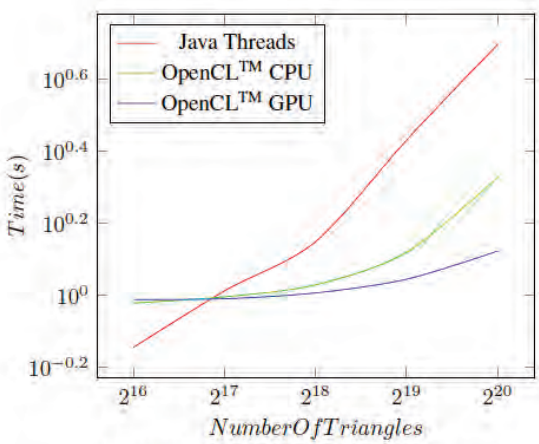
Figure 4. Execution time for loading multiple GeoTIFFs multiple times

Since the BTFC Interface needs to remain responsive and execute its tasks as quick as possible, it was decided to use ForkJoinPool since it offered the best time for the worst case scenario. When designing a user interface, it is important to always consider worst case, as one worst case can ruin the user experience.

3. Aparapi

While NASA World Wind may be capable of loading GeoTIFFs, not every file a user may want to use for BTFC Interface may be a GeoTIFF. In this case, one may have to convert a file to a GeoTIFF in order to properly render the object in NASA World Wind. This benchmark parses a triangle data file, builds a Bounding Volume Hierarchy (BVH) tree (Lauterbach, Garland, Sengupta, Luebke, and Manocha, 2009), and then does some intersection tests in order to generate height values which could be put into a GeoTIFF. Due to this benchmark's data parallel nature, Aparapi was chosen as the library of choice. Aparapi is a Java library which can convert Java byte code to OpenCL™ code, while hiding the complexities of running the OpenCL™ code and binding it to Java from the programmer. While it is still a rather new library and still has not fully matured, it has demonstrated some interesting performance benefits over native Java threading when the algorithm is data parallel. At this time, there are a few restrictions when using Aparapi, such as ensuring that there are no new arrays or structures and no passing local memory objects into a function. Fortunately, all the kernels used in this benchmark could be modified to avoid the restrictions imposed by Aparapi (Frost, 2011). The benchmark was run on two Intel® Xeon® X5675 processors and one AMD Radeon™ HD 6970.

The construction of the BVH tree can be broken down into two major steps; the first of which is the preparation step. The preparation step for building the BVH tree consists of constructing bounding boxes for each individual triangle as well as sorting the bounding boxes by dimension. Construction of the bounding boxes executes in constant parallel time while the bitonic sort occurs in $(\log N)^2$ parallel time. The execution times for the preparation kernels are shown in Figure 5. As one can see from the figure, there is a significant amount of overhead associated with Java threading; in Aparapi, each OpenCL™ thread is equivalent to a spawned Java thread, which is synchronized by cyclic barriers. By default, Aparapi may attempt to run as many as 256 threads, but it is usually best to specify a thread number no greater than the number of cores in your machine if executing the kernel in Java. Additionally, one can see from the figure that there is a relatively small increase in execution time as the number of triangles increases for the OpenCL™ execution cases. Since the slope is so small for the first few test cases, one can infer that a significant portion of the execution time is spent compiling the kernel and moving memory for these first few test cases; this is the reason why the native Java implementation is faster than the OpenCL™ implementations for a small number of triangles.



(a) Execution Time For Preparation Kernels Graph

Triangles	JTP(s)	CPU(s)	GPU(s)
65536	0.717	0.951	0.972
131072	1.028	0.989	0.977
262144	1.407	1.068	1.014
524288	2.693	1.315	1.107
1048576	4.960	2.126	1.325

(b) Execution Time For Preparation Kernels Table

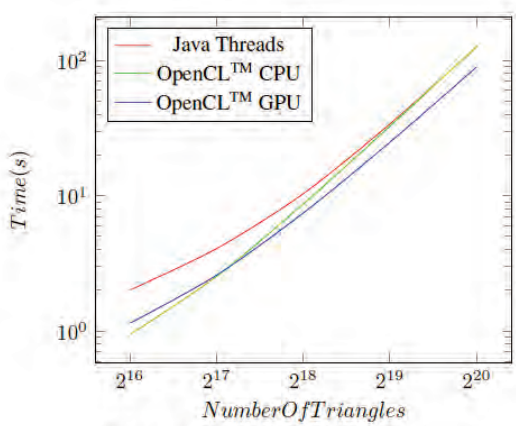
Figure 5. Execution time for preparation kernels

The second step in constructing the BVH tree is the actual building of the BVH tree. The algorithm for construction of a BVH tree, shown below in Algorithm 1, can be divided into multiple pieces of parallel portions with varying degrees of parallelism. Initially, the portion of the algorithm with a limited amount of parallelism is choosing the dimension along which to split the structures, which may adversely affect architectures designed for parallelism initially, but increasing the size of the tree should show an increased overall performance as more splits occur in parallel and the algorithm begins to spend less relative time splitting the node and more relative time reordering.

Algorithm 1 The BVH Tree Construction Algorithm

```
procedure BUILDBVHTREE
    SetUpInitialNode
    while NotFullySplit do
        for all NodesWhichExist do
            for all Dimension ∈ Dimensions do
                ChooseBestSplit
            end for
            SplitNode
            SetupNodeForDataParallelPortion
            for all Dimension ∈ Dimensions do
                if DimensionWasNotSplitDimension then
                    for all Indexes ∈ SortedIndexes do
                        CalculateNewPositionInSortedIndexes
                        Reorder
                    end for
                end if
            end for
        end for
    end while
    for all Node ∈ Nodes do
        CalculateAxisAlignedBoundingBoxes
    end for
    for Values ∈ Sorted do
        SetToSortedIndexInFirstDimension
    end for
end procedure
```

Figure 6 shows the execution time for building a BVH tree. It is interesting to note that, as the number of structures to organize gets larger, the difference in execution time between Java and OpenCL™ for the CPU decreases. This is due to the majority of the execution time being spent choosing the best split plane, which is the least parallel portion of the algorithm and has the least amount of overhead associated with the cyclic barriers.



(a) Execution Time For Build Tree Kernel Graph

Triangles	JTP(s)	CPU(s)	GPU(s)
65536	2.017	0.944	1.146
131072	4.079	2.545	2.596
262144	10.380	8.717	7.475
524288	34.065	32.765	24.936
1048576	126.286	127.443	89.760

(b) Execution Time For Build Tree Kernel Table

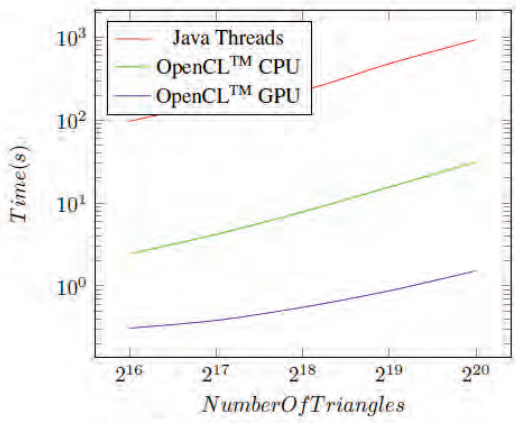
Figure 6. Execution time for build tree kernel

Lastly, this benchmark calculates which structure is the closest structure to be intersected by a particular ray for a set of rays. The algorithm for this portion is parallel across all the rays, as well as parallel across all the resulting bounding boxes in the leaf nodes of the BVH tree. The algorithm is shown in Algorithm 2 and executes in log *N* parallel time for N number of structures. It is important to note that, while testing against each leaf node is less efficient than traversing the tree, finding an intersection in this manner parallelizes across more cores and executes in less parallel time, which would mean a faster execution time if the machine can process enough threads simultaneously. Additionally, this algorithm has a better worst-case execution time than the worst-case execution time for traversing the tree.

Algorithm 2 The BVH Tree Intersection Algorithm

```
procedure INTERSECTBVHTREE
  for all Ray ∈ Rays do in Parallel
    for all Node ∈ LeafNodes do in Parallel
      if RayHitsBoundingBox then
        FindClosestStructureHit
      end if
    end for
    FindClosestNodeHit
  end for
end procedure
```

Due to the relatively limited amount of memory present in the machine, the amount of parallelism expressed in the intersection test kernel had to be restricted when the BVH tree became exceptionally large. Despite this constraint, Figure 7 clearly shows that the speed of execution is dependent on the manner in which the threads were executed as well as the amount processing power of a particular architecture. Native Java threads execute the task an order-of-magnitude slower than OpenCL™ threads, which is expected since Java has been shown to be less efficient at utilizing the hardware available than C (Taylor, Cook, Kavi, and Levine, 1998; Hundt, 2011). Additionally, a GPU is a throughput-orientated architecture which is designed to maximized overall performance by hiding latency through context switching at the expense of additional latency (Fatahalian and Houston, 2008; Garland and Kirk, 2010). Through hiding latency and maximizing performance of its many simpler cores, GPUs are shown to have more processing power than CPUs (Hawick, Leist, and Playne, 2009).



(a) Execution Time For Intersection Kernel Graph

Triangles	JTP(s)	CPU(s)	GPU(s)
65536	96.618	2.441	0.310
131072	162.488	4.192	0.385
262144	229.051	7.843	0.554
524288	481.334	15.640	0.876
1048576	933.093	31.352	1.529

(b) Execution Time For Intersection Kernel Table

Figure 7. Execution time for intersection kernel

Aparapi has demonstrated itself as a high-performance library which easily integrates with the BTFC Interface. The additional performance from executing data parallel algorithms efficiently across multiple processors as well as on processors which native Java cannot interact with enables the BTFC Interface to execute compute-intensive tasks rapidly without hanging.

4. Conclusion

When programming for many core and heterogeneous architectures, it is critical to consider how effectively the code utilizes all the processing cores. In the case of the BTFC Interface, it is critical to minimize execution time and keep the interface responsive, since the interface is the only way for the program and the Warfighter to interact. By utilizing the ForkJoinPool introduced in Java 7, it is possible to maximize CPU utilization and minimize execution time of task- parallel workloads. Additionally, it is possible to efficiently assign work to a GPU or maximize the performance of a CPU for data parallel workloads by utilizing Aparapi. Through proper usage of these libraries, it is possible to not only minimize program execution time, but also maximize the efficiency of the user's time, which is ultimately the goal of any software system. In conclusion, incorporating these libraries into the BTFC Interface assists in achieving the overall goal of minimizing system time where the server, the interface, and the Warfighter are all part of the complete system.

Acknowledgements

Support for this work is provided in part by the US Army High Performance Computing Research Center (AHPCRC).

References

Batcher, K.E., "Sorting Networks and Their Applications." *AFIPS Spring Joint Computing Conference*, pp. 307–314, 1968.

Bohr, M., "A 30 Year Retrospective on Dennard's MOSFET Scaling Paper." *Solid-State Circuits Newsletter*, pp. 11–13, 2007.

Fatahalian, K. and M. Houston, "A Closer Look at GPUs." *Communications of the ACM*, pp. 50–57, 2008.

Frost, G., "Aparapi Java Kernel Guidelines." *Google Code*, 2011.

Garland, M. and D.B. Kirk, "Understanding Throughput-Oriented Architectures." *Communications of the ACM*, pp. 58–66, 2010.

Hawick, K.A., A. Leist, and D.P. Playne, "Mixing Multi-Core CPUs and GPUs for Scientific Simulation Software." *Technical Report CSTN-091*, Massey University, 2009.

Hillis, W.D. and G.L. Steele Jr., "DATA PARALLEL ALGORITHMS." *Communications of the ACM*, pp. 1170–1183, 1986.

Hundt, R., "Loop Recognition in C++/Java/Go/Scala." *Proceedings of Scala Days 2011*, 2011.

Joshi, S., "Leveraging Aparapi to Help Improve Financial Java Application Performance." *AMD Developer Central*, 2012.

Kipfer, P. and R. Westermann, "Improved GPU Sorting." *GPU Gems 2*, pp. 733–746, 2005.

Lauterbach, C., M. Garland, S. Sengupta, D.P. Luebke, and D. Manocha, "Fast BVH Construction on GPUs." *Computer Graphics Forum*, pp. 375–384, 2009.

Lea, D., "A Java Fork/Join Framework." *Proceedings of the ACM 2000 conference on Java Grande*, pp. 36–43, 2000.

Rajan, H., S.M. Kautz, and W. Rowcliffe, "Concurrency by Modularity: Design Patterns, a Case in Point." *Proceedings of the ACM International Conference on Object Oriented Programming, Systems, Languages, and Applications*, pp.790–805, 2010.

Taylor, S.R., D.J. Cook, K.M. Kavi, and D. Levine, "A Comparison of Multithreading Implementations." *Yale Multithreaded Programming Workshop*, 1998.

Wald, I., and V. Havran, "On building fast kd-Trees for Ray Tracing, and on doing that in O(N log N)." *Proceedings of the 2006 IEEE Symposium on Interactive Ray Tracing*, pp. 61–70, 2006.

Zhou, K., Q. Hou, R. Wang, and B. Guo, "Real-Time KD-Tree Construction on Graphics Hardware." *ACM Transactions on Graphics*, pp. 126–137, 2008.

Multiple Simultaneous Domain Decomposition of Interdisciplinary Mobile Network Simulations via Distributed Shared Memory

Jerry Clarke, Kenneth Renard, and Kenneth W. Leiter
US Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
{jerry.clarke, kenneth.renard}@us.army.mil, kenneth.w.leiter2.civ@mail.mil

Abstract

Certain interdisciplinary simulations could benefit significantly from using multiple domain decomposition schemes to increase efficiency and functionality. For example, in mobile network modeling optimal parallel layout for RF propagation primarily involves spatial decomposition, while the discrete event network simulation decomposes along network node boundaries. Traditionally, this would require each parallel simulation to be cognizant of the data decomposition of another simulation in order to exchange data. However, by utilizing a novel distributed shared memory facility, originally designed and implemented at the US Army Research Laboratory, we are able to efficiently share computed quantities among multiple simulators while maintaining multiple parallel data decompositions for each simulation component. In this paper, we describe the method of coupling separate node movement, RF propagation, and network simulators with this novel concept.

1. Introduction

The Military is becoming a network-centric force, but it is difficult to design and evaluate mobile ad hoc networks with sufficient fidelity and scale. Brigade scale simulations may contain tens or hundreds of thousands of network devices and require massively-parallel systems and scalable software to complete. To accomplish this complex modeling task, it is advantageous to decompose the task into its various components: movement, RF propagation, and network communication. Thus, we are able to concentrate efforts on each component individually, with the best available technology, to arrive at a new composite capability.

Movement of military platforms and personnel is a complex topic all to itself. There are complex simulation engines to accurately model realistic movement based on a variety of factors. For our purposes; however, we can put aside these complexities for the moment and concern ourselves only with the basic functionality. From a computational science standpoint, it is sufficient to understand that this component establishes the positions of platforms and positions over time.

As platforms and personnel move over a complex terrain, or in an urban environment, the RF propagation is significantly impacted. This propagation and interference determines the ability of any two nodes to communicate. As with movement, this is also a complex area of study that can be accomplished with various levels of fidelity at the expense of additional computation.

Once potential communication links are established, the information flow in the network can be modeled either by executing actual software applications or by abstracting their communications behavior. How these communications are routed in an ad hoc network as well as the speed and latency determines the performance, and ultimately the usefulness of the entire system.

For each of these three components to be accomplished, they must decompose the computational domain across many processors. The optimal decomposition for one component is not necessarily optimal for another component. For example, RF propagation decomposition is accomplished spatially, while network decomposition is along network boundaries. In this effort, we have developed a system to allow individual parallel simulation components to be combined without the need for integrating them into a single software entity. This allows for swapping of components, with various features, as necessary. Most importantly, each individual component need not be cognizant of how other components have decomposed the problem in order to communicate information in parallel simulation.

2. DSM Background

The issue of coupling parallel simulations with different decompositions is not unique to network modeling. For example, this need has arisen in areas of blast structure interaction[1] and in-situ visualization[2]. In blast structure interaction, the blast is modeled on a structured mesh that refines itself adaptively while the unstructured mesh of the Lagrangian finite element code deforms in response to the blast pressure loading. For visualization, various parallel codes must be accessed at runtime in order to render surfaces, cutting planes and other graphical objects using parallel visualization and rendering techniques. For these tasks we have built upon the original version of Network Distributed Global Memory (NDGM)[3] to develop an efficient distributed shared memory system that works well under Message Passing Interface (MPI). NDGM was a client-server based system that implemented a byte-addressable buffer spanning multiple servers. Put/Get requests to a virtual address resulted in data exchange across a TCP network socket to the appropriate server. This concept was rewritten to work over MPI and incorporated into the eXtensible Data Model and Format (Xdmf)[4]. In addition to the Put/Get interface, a virtual file driver for HDF5 was added. This allowed access to the distributed buffer via the HDF5 API in addition to all of the metadata facilities available in HDF5. Finally, in collaboration with the Swiss National Supercomputing Centre, the system was reworked, with improvements particularly to the HDF5 virtual file driver.

3. DSM Details

As mentioned previously, the DSM uses a client-server model. One or more servers provide a byte-addressable buffer with a Put/Get interface. The amount of memory provided by each server is configurable, but typically divided equally across servers. Via a capability implemented via pthreads, a client and server can operate on the same MPI rank. In this situation, a client request to an address range on the local sever is accomplished via memcpy() as opposed to message passing. Figure 1 illustrates this buffer.

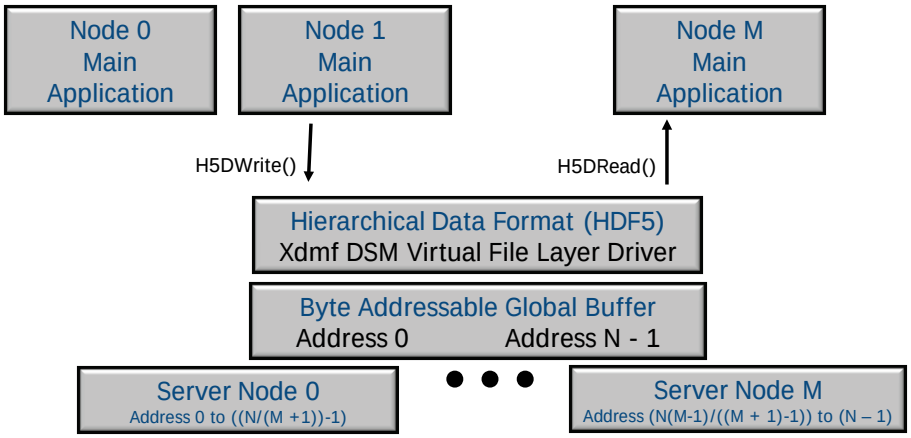


Figure 1. Non-threaded DSM with HDF5 Interface

The byte-addressable buffer can be used in any fashion by the client application. Typically, for data exchange purposes, it is necessary to share information about the size, shape and number type of the various data arrays. The HDF5 virtual file driver provides this functionality. HDF5 metadata, traditionally intended for disk files, is written to the DSM buffer along with the actual data values. This HDF5 metadata must be synchronized across all participants so, in practice, it is usually convenient for one client to open the HDF5 file in DSM and create empty datasets of the correct shape and number type for the application. Then, all participants can open the HDF5 datasets with read/write access and transfer values asynchronously.

4. Network Simulation

The US Army Research Laboratory (ARL) is working to address the challenges of high-fidelity modeling of MANETs by exploiting the high performance computing (HPC) capabilities of the Department of Defense (DoD). Commercial simulation tools such as OPNet and QualNet were evaluated on HPC platforms. While these commercial tools have a wide

range of models for specific military waveforms of interest, they did not demonstrate adequate scalability on HPC platforms. An open-source alternative for network simulation was sought that could support parallelism on HPC platforms, provide standard models of sufficient fidelity, and support custom development of models for military waveforms, applications, and high-fidelity RF propagation models. The target size of networks to be modeled is on the order of an entire division and larger (10,000+ network nodes). This capability will provide DoD researchers and planners with the ability to do analysis for acquisition, network and protocol research, and deployment optimization studies. Currently, our open-source simulator of choice is NS-3 which is a parallel C++- based network simulator to which we actively contribute.

The NS-3 Distributed Scheduler was introduced in version 3.8 in May 2010. This scheduler implements a conservative parallel discrete event simulator where networks can be broken into "federates" across specialized point-to-point links. Communications between federates is via MPI, where packets are serialized and delivered to be scheduled as "receive" events on remote federates. Simulator timing is carefully controlled among federates such that causality is preserved given a minimum network latency (look-ahead) between the partitioned sections of the simulated network. Each federate is allowed to process local events with perfect parallelism up to a specific time-step that is agreed upon by all federates. Each federate then computes its next minimum time-step where events will not be generated for any remote federate. The minimum of these collected values from all federates is granted time for each federate to process events in parallel.

NS-3 has built-in mobility models which are useful for many purposes. For our efforts; however, we wish to simulate military formations reacting as realistically as possible. This will require its own simulation. While the eventual choice for a simulator may be something more complex, like OneSAF, we are currently using the open-source general-purpose discrete event simulator SimPy. We simulate rifle squads moving in formation, attempting to maintain appropriate spacing while the point slightly varies his speed with some randomness. Each MPI rank in the simulation simulates a single squad and synchronizes simulation time via a simplified Run-Time Infrastructure (RTI) we developed for this project.

ARL has done a significant amount of research in high-fidelity RF propagation for network simulations. Implementations of Irregular Terrain Model (ITM) and ray-tracing algorithms (for urban terrain) have been developed to take advantage of GPUs in order to run fast enough to be used in real-time simulations. The details of this important work are beyond the scope of this paper, but we use an implementation of these methods for the RF calculations.

5. Simulation Setup and Execution

Figure 2 shows the overall topology of the simulation. Execution setup begins by running python in parallel that uses Mpi4Py to set up all of the necessary communicators and groups that will be used by the various components of the simulation. Once the communicators have been created, the nodes chosen as DSM servers begin to handle requests, and one of the remaining simulation nodes creates the necessary HDF5 datasets via the virtual file driver interface. With setup complete, one node starts handling RTI requests, while other groups of nodes begin executing the various components of the simulation.

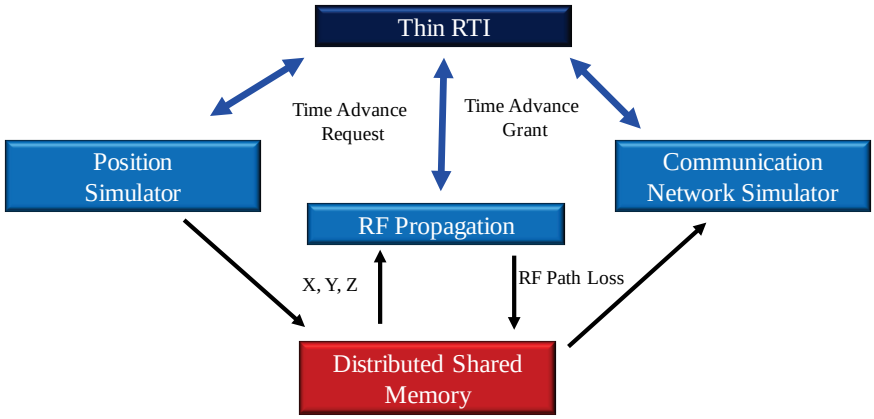


Figure 2. Simulation diagram

In NS-3, mobility and propagation models have been developed that interface with this architecture by referencing DSM. When a data frame is transmitted via radio, the RF propagation loss [computed by an external process] is simply obtained as if it were a memory reference. The same goes for mobility computations. Typically, the mobility information is only used in NS-3 to calculate the RF propagation, but it can be used in other forms such as at the application layer. For

example, we often use a Situational Awareness application that reports positions via simulated network. Each MPI rank that executes part of the NS-3 scenario creates a connection to the shared memory. Wireless and mobile nodes are then configured with propagation and mobility models that reference this connection. The only computation involved by the network simulator is then calculating the offset into DSM dataset of the desired result. While there will be a delay introduced by the computation and distribution of these values, we believe that it will be negligible with respect to the fidelity of the simulation. This will be measured and verified in future studies. Methods for reducing this latency are feasible if it is determined that the latency does have an impact on results.

This work has been prototyped and run a small-scale thus far, using up to 8 cores total for DSM processing, mobility control, RF propagation, and NS-3 execution. Figure 3 shows the simulation running as it updates positions for two rifle squads in parallel.

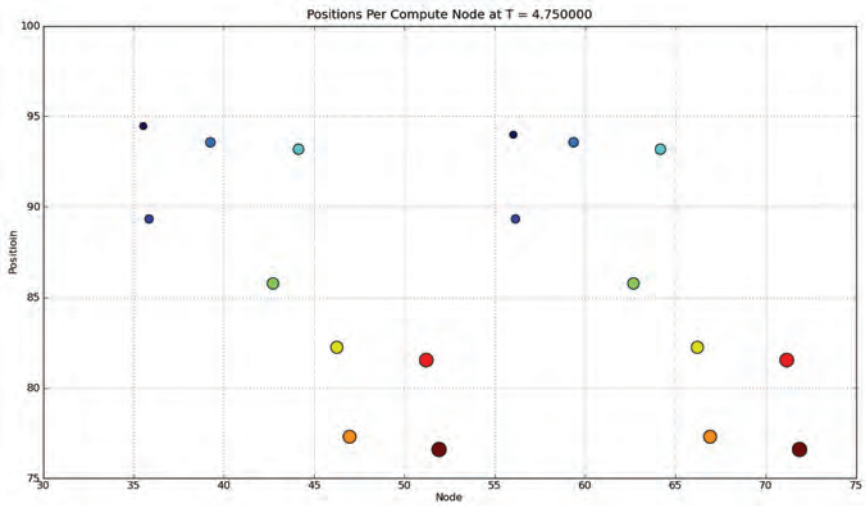


Figure 3. Two Rifle squad simulation

6. Conclusion

We have established a method to couple multiple parts of an inter-disciplinary simulation via Distributed Shared Memory that allows orthogonal decomposition of each computational problem. This can lead to better performance of each component with a small overhead for communication and coupling. The immediate gain for this work is the ability to perform co-simulation; using different simulation applications, each with their respective specialties (e.g., force modeling and movements, RF propagation, network modeling). This work has the potential to improve performance through the use of highly-optimized codes, as well as provide scaling of a large set of simulations into a coupled model. We plan to extend this work to integrate additional codes such as GPU-accelerated RF propagation and OneSAF forces modeling.

References

1. Clarke, J.A. and R.R. Namburu, "A Generalized Method for One-Way Coupling of CTH and Lagrangian Finite-Element Codes With Complex Structures Using the Interdisciplinary Computing Environment." *Tech. Rep. ARL-TN-230*, US Army Research Laboratory, Aberdeen Proving Ground, MD, November 2004.

2. Soumagne, J., J. Biddiscombe, and J. Clarke, "An HDF5 MPI virtual file driver for parallel in-situ post-processing." *EuroMPI'10 Proceedings of the 17th European MPI Users' Group Meeting*, pp. 62–71

3. Clarke, J.A., "Emulating Shared Memory to Simplify Distributed-Memory Programming." *IEEE Comput. Sci. Eng.*, vol. 4, no. 1, pp.55–62, 1997.

4. Clarke, J.A. and E.R. Mark, "Enhancements to the eXtensible Data Model and Format (XDMF)." *HPCMP-UGC '07: Proceedings of the 2007 DoD High Performance Computing Modernization Program Users Group Conference*, Washington, DC, IEEE Computer Society, pp. 322–327, 2007.

5. Weingartner, E., H. vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators." *Proceedings of the IEEE International Conference on Communications 2009 (ICC 2009)*, Dresden, Germany, 2009.

6. Renard, K., C. Peri, and J. Clarke, "A Performance and Scalability Evaluation of the NS-3 Distributed Scheduler." *Workshop on NS-3 (WNS3 2012)*, 2012

7. *Hierarchical Data Format (HDF5)*, http://hdf.ncsa.uiuc.edu/.

8. Lofstead, J.F., S. Klasky, K. Schwan, N. Podhorszki, and C. Jin, "Flexible IO and integration for scientific codes through the adaptable IO system (ADIOS)." *CLADE '08: Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments*, New York, NY, USA, ACM, pp. 15–24, 2008.

9. Clarke, J.A. and R.R. Namburu, "Simplifying Partitioning Complexities by Using a Common Data Hub." *Tech. Rep. ARL-TR-2799*, US Army Research Laboratory, Aberdeen Proving Ground, MD, August 2002.

10. Chilan, C.M., M. Yang, A. Cheng, and L. Arber, "Parallel I/O Performance Study with HDF5, a Scientific Data Package." National Center for Supercomputing Applications, University of Illinois, Urbana- Champaign, Tech. Rep., 2006.

11. Yang, M. and Q. Koziol, "Using collective IO inside a high performance IO software package - HDF5." National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign, Tech. Rep., 2006.

Technology Transfer to Army Fielded Combat Robots

Peter Raeth and Juan Carlos Chaves
Dynamics Research Corporation, Fairborn, OH
{praeth, jchaves}@drc.com

Sean M. McGhee
STG, Inc., Reston, VA
sean.m.mcghee.ctr@mail.mil

Abstract

The US Army Research Laboratory (ARL) develops technology for future and more capable unmanned systems. These improvements are enabled through advancements in intelligent control, machine perception, human-machine interaction, mechanics, and propulsion. This interdisciplinary research crosses the boundaries of land, sea, and air, and addresses a wide variety of needs for military unmanned systems. These range in size from large Future Combat Systems vehicles to micro-scale soldier-carried robotic platforms. The Army, industry, and academia combine resources to upgrade fielded combat robots under the ARL-managed Robotics Collaborative Technology Alliance. The use of robots on the battlefield is among the most exciting advancements of technology in the Department of Defense (DoD). The authors provided support to the ARL Human Research and Engineering Directorate (HRED) robotics team by transferring technology that originates in university research. The effort inserted machine vision into robots used in urban combat. This technology transition benefits from the very same codes being concurrently exploited in Windows-based robots and High Performance Computing Modernization Program (HPCMP) high performance computing environments for the benefit of multiple applications. This project was carried out in two months, which compares well to the Army's normal expectation of eight years to bring new technology to the battlefield.

1. Introduction

A process was established to examine results from university, contractor, and government researchers. Results that appear promising are built into stand-alone libraries for integration with application-specific drivers. In this project, we are working with MATLAB, C/C++, and OpenCV. However, there is the potential for the integration of other libraries and languages. This process is illustrated in Figure 1.

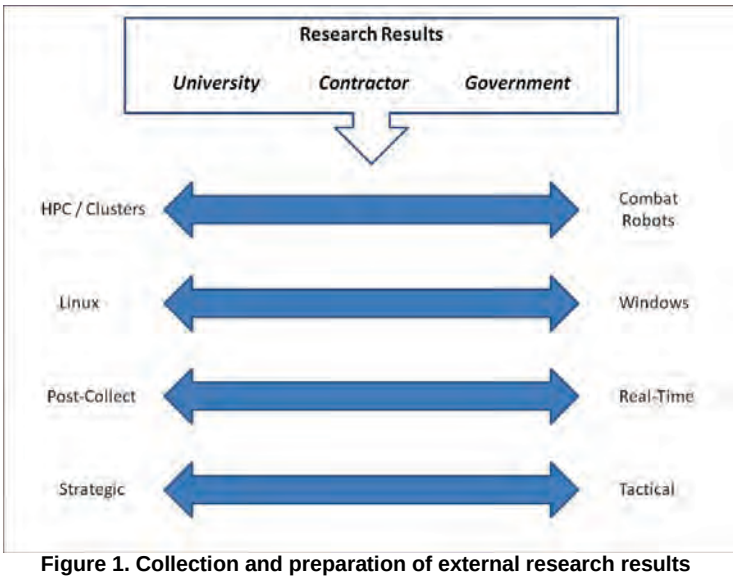


Figure 1. Collection and preparation of external research results

The use of all collected material is used as background to satisfied mission requirements (Figure 2). Taken together, all components drive a technology transition process that significantly reduces time to insertion. In this case, only two months were required to insert university research into a fielded combat robot. That length of time was due to the considerable learning that was necessary. For the robot in question, only two weeks would now be needed for upgrades and perhaps one month for a new insertion. This compares well with the Army's expected eight years to bring new technology to fielded systems.

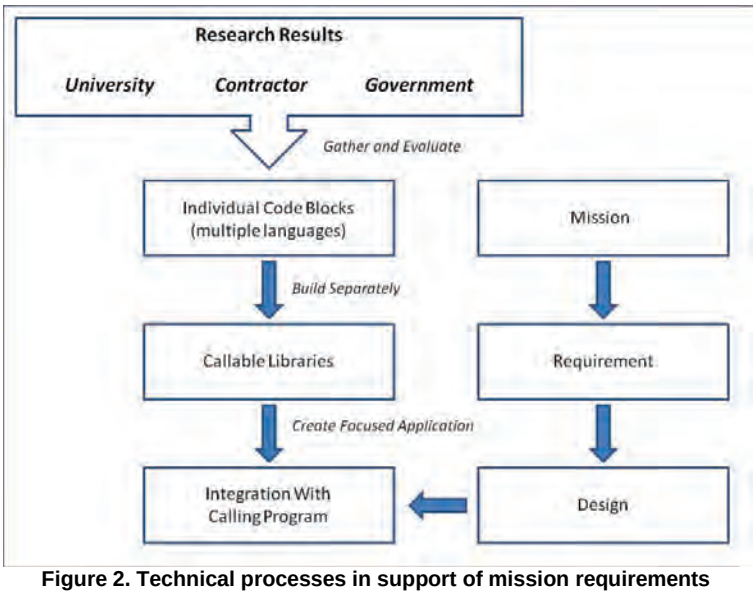


Figure 2. Technical processes in support of mission requirements

2. Application to Combat Robots

Two robots were involved, the Pioneer DX laboratory robot (Figure 3) and a combat-deployable PackBot produced by iRobot (Figure 4). This Army project provides continuous upgrades in combat capability and serves to lessen the risk to personnel in highly-dangerous situations. Besides Department of Defense (DoD) applications, the PackBot is used for plant security, search and rescue, as well as chemical and nuclear accident recovery.

Figure 3. Pioneer DX prototype robot

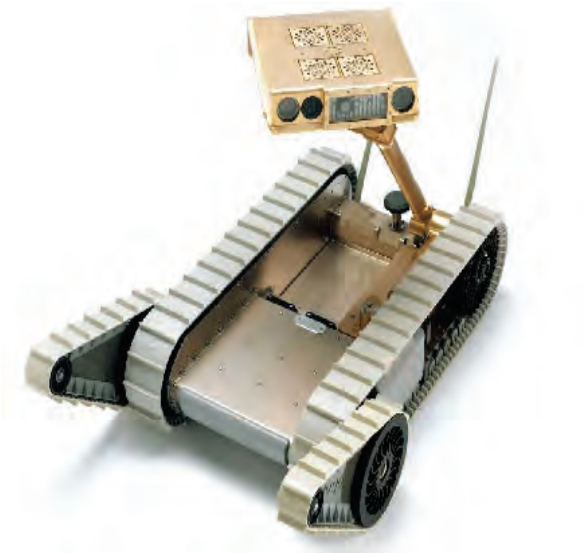


Figure 4. PackBot configured for sniper detection

3. Significance to ARL HRED Mission

The following benefits directly accrued to target the US Army Research Laboratory (ARL) HRED project:

- Transitioning Linux-based image processing capabilities to Windows-based combat robots.
- Minimizing barriers, costs, risks, and timelines to affect such transitions.
- Enabling transition of Windows-based real-time combat capability to Linux-based post-collect processing on cluster platforms, and vice-versa.
- Ready integration of research results to meet deployment requirements and to overcome emerging threats.
- Merging of disparate computer languages and programs to create new capability in response to evolving requirements and missions.

4. Significance to DoD Mission

Unmanned (autonomous and non-autonomous) robotic systems are highly desired by combatant commanders (COCOMs) for the many roles these systems can fulfill. Tasks such as mine detection; signals intelligence; precision target designation; chemical, biological, radiological, nuclear, explosive (CBRNE) reconnaissance; and communications and data relay rank high among the COCOMs' interests. These unmanned capabilities help reduce complexity and time-lag in the sensor component of the sensor-to-shooter chain for prosecuting actionable intelligence. Unmanned systems are changing the conduct of military operations in the Global War on Terrorism by providing unrelenting pursuit combined with the elimination of threats to friendly forces.

References

iRobot's Packbot public information: http://www.irobot.com/us/search_results.aspx?q=packbot.

Pioneer DX robots: http://www.mobilerobots.com/researchrobots/pioneerp3dx.aspx.

Unsteady Aerodynamic Simulations of Maneuvering Projectiles

Jubaraj Sahu, James DeSpirito, and Karen R. Heavey
US Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
jubaraj.sahu.civ@mail.mil

Abstract

This paper describes the recent progress made in the development and/or application of advanced computational capabilities for prediction of unsteady aerodynamics and flight dynamics of projectiles with and without control maneuvers. These advanced computational capabilities include time-accurate Navier-Stokes computational fluid dynamics (CFD) and coupled CFD/rigid-body dynamics (RBD)/flight control system (FCS) techniques. Numerical computations have been performed using an advanced scalable unstructured flow-solver for both lateral jet control and canard control maneuvers. Computed results have been obtained for a basic finned-stabilized projectile to investigate the jet interaction effects from a jet issuing into a supersonic cross-flow using steady rolling and pulsed-jet simulations. These results show that unsteady jet effects were small, and steady-state results can be used to determine the projectile response for these jets and cross-flow conditions. For simulation of guided control maneuvers, development of a coupled CFD/RBD/FCS capability has been continued. The coupled CFD/RBD/FCS capability has been applied to a canard-controlled projectile and the effect of canard deflection angle on the aerodynamics and flight dynamics of this projectile has been investigated using a roll-control maneuver and a cross-range control maneuver. Computed results obtained from these maneuvers show the time-dependent response of the projectile for given FCS designs.

1. Introduction

As part of a Department of Defense (DoD) high performance computing (HPC) Grand Challenge Project, the US Army Research Laboratory (ARL) has recently focused on the development and application of state-of-the-art computational fluid dynamics (CFD), and coupled CFD and rigid-body dynamics (RBD) techniques for large-scale simulations of projectile aerodynamics and flight dynamics with and without flow control maneuvers.[1–8] Advanced CFD techniques are being exploited on modern HPC machines to provide accurate aerodynamics required for design and analysis of affordable precision munitions.

For maneuvering munitions, the effect of many weapon control mechanisms such as canards, reaction jets, deployable pins, pulsed flaps, and micro-jets on aerodynamics and flight dynamics is critical to overall guided flight performance.[9–15] The control devices are used to provide control forces needed for desired maneuvers. The idea is to determine if these control devices can provide the desired control authority for course correction for munitions. Many of these mechanisms fall outside the range of conventional aerodynamic control and accurate well-validated tools for prediction of aerodynamic loads are needed. These control mechanisms result in highly complex, unsteady flow interactions and their accurate modeling during guided flight with active control is a major challenge. To meet this challenge, a set of well-validated advanced CFD and CFD coupled with flight dynamics and flight control system (FCS) tools is needed for prediction of aerodynamic loads and unsteady aerodynamics associated with conventional and new aerodynamic control technologies for maneuvering precision munitions.

A lot of progress has been made recently with the computational technology involving CFD and RBD. These advances in CFD technologies have the potential for greatly reducing the design costs and providing a more detailed understanding of the complex aerodynamics than the understanding achieved through experiments and actual test firings. Coupling of CFD and RBD has led to the notion of physics-based virtual fly-out simulations of projectile free-flight motion in a time-accurate manner.[1,2] In the virtual fly-out method, coupled CFD/RBD technique is used in conjunction with a grid motion capability to fly a projectile on the supercomputer similar to what happens with the actual free-flight of the projectile inside an aerodynamics experimental facility. Aerodynamic forces and moments are computed with the CFD solver, while the free-flight motion of the projectile is computed by integrating the RBD equations of motion. This approach was first

demonstrated on a finned projectile at a supersonic speed, and validated with excellent agreement between the coupled CFD/RBD results and spark range measurements. The coupled CFD/RBD virtual fly-out technique has been successfully extended to subsonic[9] and transonic speed regime[5] and has been demonstrated on spinning projectiles. In some cases, validation of computed results from the virtual fly-out simulations was accomplished with comparison to free-flight test data. For simulation of control maneuvers, our recent focus has been in the development of an initial CFD/RBD/flight control system (FCS)[8] with the coupling of FCS into the coupled CFD/RBD procedure. Currently, research efforts are underway to further develop and validate this coupled CFD/RBD/FCS capability for prediction of unsteady aerodynamics and flight dynamics associated with conventional and new aerodynamic control technologies for maneuvering precision munitions.

This paper describes the recent progress made in the development and application of the advanced time-accurate CFD and CFD/RBD/FCS methods to projectile aerodynamics. The following sections describe the solution technique and coupled CFD/RBD/FCS procedure and the computed results obtained for a number of projectiles without and with control maneuvers.

2. Computational Methodology

2.1 CFD Flow-Solver

The complete set of three-dimensional (3D) time-dependent Reynolds-averaged Navier-Stokes (RANS) equations[16] is solved in a time-accurate manner for simulations of unsteady flow-fields associated with both spin-stabilized and finned projectiles during flight. The 3D time-dependent RANS equations are solved using the finite volume method[16,17]:

$$\frac{\partial}{\partial t}\int_V \mathbf{W}dV + \oint\left[\mathbf{F}-\mathbf{G}\right]\cdot dA = \int_V \mathbf{H}dV \tag{1}$$

where **W** is the vector of conservative variables, **F** and **G** are the inviscid and viscous flux vectors, respectively, **H** is the vector of source terms, V is the cell volume, and A is the surface area of the cell face. Second-order discretization was used for the flow variables and the turbulent viscosity equations. Two-equation k-ε[18] turbulence models were used for the computation of turbulent flows. Dual time-stepping was used to achieve the desired time-accuracy. The virtual fly-out technique using coupled CFD/RBD/FCS procedures that is of interest here also uses dual time-stepping.

2.2 Coupled CFD/RBD/FCS Technique

The coupled CFD/RBD/FCS[8] technique used in the Challenge Project coupled all three disciplines (CFD, RBD, and FCS). The BoomFCS[10] code which was specifically designed to predict the atmospheric flight mechanics of smart weapon systems readily provides the needed FCS. In BoomFCS[10], control of a projectile can be achieved in many different ways, e.g., with canards and pulse jets during the flight. Not only are many different control mechanisms used to control projectiles, many different strategies are employed to guide and control a projectile. The control system strategy is conveyed through the control law. The control law stipulates a set of operations that are performed on sensor data to determine how the controls should be changed in flight.

For simulations of controlled flights of projectiles, an interface is used for easy transfer of both the RBD state variables, and the control system variables of interest between the flow-solver[6–17] and BoomFCS[10] code. The aerodynamic forces and moments are computed at every time-step in the CFD part and transferred to BoomFCS[10] which does both RBD and FCS simulations. The FCS simulation provides as output the flight control variables based on a given FCS design. The FCS design allows for modeling of both controlled and prescribed motions. The output of RBD state and the control variables are transferred to the CFD flow-solver, which then computes the aerodynamic forces and moments subject to these RBD state and control variables. For example, for a canard-controlled projectile, the output of the FCS variables would be the canard deflection angles. As canards are deflected with the FCS-generated deflection angles, the flow-solver must take into account the canard motion. Similarly, a set of different FCS variables could be used for a jet-controlled maneuver. As this FCS coupling matures, additional control variables will be used for a variety of controlled flights.

Typically, we begin with a computation performed in "steady-state mode" with the grid velocities prescribed to account only for the translational motion component of the complete set of initial conditions to be prescribed. At this stage, we also impose the angular orientations from the initial conditions, and spin-rate is added. Computations are performed with the spin in a time-accurate mode for a desired number of spin cycles. Converged solution from this step provides the initial condition for the next step, where a completely-coupled CFD/RBD/FCS computation is performed in time-accurate mode. Here, a complete set of initial conditions includes all translational and rotational velocity components and accounts for initial

position and angular orientations. In addition, FCS is activated at the beginning of this coupled run for physics-based guided virtual fly-out simulations of munitions.

2.3 Parallel Computational Aspects

The flow-solver used in the Challenge Project can be run in parallel on a wide variety of hardware platforms and parallel computers, including those from Silicon Graphics, Cray, and PC workstation clusters. The numerical method used is scalable on newer computers such as the SGI Altrix and Cray XE-6 cluster. Inter-CPU communications are included at the fine-grid level as well as all the multi-grid levels. This strategy ensures high degree of robustness in the flow-solver and is independent of the number of CPUs being employed.

The mesh files needed in the computations are generated only once. The same mesh files can be used for any multi-CPU runs. The same is true of the restart files. For multi-CPU runs, a domain decomposition file is needed; again this file is obtained only once at the beginning of the run. The domain decomposition file defines the association between cell numbers and CPUs. Domain decomposition was done using the METIS tool developed at the University of Minnesota.

All numerical simulations were performed on the 10,752-core SGI Altix ICE 820 System (Harold) at the ARL Defense Supercomputing Resource Center (DSRC), and the 43,172-core Cray XE-6 system (Raptor) at the US Air Force Research Laboratory (AFRL) DSRC. Typically, the jobs were run with either 128 or 256 processors on these HPC machines. Very good performance of the flow-solver is achieved on many modern platforms. The parallel performance of the flow-solver on both parallel machines (Harold and Raptor) is obtained using a number of benchmark runs and is shown in Figures 1 and 2. Four different mesh sizes were used in the benchmark runs for scalability of the code. It should be noted that these benchmark runs were done on these systems in a real multi-user multi-job environment and not on a dedicated machine. Figure 1 shows the parallel performance of the code on the Cray XE-6 system (2.4 GHz). Very good parallel performance has been observed up to 256 processors for three grids (10, 20, and 30M); performance drops some with an increase in the number of processors. Parallel performance results obtained with these different meshes were similar. With the 60M grid, more work is available to each processor, and in this case one can see a perfect linear speed-up for all processors up to 1,024. The parallel performance of the code on Harold is shown in Figure 2 for a 30M grid; again, good performance is obtained on this machine for number of processors up to 256 and less reasonable speedup up to 1,024 processors compared to Raptor.

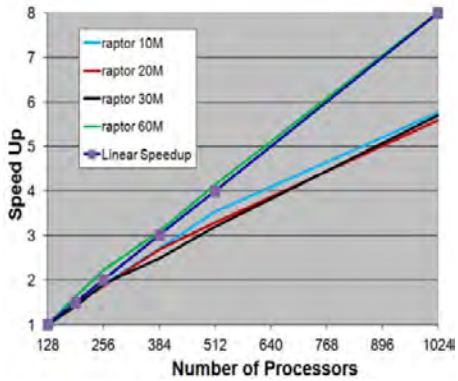


Figure 1. Parallel speedups on a Cray XE-6 (Raptor)

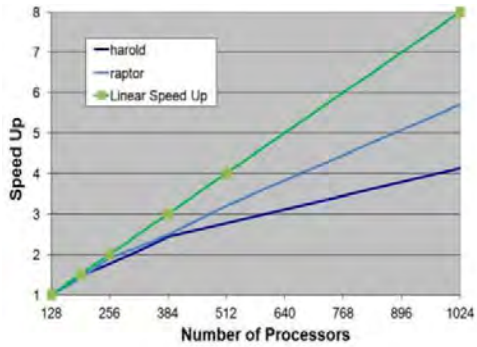


Figure 2. Parallel Speedup Comparison on a 30-Million grid

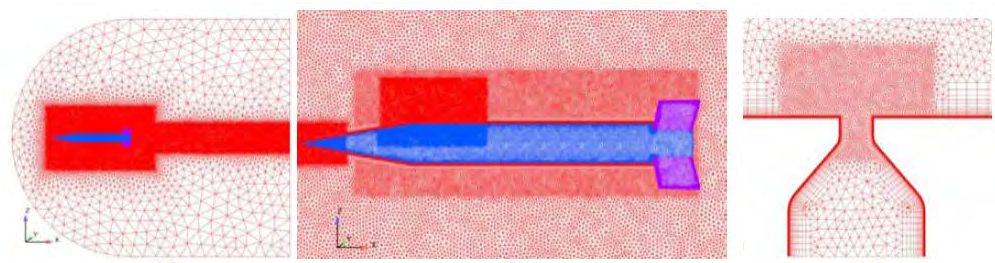
3. Results

The time-accurate CFD and coupled virtual fly-out approaches were used to generate aerodynamics and flight dynamics of a number of projectiles. Results obtained for some of these cases are presented next.

3.1 Time-Accurate Computations of Control Jet Interaction Effects for a Finned Projectile

The primary purpose of a lateral reaction jet[12,13] is to generate a force or moment to provide attitude or roll control of a flight vehicle. Lateral reaction jets have several advantages over conventional aerodynamic controls such as canards or fins; e.g., increased maneuver authority when operating in low dynamic pressure (low velocity or high altitude), small-time delay for actuation effect, compact design, and the external aerodynamics of the flight vehicle is unaffected except during the actuation period of the jet. The main disadvantage in atmospheric flight is the effect of the jet interaction (JI) flow-field—an interference flow between the jet plume and the flow over the vehicle—on the resulting control forces and moments.

Accurate prediction of the JI effects is important for predicting the overall forces and moments imparted to the projectile, as the presence of the lateral jet affects the entire flow-field. The flow disturbances due to the JI alter the forces and moments that would otherwise be expected to be produced from the jet thrust alone. The overall effect is that both the control force and moment produced by the lateral jet may be augmented or attenuated due to JI effects. Previous studies[14,15] investigated several parameters, such as pressure ratio, throat area to nozzle exit area ratio, jet location, and projectile angle-of-attack (α) on the JI effects on a generic finned-projectile (Figure 3) in a supersonic cross-flow. Recent work extends those results to determine the effects of both steady-state roll and a transient, pulsed-jet on the JI force and moment predictions.



(a) (b) (c)

Figure 3. Mesh used for ANF simulations; (a) full-domain, (b) density boxes near projectile, and (c) jet nozzle and plenum

The basic Army-Navy Finner (ANF) projectile (see Figure 3) used in this study is a basic cone-cylinder design, 10 cal. long with a 2.84 cal. (1 cal.=30 mm) conical nose. There are four uncanted, 1 cal. square planform fins mounted flush with the base of the projectile. The c.g. is located 5.5 calibers from the nose of the projectile. The previous steady-state simulations[14,15] compared the JI effects with the jet located at seven different axial locations along the uppersurface. The current work investigates the effects of steady-state rotation and transient, pulsed-jet injection at one axial location, 4.25 cal. (127.5 mm) from the nose of the projectile. The extent of the computational domain and the mesh (19.5M cells) used are shown in Figure 3. The computational domain was meshed with MIME from Metacomp Technologies and consisted of tetrahedral cells and triangular prism layers projected from the solid-wall surfaces. A full three-dimensional mesh was used, with density boxes to refine the mesh in expected regions of high-gradients, especially in the wake of the jet.

The inlet to the nozzle plenum was modeled as a subsonic reservoir boundary inflow with a specified total temperature and pressure. A time-varying version of that boundary condition was used in the pulsed-jet simulations to control the jet thrust. The free-stream conditions were based on standard sea-level conditions: a static pressure of 101,325 Pa and a static temperature of 288K for a Mach (M) 1.5 (510.4 m/s) flow. The jet total pressure was 34.5 MPa, giving a jet total-to-free stream static pressure ratio (PR) of 340.5. The jet total temperature was 2,700K.

The choice of turbulence model is a key factor in the numerical modeling of complex flows such this and CFD++[17] has a large set of turbulence models available. For this study, the two-equation Menter's Shear Stress Transport (SST) model.[19] The mass and energy flux through the jet orifice was tracked and usually converged before the projectile forces and moments. Typically, 3,000 to 4,800 iterations were required to reach steady-state solutions. Dual time-stepping was used for the transient simulations using 5 to 20 inner-iterations.

A. Effects of Spin—Rotating Reference Frame

The effect of spin on the JI was first investigated using a rotating reference frame simulation—a steady-state simulation at zero angle-of-attack. Weinacht[20] previously found, in a supersonic cross-flow, that only relatively high rates-of-spin affected the JI and that non-spinning simulations could be used to predict the JI on rolling munitions. Figure 4 shows the pressure contours resulting from steady-state jet on the top and the port side of the projectile at spin-rates of 0, 10, 100, and 1,000 Hz (non-dimensional spin rates of 0, 0.0018, 0.018, and 0.18, respectively). On the top surface, no noticeable effect is observed until the spin rate of 1,000 Hz. The side view shows little effect on the tail-fin/hub region at 100 Hz (Figure 4g).

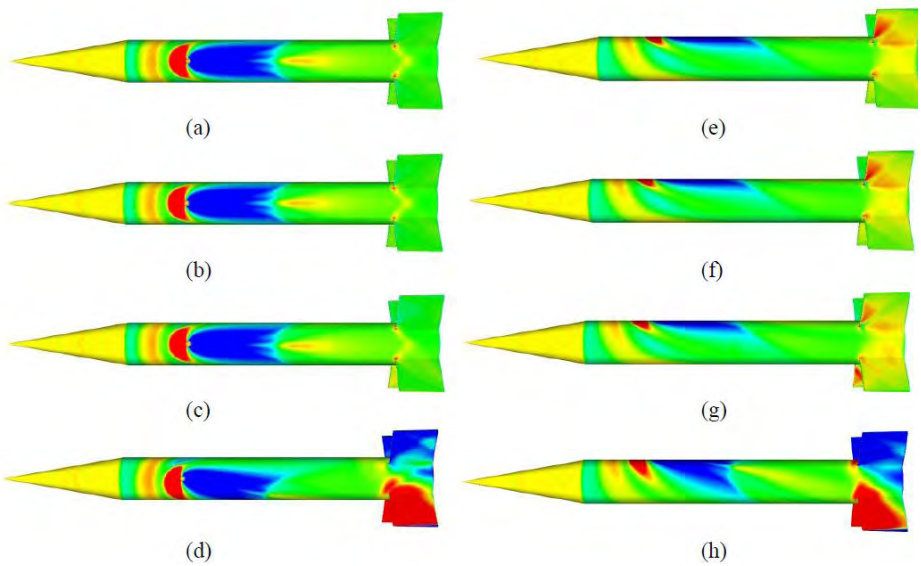
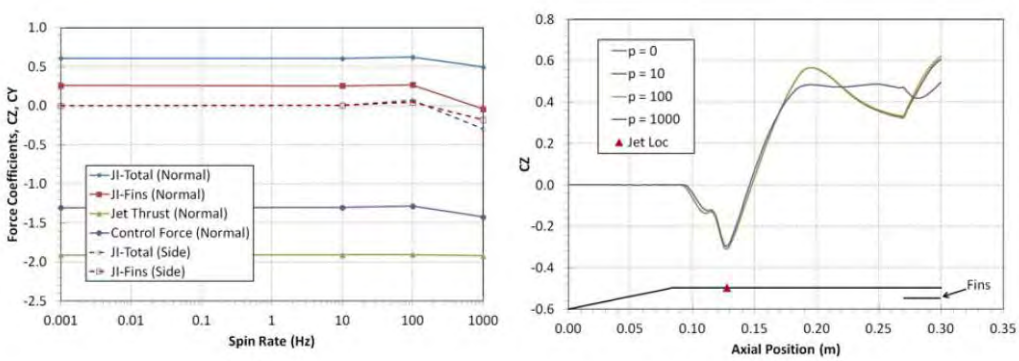


Figure 4. Pressure coefficient contours (scale: -0.4≤c_p≤0.3) on top (a-d) and port side (e-h) of projectile for spin-rates of 0, (a, e), 10 (b, f), 100 (c, g), and 1,000 (d, h) Hz for *M*=1.5, PR=340, α=0°

These effects are quantified in Figure 5a, which shows the normal and side forces due to the lateral jet. Note that the jet acts in the normal plane, so the jet thrust side force is zero and the side control force (i.e., jet thrust force+JI force) is equal to the side JI total force. Figure 5b shows the distributed JI force along the projectile for the four spin-rates. The forces for the 0 and 10 Hz cases overlay, while there is a very small difference for the 100 Hz case generated along the tail region, consistent with the observation in Figure 4. The effects of roll for the 1,000 Hz case are generated primarily aft of the jet, as might be expected. Also note that there is a significant effect forward of the fins.



(a) (b)

Figure 5. (a) Normal and side force vs. spin-rate and (b) distributed JI force; *M*=1.5, PR=340, α=0°

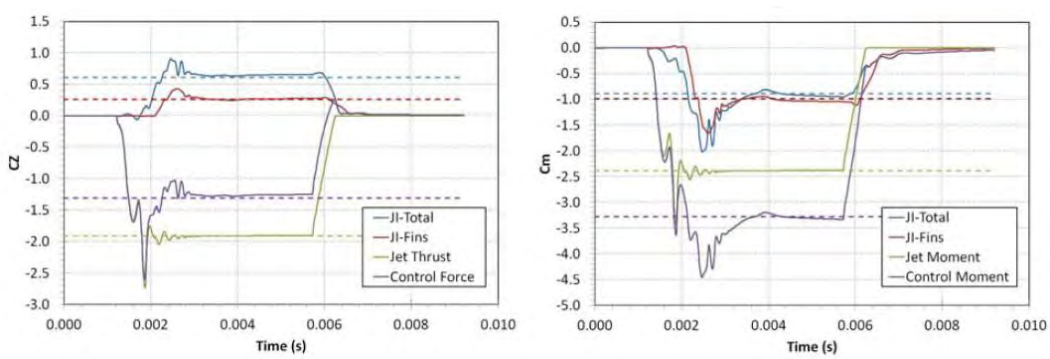
These observations can be explained by comparing the time it takes a particle to travel from the jet exit to the base of the projectile.[20] For *M*=1.5 flow with the jet in the location shown in Figure 4, this average time is about 0.34 ms. In this time, the projectile as rotated only 1.2° at 10 Hz and 12.2° at 100 Hz, both much less than a quarter of a roll cycle. However,

at 1,000 Hz, the projectile has rotated 122°, or one-third of a roll cycle. The roll induced effects at the high spin-rate leads to a reduced JI force, which will increase the jet force amplification factor.

For rolling fin-stabilized projectiles, the roll-rate is typically less than 100 Hz, therefore simulations of a non-rolling projectile should adequately predict the JI effects and those predicted forces and moments can be used in flight simulations of rolling projectile. However, spin-stabilized projectiles typically spin at rate well above 100 Hz and roll-effects should be quantified for lateral jet simulations in that case. Note that in Figures 4d and 5b, the JI effects due to spin begin well-forward of the fins, indicating that a large portion of the jet-wake is affected.

B. Transient Pulsed-Jet Effects

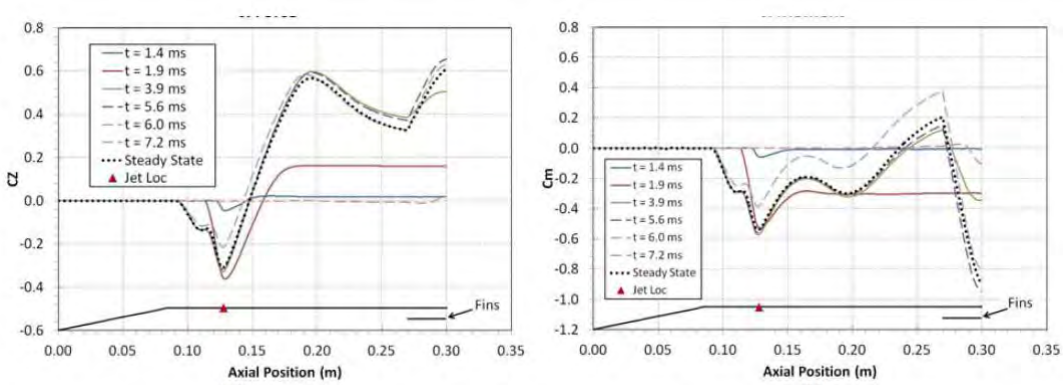
The performance of a pulsed-jet was investigated and are compared to previously obtained steady-state jet simulations.[14] A 5-ms pulse with a jet PR of 340 was simulated (1.08 N-s impulse jet) with a 10% rise and fall time so that the full pressure-ratio was on for 4 ms. Figures 6a and 6b show the resulting forces and moments, respectively, for a 9.2 ms total-time. The forces and moments are shown for the total JI on the projectile ("JI-Total") and the JI on the fins alone. Also shown is the jet thrust force and the moment due to the jet thrust and the resultant control force and moment. The equivalent values resulting from steady-state simulations are shown as dashed lines. The jet thrust "jet-on" transient is about 1.5 ms, but the control force "jet-on" transient is about 2 ms due to the JI force transient effects. The moment due to the jet thrust "jet-on" transient is about the same duration as the jet force transient. However the control moment "jet-on" transient is about 3 ms, due to the longer duration of the JI moment transient. The "jet-off" JI force transient is about 1 ms, while the JI moment transient is about 2–3 ms in duration. Once the "jet-on" transient has ended, the pulsed forces and moments are very similar to those generated in the steady-state simulations.



(a) (b)

Figure 6. Forces (a) and moments (b) during the pulse jet, *M*=1.5, PR=340, α=0°, t_{jet}=5 ms

Figure 7 shows the distributed JI force and moment at six times during the pulse and also the steady-state simulation result. The times correspond to one-half the rise time of the pulse (1.4 ms), the peak of the "jet-on" transient (1.9 ms), one-half the pulse (3.9 ms), just before the pulse decrease (5.6 ms), one-half the fall time of the pulse (6.0 ms), and the end of the pulse (7.2 ms). The trace for *t*=5.6 ms, near the end of the plateau region



(a) (b)

Figure 7. Distributed JI force (a) and moment (b) along projectile vs. time, *M*=1.5, PR=340, α=0°, t_{jet}=5 ms

Results presented here show, as observed by Weinacht,[20] that the steady-state simulations provide reasonable predictions of the magnitude of the JI forces and moments for a rolling, fin-stabilized projectile. Future work will include investigating the transient JI effects for the case of a rolling projectile to determine if any unsteady effects are present due to roll; performing six degrees-of-freedom flight trajectory simulations to compare "idealized" control force pulse inputs from the steady-state predictions to actual control force pulse inputs generated from the transient simulations; and performing simulations using a coupled CFD/RBD technique to determine if there are any unsteady, flow-coupling effects that need to be taken into consideration (e.g., will the JI force and/or moment change due to the dynamics of the rigid-body projectile motion?). From the results presented here, it is not expected that there will be flow coupling in the supersonic flight regime. However, it may be important in the subsonic and transonic flight regimes to be investigated later.

3.2 Guided Canard Maneuvers for a Canard-Controlled Projectile in Transonic Flight

One of our objectives in this Challenge Project is to develop and apply coupled CFD/RBD/FCS techniques for physics-based simulations of guided maneuvers. Some examples of these coupled virtual fly-out calculations with canard-control maneuvers are described next.

The trajectory of a projectile can also be controlled using other maneuver control mechanisms such as canards and micro-flaps. For example, the canards located in the front section of a canard-controlled projectile can be used to provide the control forces needed to maneuver a projectile. Of interest here is the control of projectile using canard maneuvers. The canards are deflected to generate control forces required to maneuver the projectile in roll, pitch, or a combination of both. For guided flight simulations, the coupled CFD/RBD/FCS method described earlier is used to predict the unsteady aerodynamics and the flight dynamics of a canard-controlled finned-projectile during the canard-control maneuvers.

The canard-controlled projectile considered in this study has two canards and six fins (Figure 8). The control maneuver is achieved by two canards located in the nose section of the projectile. All six fins are located at the aft end of the projectile. An unstructured grid is generated for the projectile body with the fins which forms the background mesh. Unstructured grids are generated about each canard separately (see Figure 9). The two canard grids are then overset with the background projectile mesh to a Chimera overlapped mesh for the canard-controlled projectile. The total number of grid-points in this case is approximately 6.3 million. This Chimera procedure[21] requires proper transfer of information between the background mesh and the canard meshes. However, the advantage is that the individual grids are generated only once, and the Chimera procedure can then be applied repeatedly as required during the canard motion. There is no need to generate meshes at each time-step during the canard maneuvers.

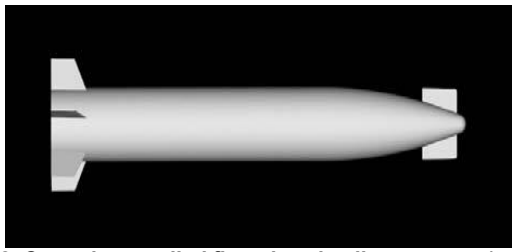


Figure 8. Canard-controlled finned-projectile geometry (top view)

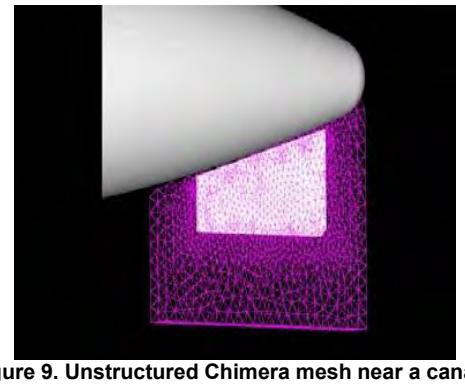
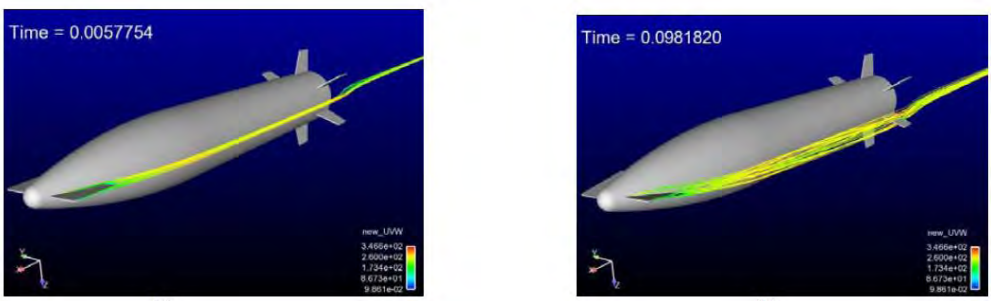


Figure 9. Unstructured Chimera mesh near a canard

Work is ongoing to demonstrate the capability of the coupled CFD/RBD/FCS with two canard-control maneuvers, a pure roll control maneuver and a cross-range control maneuver. Appropriate FCS designs or controllers have been developed for use in the coupled CFD/RBD/FCS calculations for both these cases. A simple proportional derivative control law is used for the feedback loop in the controllers for each case. In each of these two cases, the projectile initially is a straight-level flight and the initial spin-rate is set to zero. Also, the maximum canard deflection angle was prescribed. The first step in the coupled virtual fly-out simulations is the generation of a steady-state solution for the projectile moving forward at a velocity (M=0.9) at zero degree angle-of-attack. Time-accurate calculations are performed for a few hundred time-steps in an uncoupled mode. A solution obtained in the uncoupled mode serves as the starting solution for the fully-coupled CFD/RBD/FCS run. The coupled run starts to deflect the canards in accordance with the control law in the FCS design or the controller. The canard deflections are passed onto the CFD solver where the canards are actuated accordingly and CFD results are obtained in a time-accurate manner. CFD predicted aerodynamic forces and moments are then transferred to the RBD/FCS code and new RBD state variables and FCS variables (canard deflections) are computed. Some representative results obtained for both roll and cross-range control cases using coupled calculations are shown next.

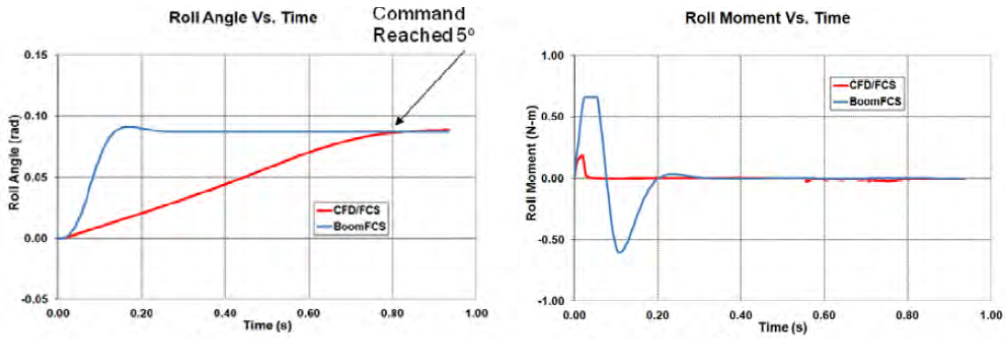
A first test case considered is the pure roll control of the canard-controlled finned-projectile. For this roll control case, canards on each side are deflected in opposite directions and the projectile starts to roll. In this case, a FCS design, i.e., a roll controller, was developed and used to command the projectile to a desired roll angle of 5°. The maximum canard deflection angle was set to 5°. Also, the canards are deflected in opposite directions that give rise to a differential force so the projectile begins to roll. The canard deflection produces positive or negative force depending on the sign of the canard deflection angle. The movement of the canards affects not only the flow locally in the nose region, but also, the flow-field downstream nears the aft body with the fins. Figure 10(a) shows the particle traces emanating from one of the canards and as shown in this figure, these particle traces pass through the region in between the two rear fins without affecting the flow near the fins. On the other hand, with the canard deflected (see Figure 10b) the canard particle traces clearly affect the flow on one of the rear fins. This interference effect of the canard on the rear fin can be significant and is taken into account in the coupled CFD/RBD/FCS calculations.



(a) (b)

Figure 10. Effect of canard particle traces on rear fins, M=0.9, α=0°; a) canard undeflected (b) canard deflected

Figures 11(a) and 11(b) show the time histories of the roll angle and the roll moment of the projectile, respectively. A computed result from the coupled CFD/RBD/FCS calculation is shown in red and is compared with the results obtained using a standard control design method (BoomFCS) shown in blue. As shown in Figure 11(a), initially the roll-angle is zero, starts to increase and reaches the commanded value, 5° quickly (in about 0.2 sec) in the standard control simulation. However, the coupled CFD/RBD/FCS calculation predicts a much slower response and reaches the commanded roll-angle in about 0.8 sec. Figure 11(b) shows the roll moment of the projectile used in both methods. As seen here, the roll moment predicted in the coupled method using CFD is generally much smaller in magnitude compared to that used in the standard control design method. Unlike the standard control method, the coupled method takes into account the aerodynamic interference effects of the canards on the downstream on the body including the fins. Although the canard deflections produce a positive roll, the interference effect on the rear fins adversely results in a negative roll moment contribution. The total roll moment in the coupled calculation is, thus, not as large as that used in the standard control method.



(a) (b)

Figure 11. Roll-angle (left) and roll moment (right) as a function of time

Figure 12 shows the spin-rate as a function of time. Results obtained from both methods are compared to each other. Again, the time responses are quite different. Initially, both methods show the spin-rate to increase to about 0.1 and match well. In the coupled method it then remains constant until t=0.6 s and gradually drops to zero with increasing time. The standard method shows a big rise in the spin-rate to 1.1, which is consistent with the larger rolling moment used. Figure 13 shows the actual deflection of one of the canards used in the coupled CFD/RBD/FCS method. With the FCS on, the canard starts deflecting and reaches -5° its minimum, very quickly and stays constant at that value for a while until t=0.55 s and reaches almost zero towards the end of the simulation. These results correspond to one FCS control design. The maximum canard deflection angle and control law parameters can be adjusted and varied to obtain a desired roll-control characteristics.

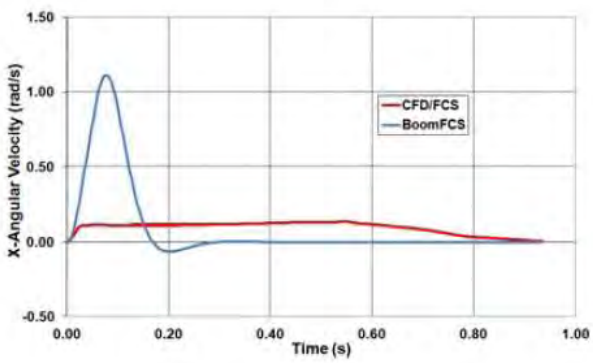


Figure 12. Spin-rate vs. time

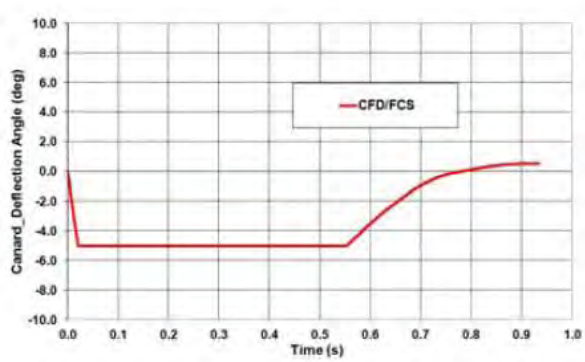
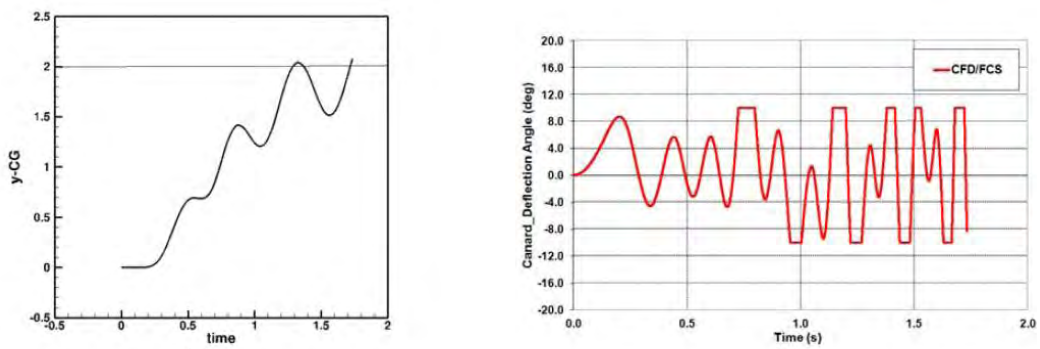


Figure 13. Canard deflection angle vs. time

Another case considered is the cross-range control of the same canard-controlled finne-projectile. In this case with the cross-range control, canards on each side are deflected in the same direction as the projectile starts to spin and move forward. In this case, a FCS design i.e., a cross-range controlle, was developed and used to command the projectile to a desired cross-range (y) of 2m.

Figure 14(a) shows a preliminary result obtained using the coupled-CFD/RBD/FCS simulation. It shows the cross-range (y) as a function of time. As seen in this figure, it takes a little less than two seconds to reach the commanded cross-range value of 2 m. Although we are able to guide the projectile to its commanded value, the response seems to be oscillatory in nature, and is perhaps not desirable. The corresponding canard deflections used in the coupled calculation is shown in Figure 14(b). Initially, the canard deflections are smooth up to 0.6s. Later in time one can notice that the canard deflection reaches the maximum (+10°) and minimum (-10°) which indicate that the controller is perhaps saturated. The coupled results shown here correspond to one cross-range controller design. The response of the cross-range control maneuver can be improved with other controller designs, but the coupled results do demonstrate the capability of the coupled-CFD/RBD/FCS for guided flight of a canard-controlled projectile.



(a) (b)

Figure 14. Cross-range distance (left) and Canard deflection angle (right) as a function of time

Currently, work is in progress and coupled-CFD/RBD/FCS simulations are being performed to investigate the effect of various CFD as well as FCS parameters. Details of these results will be presented at the 7th International Conference on CFD.[22]

4. Concluding Remarks

This paper describes recent progress in the development and application of advanced time-accurate CFD and coupled-CFD/RBD/FCS multidisciplinary techniques for prediction of unsteady aerodynamics and flight dynamics of maneuvering projectiles. All computations were performed on highly-parallel Cray XE-6 and SGI Clusters using an advanced scalable unstructured flow-solver.

Computed results have been obtained for steady and pulsed-jet interactions with supersonic free-stream flow for a finned-projectile. Computed steady-state results show reasonable predictions of the magnitude of the JI forces and moments for a rolling, fin-stabilized projectile. More research is planned to investigate the transient JI effects for the case of a rolling projectile to determine if any unsteady effects are present due to roll, and also to perform simulations using a coupled CFD and RBD technique to determine if and how the JI force and/or moment change due to the dynamics of the rigid-body projectile motion.

Development of the coupled-CFD/RBD/FCS procedure continues. This coupled capability is being exercised and demonstrated on a canard-controlled projectile. Physics-based, virtual fly-out simulations were performed for a guided pure roll-control maneuver and a guided cross-range control maneuver using the coupled procedure. Computed results clearly show the projectile maneuvered to its commanded values. Currently, work is in progress to improve the commanded response of the projectile with improved FCS controllers. Future research efforts will extend the capability of the coupled procedure for simulation of guided control maneuvers due to jets, micro-flaps, and other flow control mechanisms. These research efforts will provide accurate unsteady aerodynamics and flight dynamics predictions associated with guided maneuvers. This will ultimately lead to increased maneuverability for the next-generation low-cost munitions to hit moving targets.

Acknowledgments

The authors would like to thank the Department of Defense High Performance Computing Modernization Program Office for sponsoring this work as a part of a Grand Challenge Project and for providing critically needed HPC resources ARL DSRC and AFRL DSRC for successful accomplishment of the work. The authors also wish to thank Prof. Mark Costello of Georgia Institute of Technology and Dr. Sukumar Chakravarthy of the Metacomp Technologies for their technical assistance in the continued development of coupled technique.

References

1. Sahu, J., "Physics-based Virtual Fly-outs of Projectiles on Supercomputers." *Proceedings of the 23rd Army Science Conference*, Orlando, FL, November 27–30, 2006.

2. Sahu, J., "Time-Accurate Numerical Prediction of Free-Flight Aerodynamics of a Finned Projectile." *AIAA Journal of Spacecraft and Rockets*, Volume 45, Number 5, September/October 2008.

3. Silton, S., "Navier-Stokes Computations for a Spinning Projectile from Subsonic to Supersonic Speeds." *Journal of Spacecraft and Rockets*, Vol. 42, No. 2, pp. 223–231, 2005.

4. DeSpirito, J. and Plostins, P., "CFD Prediction of M910 Projectile Aerodynamics: Unsteady Wake Effect on Magnus Moment." *AIAA-2007-6580*, August 2007.

5. Sahu, J., "Computations of Unsteady Aerodynamics of a Spinning Body at Transonic Speeds." *AIAA paper No. 2009-3852*, 27th Applied Aerodynamics Conference, San Antonio, TX, 22–25 June 2009.

6. Sahu, J .and Heavey, K.R., "Progress in Simulations of Unsteady Projectile Aerodynamics." *DoD Users Group Conference*, Schaumburg, IL, June 2010.

7. Sahu, J. and Heavey, K.R., "Time-Accurate Computations for Rapid Generation of Missile Aerodynamics." *AIAA Atmospheric Flight Mechanics Meeting*, Toronto, Canada, 2–6 August 2010.

8. Sahu, J .and Heavey, K.R., "Numerical Simulations of Unsteady Projectile Aerodynamics." *DoD Users Group Conference*, Portland, OR, June 2011.

9. Sahu, J. and Heavey, K.R, "Unsteady CFD Modeling of Micro-Adaptive Flow Control for an Axisymmetric Body." *International Journal of Computational Fluid Dynamics*, Vol. 5, April–May 2006.

10. Costello, M., "Extended Range of a Gun Launched Smart Projectile Using Controllable Canards." *Shock and Vibration*, Vol. 8, No. 3–4, pp. 203–213, 2001.

11. Sahu, J., "Unsteady CFD Modeling of Aerodynamic Flow Control over a Spinning Body with Synthetic Jet." *AIAA Paper 2004-0747*, Reno, NV, 5–8 January 2004.

12. Margason, R.J., "Fifty Years of Jet in Cross Flow Research." *AGARD 72nd Fluid Dynamics Panel Meeting, Paper 1*, 1993.

13. Champigny, P., and Lacau, R. G., "Lateral Jet Control for Tactical Missiles." *Special Course on Missile Aerodynamics, AGARD-R-804, Paper 3*, 1994.

14. DeSpirito, J., "Factors Affecting Reaction Jet Interaction Effects on Projectiles." *AIAA-2011-3031*, June 2011.

15. DeSpirito, J., "Lateral Jet Interaction on a Finned Projectile in Supersonic Flow." *AIAA-2012-0413*, January 2012.

16. Peroomian, O., S. Chakravarthy, S. Palaniswamy, and U. Goldberg, "Convergence Acceleration for Unified-Grid Formulation Using Preconditioned Implicit Relaxation." *AIAA Paper 98-0116*, 1998.

17. Peroomian, O. and Chakravarthy S., "A 'Grid-Transparent' Methodology for CFD." *AIAA Paper 97-0724*, Jan. 1997.

18. Goldberg, U., O. Peroomian, and S. Chakravarthy, "A Wall-Distance-Free K-E Model With Enhanced Near-Wall Treatment." *ASME Journal of Fluids Engineering*, Vol. 120, pp. 457–462, 1998.

19. Menter, F.R., "Two-Equation Eddy-Viscosity Turbulence Models For Engineering Applications." *AIAA Journal*, Vol. 32, No. 8, pp. 1598–1605, 1994.

20. Weinacht, P., "Lateral Control Jet Aerodynamic Predictions for a 2.75-in Rocket Testbed Munition." *ARL-TR-3165*, Army Research Laboratory, Aberdeen Proving Ground, MD, April 2004.

21. Sahu, J. and C.J. Nietubicz, "Application of Chimera Technique to Projectiles in Relative Motion." *Journal of Spacecraft & Rockets*, September–October 1995.

22. Sahu, J., Costello M., and Montalvo, C., "Development and Application of Multidisciplinary Coupled Computational Techniques for Projectile Aerodynamics." *7th International Conference on Computational Fluid Dynamics*, Kohala Coast, HI, June 2012 (accepted).

Workstation Footprint Tactical Computing Demonstration

Song J. Park, Dale R. Shires, and Brian J. Henz
US Army Research Laboratory, Aberdeen Proving Ground, MD
{song.j.park, dale.r.shires, brian.j.henz}.civ@mail.mil

James A. Ross and Jordan J. Ruloff
Dynamics Research Corporation, Andover, MA
{jruloff, jaross}@drc.com

David A. Richie
Brown Deer Technology, Forest Hill, MD
drichie@browndeertechnology.com

Abstract

In terms of computing hardware, heterogeneous processor types have now become an integral part in a number of modern devices. In particular, graphics processing units are complementing central processing units from portable smartphones to large scale supercomputers. Having a mixture of resources available, optimally mapping algorithms to computing architectures improves performance and saves power. With the advances in raw theoretical floating-point processing power of massively parallel graphics processors, mobile high performance graphics processing units (GPU)-populated workstations are now a feasible option for advancing compute-intensive tactical computations onboard. This compact form of computing system is evaluated and leveraged for enhancing a tactical operation scenario consisting of determining ballistic threat field in a three-dimensional urban environment. The core algorithm performs ray-plane intersection tests on a triangle mesh input data using kernels developed in the OpenCL framework to exploit the parallel processing elements of AMD and Nvidia products. To deliver an intuitive interface to the end user that serves as a layer of abstraction to GPU computing, the interactive world viewer named World Wind was employed for the static optimization ballistic threat demonstration. The World Wind package provides a display and interaction functionality for the workstation with multiple consumer-grade graphics cards installed. Opportunities of hybrid core deployable systems as tactical computing assets are investigated in respect to performance and programming development.

1. Introduction

The popularity of heterogeneous computing architectures, such as central processing units (CPUs) with graphics processing units (GPUs), is spreading from personal computers and laptops to mobile devices and high performance computing (HPC) facilities. The use of graphics accelerators can be observed in ARM-based PowerVR incorporated processors in smartphones[1] and in AMD or NVIDIA video card augmented supercomputers[2–4]. Similarly, the hybrid computing concept is being leveraged by select Intel's Sandy Bridge processors, iPhone's A5 processors, and AMD's Fusion processors. Undoubtedly, the collaborative CPU and GPU technology is being utilized in many different fields. This multitude of driving forces, along with traditional purposes such as driving displays and gaming, continues to push the demand for video processing forward. Note that, unlike specialized and custom designed processors, a popular consumer demand exists for a various GPU products. Pervasive characteristic of GPU products typically leads to continual advances in hardware and software that increase performance as a function of time. The risk of technology fading away over time is mitigated for the GPU case. Low entry point of cost for GPU computing technology is another resulting aspect of mass marketed GPUs.

In addition to being ubiquitous, GPUs pack a large number of math function processing units since the parallel nature of graphics applications calls for a highly parallel architecture. For instance, a single GPU chip from AMD's Radeon HD 7970 contains 2,048 stream processors leading to a peak performance of 3.7 trillion floating-point operations per second

(TFLOPS) for single-precision[5]. As indicated by the theoretical performance, discrete graphics cards have become quite powerful in recent years, ideal for applications with high-level of data parallelism. For this project, the main objective was to leverage GPU's floating-point capabilities for advancing military applications. For instance, tackling compute intensive algorithms with GPU assisted systems. Basically, using graphics hardware for the purposes of number crunching rather than rendering images on a screen.

2. Tactical Mobile HPC

One of the visions behind this research was to provide compute capabilities similar to supercomputing in the hands of the Soldiers. The objective is to strike a balance between portability and performance. Limiting physical dimensions of such system to a size of a workstation, the project attempted to pack as much processing power as possible. Graphics cards were the preferred hardware of choice to ramp up the raw floating-point calculations capability of a workstation footprint system.

At the high-end of the spectrum, a dual-GPU consumer card from AMD rated at 1.2 TFLOPS in double-precision are available. A few years back, stacking four of these cards would equate to a theoretical peak performance of 4.8 TFLOPS, and would place the system in the TOP500 list[6]. Differences being the following: single outlet socket for power, lacking dedicated cooling infrastructure, and no specialized room requirement. In other words, a five-year-old supercomputer can be constructed using off-the-shelf consumer products within a workstation form factor. Moreover, cost associated with general-purpose computing with GPU platforms is minimal and thus redundancy placement of these resources can be achieved to address frequently disconnected users in the field as described in Reference 7.

The HPC workstation box represents a portable computational power in the field that allows for the possibility of onboard processing of complex computations. With enhanced computing resources, time sensitive information can be extracted near-real-time, fidelity of certain calculations can be improved, and greater amounts of data can be processed. For the mobile HPC study, three different hardware configurations of GPU equipped workstations were procured for evaluations. All machines contain dual Intel Xeon X5675 CPUs and 24GB of memory as a base configuration. The first machine has four AMD Radeon HD 6770 cards, the second machine has four NVIDIA GeForce GTX 580 boards, and last machine has one NVIDIA GeForce GTX 590 model installed. Figure 1 illustrates the workstation from SuperMicro populated with four NVIDIA GPUs.

Figure 1. Four NVIDIA GPUs installed in a workstation

Adding performance costs power. Typical thermal design power (TDP) for high-end graphics cards is 250 W for single GPU cards and 375 W for dual-GPU cards. TDP is a value calculated by the manufacturer that describes an average maximum power under normal and realistic use. It does not, however, indicate an absolute maximum, where TDP is 20-30% less than the maximum typically. With four single graphics chip boards, 1400 W power supply handles the power requirement of the quad GPU workstation. To identify power source availability in a mobile setting, a search was performed on vehicle power system options and found that heavy-duty inverters from Analytic Systems can output 2000 W[8], and advanced custom-engineered power solutions can produce up to 30 kW at higher engine speeds for military vehicles integrated with a 45 kW generator[9]. On-board vehicle power supply requirement by Marine Corps specifies minimum output of 1800 W for inverters and 10 kW for the on-the-move HMMWV-based systems[10]. Therefore, a cursory look at these vehicle power specifications insinuates that the electrical power requirement of a multi-GPU workstation in a mobile platform is within attainable scope.

3. Ballistic Threat Surveillance Optimization Application

The core computation in the ballistic threat surveillance optimization algorithm is the calculation of line-of-sight ballistic hit probability in an urban environment. Figure 2 displays the three-dimensional (3D) visualization of the input triangle mesh representing building structures and terrain information. Based on the first-hit ray tracing algorithm, quad-

tree spatial decomposition was adopted to accelerate the calculations of line-plane intersections. In computing line-plane intersections, a series of rays are cast out from the shooter's position and propagated until a first intersection with a polygon is detected. In addition to computing line-of-sight, a ballistic hit probability formula is applied to the calculation to model ballistic characteristics as a function of distance. Figure 3 illustrates an output result for the ground plane line-of-sight ballistic threat calculation, watermarked in red, for a single shooter on the roof.

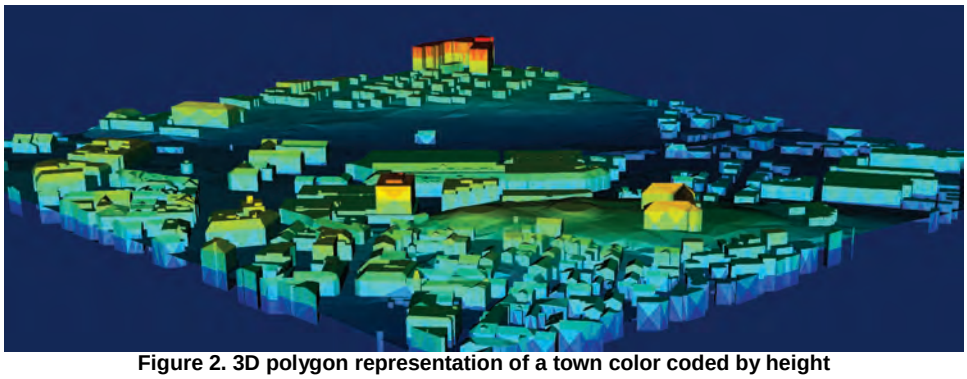


Figure 2. 3D polygon representation of a town color coded by height

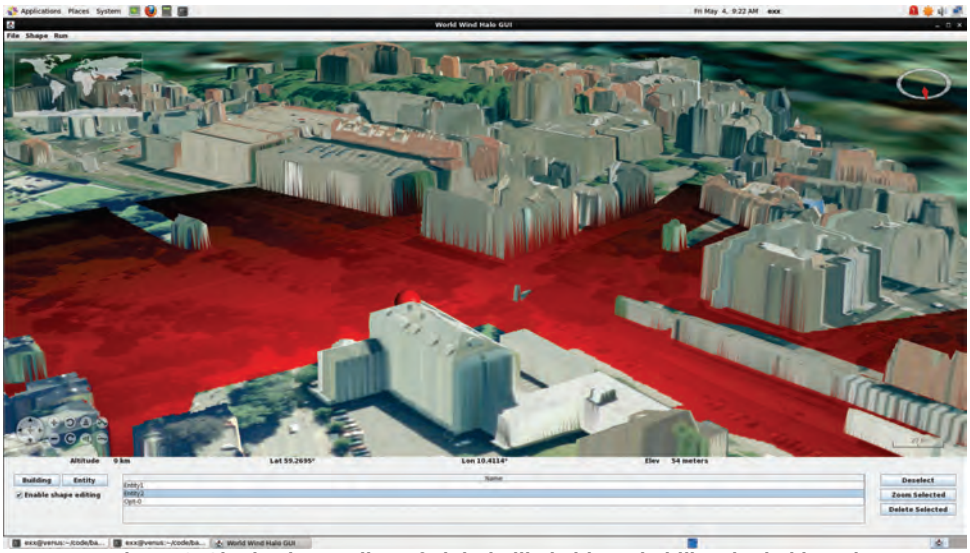


Figure 3. Single shooter line-of-sight ballistic hit probability shaded in red

The algorithm supports a multitude of scenarios by using an extensible XML format as scenario inputs that enables arbitrary multiplicity in entities and computations. The construction of scenarios would be specified by the XML file. The technical objective was to support arbitrary numbers of simulation elements. The basic elements of a scenario are captured by the object types categorized as map, view, entity, and evaluator. As the name implies, the map field would contain terrain input information, which could be LIDAR or 3D polygon representation. The type view can represent rendering engine, camera perspective, or window size. The type entity can specify shooters, observers, vehicles, or sensors. And, the type evaluator is a compute kernel that performs computationally intensive calculations. Once the functionality of ballistic threat field calculation was implemented, the application was augmented to solve for situations calling for a static optimization technique.

The objective of the static optimization was to position recon Soldiers such that the ballistic threat is minimized while maximizing the line-of-sight observation to a point of interest. The goal is to determine the optimum locations of one or more Soldiers while minimizing risk. Markov Chain Monte Carlo sampling technique was employed to obtain the optimal locations in a high dimensional space. Markov Chain Monte Carlo provides a means to generate a sequence of random samples that explores the space of high probability. Surveillance optimization application is interfaced with the World Wind software allowing for users to place hostile shooters and watch points interactively. World Wind Java from NASA provides an interactive and intuitive functionality to the compute intensive OpenCL code. Figure 4 is the demonstration interface presenting the optimization results, indicated in blue dots, for the input parameter of two shooters and two watch

points. For this example, watch points were placed on the entry points of buildings and hostile shooters were separately located; one on the building's roof and the other on the field.

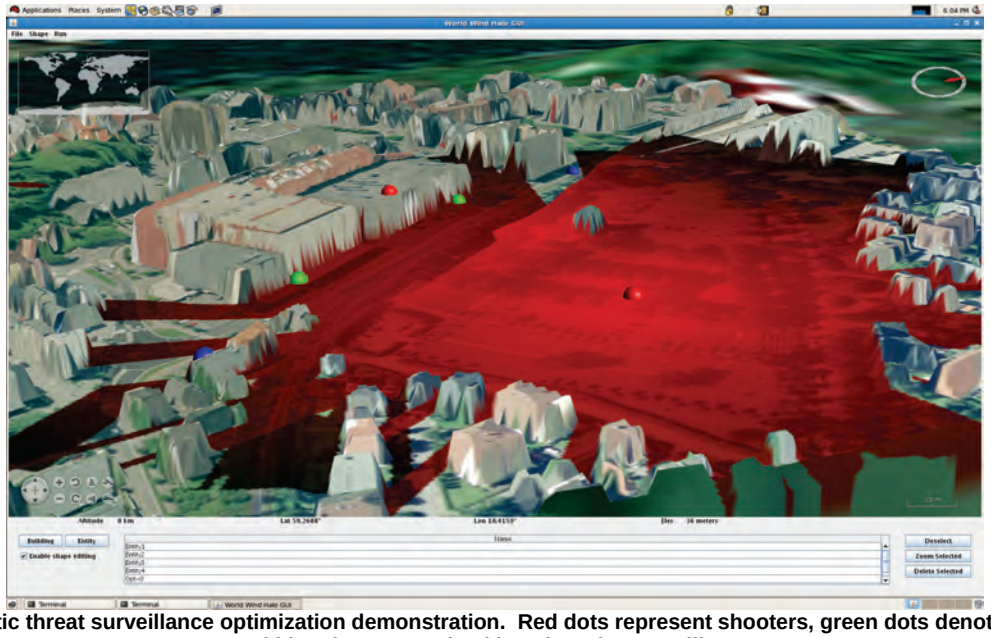


Figure 4. Ballistic threat surveillance optimization demonstration. Red dots represent shooters, green dots denote watch points, and blue dots are optimal locations for surveillance.

4. Timing Results

Execution runtime measurements for the case with four shooters and four watch points was benchmarked on both x86 processors and multiple discrete graphics processing units. One of the advantages of OpenCL source code is the portability across various processing architectures. Furthermore, the x86 implementation of OpenCL optimizes for execution on all available logical cores and enables SSE operations for traditional CPUs. On a dual-core Intel processor, the execution time exceeded six minutes, which fails in creating a quick and responsive interactive user experience. Even dual hexa core from Intel, equivalent to 24 logical cores, requires over half a minute to complete the ballistic threat surveillance optimization calculation. By leveraging four GPUs in a workstation, the wait time falls within a reasonable window for interactive experience.

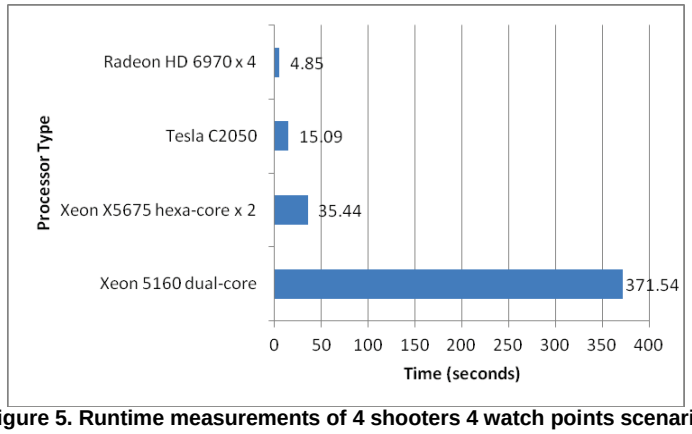


Figure 5. Runtime measurements of 4 shooters 4 watch points scenario

Parallel GPU execution presented unforeseen challenges during the development stage. Linux support for single node concurrent multiple GPU execution failed to work correctly depending on the version of the graphics drivers. The workload was divided and distributed across separate graphics cards, but the actual execution on each GPU was serialized, failing to achieve concurrent execution in all GPUs. Rolling back to the previous generation of driver fixed the multi-GPU

issue. However, odd system behaviors were observed when reverting back to a previous graphics driver version, requiring a fresh installation. So far, developers can't expect to see the same level of stability seen in the x 86 tool chains in GPU technology.

5. Conclusion

Driven by the mass market demand, theoretical computational power has become affordable to the general public. In terms of peak performance, previous generations of supercomputing capability are attainable in a smaller form factor with manageable power. The project intended to evaluate the transfer of computational capability into a workstation footprint to assist tactical computing. However, optimally leveraging underlying compute resources rests on software development. Regardless of processor type, embracing parallelism is the key for achieving higher performance. Even the mainstream processors from Intel and AMD require parallel programming to take advantage of their multiple cores. The allure of OpenCL is the code portability across different architectures. Once an algorithm is developed in OpenCL framework, it can run on AMD graphics cards, NVIDIA boards, and Intel processors. OpenCL approach, although low-level, allows developers to circumvent being restricted to a particular vendor or architecture.

This research explored a deployable high performance computing (HPC) system in a workstation footprint. With the goal of supporting a realistic tactical application, the ballistic threat surveillance optimization algorithm was implemented for assessment. The amount of parallelism and the computational requirement of line-of-sight hit probability calculations make the algorithm a good candidate for justifying tactical HPC. The combination of mobile HPC and an intuitive software interface delivers enhanced capability out in the field for information and processing superiority.

Acknowledgements

The authors would like to acknowledge the support provided by the DoD High Performance Computing Modernization Program Office.

References

Imagination Technologies, *PowerVR graphics driving mass market smartpone graphics adoption*, http://www.imgtec.com/corporate/newsdetail.asp?NewsID=669.

IEEE Spectrum, E Guizzo, *China Builds World's Fastest Superomputer*, Nov 2010.

GSIC. TSUBAME Computing Services High Performance Computing at Tokyo Tech. http://tsubame.gsic.titech.ac.jp/en/tsubame2-system-architecture.

Vuduc, Richard; Czechowski, Kent; , "What GPU Computing Means for High-End Systems," *Micro, IEEE* , vol.31, no.4, pp.74-78, July-Aug. 2011.

Advanced Micro Devices, Inc, *AMD Radeon HD 7970 Graphics*, http://www.amd.com/us/products/desktop/graphics/7000/7970/Pages/radeon-7970.aspx.

TOP500 Supercomputer Sites, *TOP500 List – November 2006 (401-500)*, http://www.top500.org/list/2006/11/500.

MG Flynn, Michael T., Memo to the Department of Defense, 2 July 2010.

Analytic Systems, *Power Conversion Solutions Guide*, August 2009.

BAE Systems, *Advanced Power Management for Military Vehicles*, March 2007.

Marine Corps, "On-Board Vehicle Power Systems for Legacy Military Vehicles." *Joint Service Power Exposition 2007*.

6.2 Applied Research

Navy

Application Development on Convey's HC-1ex Using Bluespec Tools at the Naval Research Laboratory

Stephen Bique
US Naval Research Laboratory (NRL-DC), Washington, DC

Rishiyur Nikhil
Bluespec, Inc., Framingham, MA
nikhil@bluespec.com

Abstract

We present an overview of Bluespec's development tools and libraries aimed at dramatically simplifying the development of applications for the Convey HC-1ex platform. After briefly reviewing the HC-1ex architecture, we describe Bluespec's BSV high-level hardware description language, and the libraries and transactors that provide a high-level interface to the base hardware facilities in the Convey HC-1ex. We describe Bluespec's infrastructure for "whole-system" simulation before actual deployment on the Convey hardware, and the final deployment process. We show results based on two illustrative examples: a simple vector-add, and a larger DNA sequence exact-matcher involving complex memory access patterns.

1. Introduction

One direction for future HPC (high performance computing) is towards "Extensible CPUs" or "Hybrid Architectures", where an existing base instruction set is extended with custom instructions for computational kernels that are implemented in adjoining hardware in field-programmable gate array (FPGA) or application-specific integrated circuit (ASIC) technology. The kernel of a target application is separated into a part that is implemented in hardware. This is invoked by the remainder of the application, which remains in software. The hardware part exploits fine-grain parallelism and direct hardware implementation for improved performance and power efficiency, both of which are of significance to Department of Defense (DoD).

Example products today include the Convey HC-1ex and Convey HC-1 (with Intel x86), Xilinx Zynq (with ARM), Nallatech (with Intel x86), DRC (with AMD x86) and TenSilica (with TenSilica Xtensa). In some of these new hybrid architectures, the CPU and the extension have a shared view of memory.

We first describe the Convey HC-1ex platform, and its existing application development facility known as the Convey "PDK" (Personality Development Kit). Then we describe Bluespec's BSV high-level language for hardware description and facilities for application development on the Convey platform. Section 4.2 we describe the steps for deployment of the tested application on the Convey platform. Next, we describe two example applications developed this way, and we conclude with status and prospects.

2. The Convey HC-1ex Hybrid Computing Platform

2.1 Architecture

Figure 1 shows the architecture of the Convey HC-1ex[2,3]. In a standard 2-socket Xeon server motherboard, one of the sockets is connected to a Convey board containing a scalar coprocessor, four Xilinx Virtex 6 LX760 FPGAs, and eight banks of memory (the only difference in the HC-1 variant is that it uses Virtex 5 LX330 parts instead of Virtex 6 FPGAs). Under the hood, the Convey board contains additional FPGAs, but they cannot be used as targets for user code. In the sequel, FPGA refers to one of the four so-called "Application Engines" (AEs).

The memory subsystem is perhaps the most interesting. Each FPGA has eight independent channels to the eight memory banks, through a memory interconnect. The overall bandwidth to the eight memory banks is 80 GBytes/sec. Read requests, read responses, and write requests are sent on separate sub-channels, and are highly pipelined. Reads may complete out of order, so read requests and responses carry a 32-bit "read control" tag for identification. Write requests are "fire-and-forget" from each FPGA (i.e., no responses); they may complete out of order. When it is necessary to enforce write-orderings from the FPGA, each memory channel also has a "flush" functionality. The memory on the x86 board and the eight memory banks on the coprocessor board are all seen by an application's software (SW) and hardware (HW) in a common, cache-coherent, virtual address space (but there is a performance penalty if the x86 or an FPGA "reaches across" to access memory on another board).

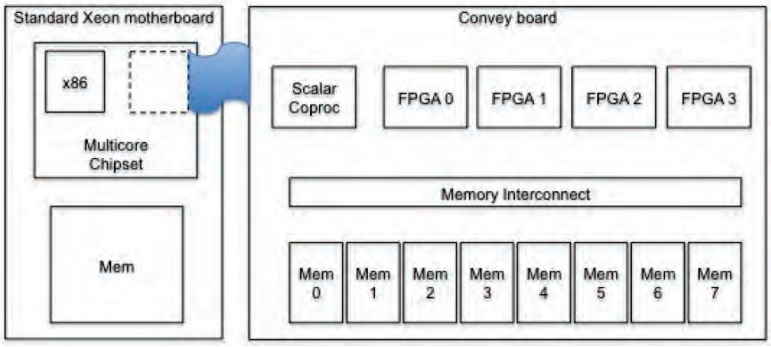


Figure 1. The Convey HC-1ex and HC-1 architecture

The Convey scalar coprocessor on the FPGA board is a complete traditional integer processor. Some of its instructions communicate values to and from the FPGAs. Routines on the coprocessor are invoked by standard x86 coprocessor instructions running on the x86. The FPGAs are also connected in a ring allowing direct neighbor communication.

2.2 Convey Personality Development Kit (PDK)

The meaning of each x86 coprocessor instruction depends on both the programming of the scalar processor and the FPGAs. Convey refers to each such configuration as a "personality" for the x86[5]. Many canned personalities are available, but Convey also supplies a PDK so users can create new custom computations. This customization involves the following coding:

- Creating the code for the scalar coprocessor. This can be written in C or assembler and compiled using Convey's compilers[4].
- Programming the FPGAs.

Fortunately in the latter coding, the programmer does not have to struggle with pin-level I/O on the FPGAs. Convey provides Verilog adapters that handle pin-level I/O, and offers a Verilog API to the programmer's Verilog. This abstraction is illustrated in Figure 2. A programmer creates a Verilog module `cae_pers.v` whose interface has four groups. The "Dispatch" and "Management" interfaces manage communication with the scalar coprocessor (the latter interface is mainly for debugging and diagnostics). The "MC" interface handles communication with the memory subsystem. The "AE-to-AE" interface manages FPGA-to-FPGA neighbor communication. Each of these Verilog interfaces is described with traditional Verilog ports and flow control (enable and stall signals). The Convey Verilog infrastructure also provides clocks and reset signals.

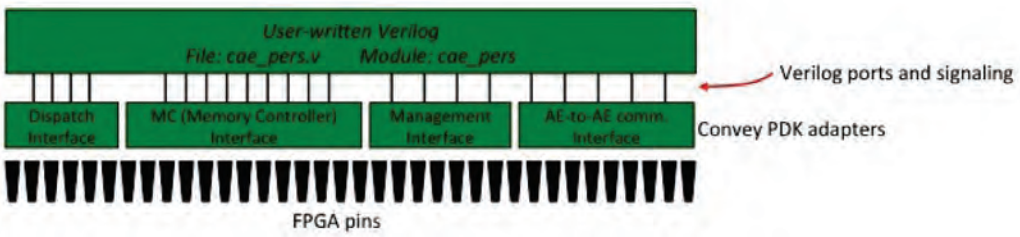


Figure 2. The Convey PDK

2.3 Debugging with Simulation Using the Convey PDK

Convey's PDK includes a simulation setup for debugging the combined software/hardware co-design. A simulation runs as two processes under Linux, communicating via a TCP link. One process represents the x86 host; it runs the user's application C/C++/FORTRAN compiled code, linked with Convey's libraries. The other process represents the Convey coprocessor; it runs a Verilog simulator such as Mentor Graphics' Modelsim, which simulates Convey's PDK Verilog plus the user's FPGA Verilog. The setup includes models of the coprocessor's memory subsystem, etc. Thus, the entire application can be simulated and debugged on any Linux machine.

3. Programming the Convey FPGAs with Bluespec's BSV Language

3.1 The BSV High-Level Language for Hardware Description

BSV (Bluespec SystemVerilog) is a modern high-level language for describing hardware at all levels of abstraction, from architectures to micro-architectures. BSV aims to advance the state-of-the-art from the two most commonly used hardware description languages (HDLs), Verilog, and VHSIC hardware description language (VHDL)[11,8], that date back to the 1980s. These languages significantly lag behind modern programming languages in structural abstractions such as types, encapsulation and parameterization. Their low-level behavioral semantics based on clocks (events), signals and wires, make them notoriously difficult to use on complex designs. Even recent evolutions of these languages, SystemVerilog and SystemC[9,10] are not significant improvements, particularly when restricted to their so-called "synthesizable subsets."

Some researchers and even some products today are attempting to use ANSI C or C++ to describe hardware computations, and to use so-called "High-Level Synthesis" to compile them into hardware[6]. This approach faces the same limitations encountered over the last 40 years in vectorizing compilers and automatic parallelization, namely the difficult problem of converting code written for sequential semantics and flat memory into an efficient parallel implementation. A deeper critique of this approach is that traditional sequential von Neumann languages are not very suitable for describing parallel algorithms or architectures.

The design of the BSV language, in contrast, is informed by two rich veins of prior art that seem more suitable for hardware design. The first is the functional programming language Haskell[19], and the second is the computational model variously known as Guarded Atomic Actions[7] or Term Rewriting Systems[12]. BSV uses Haskell's excellent facilities for types (polymorphic types, overloading, and strong type-checking), parameterization (including higher-order functions), and encapsulation (including modules and monads). BSV uses Guarded Atomic Actions to express all behavior. This computation model is well-known to be excellent for expressing complex concurrency (hardware!), and is found in several formal specification systems for complex concurrent systems, such as TLA+[13], UNITY[1], and Event-B[16].

For further reading on BSV, please see the journal article[17] or the book, "*BSV By Example*"[18].

3.2 BSV Abstractions for Interfacing with Convey Hardware Facilities

Figure 3 illustrates how Bluespec provides BSV interfaces for each of the Convey PDK Verilog interfaces. All the functionality of the Verilog APIs is available in a language with much higher-level semantics and strongly-typed interfaces.

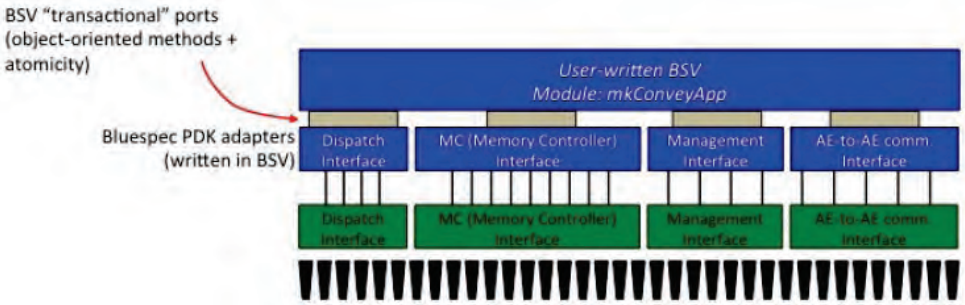


Figure 3. BSV abstractions for interfacing with Convey PDK

For example, the full interface required of the user's BSV FPGA code is specified by the following BSV interface type declaration:

```
// Top-level BSV interface for App

interface Convey_BSV_App_IFC;
    // ----------------------------------------------------------------
    // Dispatch Interface interface Convey_Dispatch_IFC dispatch_ifc;

    // ----------------------------------------------------------------
    // MC Interfaces
    interface Vector #(8, Convey_MC_Port_IFC) mc_e_ifcs;
    interface Vector #(8, Convey_MC_Port_IFC) mc_o_ifcs;

    // ----------------------------------------------------------------
    // AE-to-AE interface
    interface Convey_AE_to_AE_IFC ae_to_ae_ifc;

    // ----------------------------------------------------------------
    // Management interface
    interface Convey_Management_IFC management_ifc;
endinterface
```

By using vectors and sub-interface types, each of the lines for `mc_e_ifcs` and `mc_o_ifcs` represents 8×13 verilog ports (representing >1,700 wires), and correspond to Convey's eight "even" and "odd" memory interfaces. The sub-interface type `Convey_MC_Port_IFC` is specified as follows:

```
interface MC_Port_IFC;
    interface Get#(MC_rd_req)            get_rd_req;
    interface Get#(MC_wr_req)            get_wr_req;
    interface Put#(MC_rd_rsp)            put_rd_rsp;
    interface Get#(MC_flush_req)         get_flush_req;
    interface Put#(MC_flush_cmplt_rsp)   put_flush_cmplt_rsp;
endinterface
```

which completely captures the "object-oriented" semantics of communication in and out of the module. The types of payloads communicated are declared with conventional '`struct`' declarations like this one:

```
// Mem read requests
typedef struct {
    ConveyDataSize size;
    Vadr vadr;
    RdCtl rdctl;
    } MC_rd_req
    deriving (Bits);
```

Although describing hardware, these notations and semantics are familiar to anyone who has programmed in an object-oriented software language. In contrast, the descriptions of the Verilog APIs in the Convey PDK manuals take several pages of pictures of Verilog ports and textual descriptions of the ports and flow-control protocols.

BSV's Haskell-like types and functions enable a powerful "compositionality," as seen in Figure 4. The "ordering" module adds a layer to the basic memory interface, providing the same type of memory interface, except that its clients can rely on memory reads completing in order. This module encapsulates the infrastructure needed for managing memory ordering tags, so that the application need not worry about it. This functionality is optional because, of course, many applications (including the examples described later) can exploit out-of-order memory accesses for higher performance.

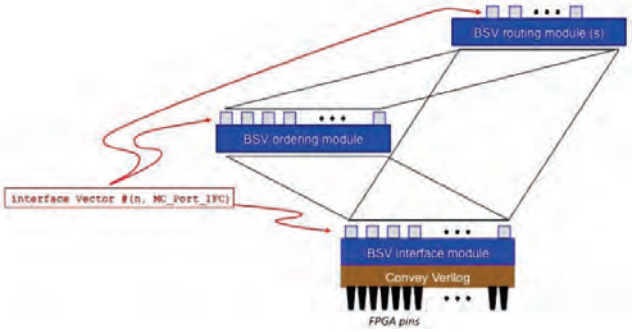


Figure 4. Composing interfaces

The "routing" module takes care of another issue for some applications. In the Convey hardware, memory addresses are interleaved across the eight memory banks and requests must be sent to the correct memory bank. For some applications with simple, regular data structures and predictable accesses, this targeting can be compiled in to the code. But for applications with dynamic and complex data accesses, memory requests must be routed to the correct memory channel, and this routing is performed by this module.

BSV's clean separation of module interfaces from module implementations allows easy substitution of the routing module with alternatives that have different performance (latency, bandwidth) and cost (FPGA area resources), from simple ring networks or "ring nets" to butterfly switches to full crossbars.

3.3 SW-HW Communication

Although the Convey scalar coprocessor is a fully programmable integer processor, for simplicity we use it in only a very minimal, fixed (application-independent) manner: to communicate one 64-bit value from the x86 to the FPGA, to start the FPGA computation, and to await its finish. Typically the 64-bit value communicated is a pointer to a data structure in shared memory that contains all the remaining arguments or results to be communicated. Communications during the computation are done through shared memory.

4. Flow, Debugging, Testing and Deployment

Figure 5 illustrates the BSV-specific aspects of the development cycles in creating an application for the Convey platform using Bluespec tools. C/C++ compilation, BSV compilation, and Verilog generation are very fast (few minutes), but unfortunately Xilinx synthesis can take several hours. Thus, much of the early development and debugging is done purely in simulation, prior to Xilinx synthesis and deployment on the actual Convey platform.

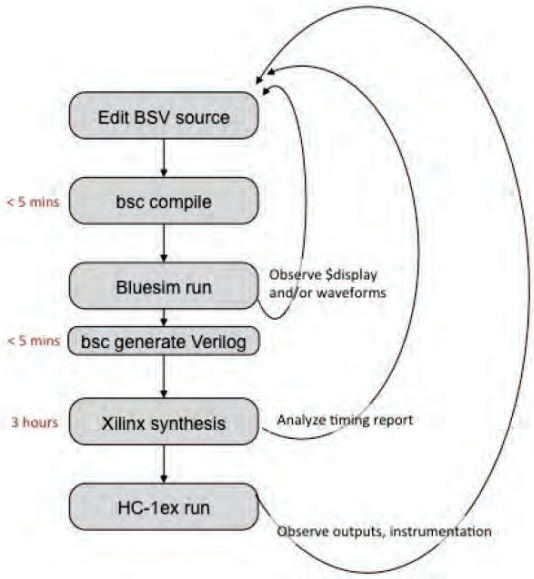


Figure 5. Development cycles

4.1 Bluesim Simulation

Bluespec's tools permit compilation of BSV source code into a native-compiled BSV simulation called "Bluesim," in addition to generating Verilog. Bluesim simulation is 100% cycle-accurate with respect to the Verilog, and indeed Bluesim can generate VCD files by which one can observe exactly the same waveforms as the generated Verilog. Further, BSV has an "import" capability by which C code can be embedded in the simulation. Based on these capabilities, Bluespec has created infrastructure to easily run a "whole-system" simulation for a BSV-based Convey application, i.e., containing the actual host x86 C/C++ code and the actual BSV FPGA code, just substituting emulated versions of the Convey host-FPGA communication and coprocessor memory. Figure 6 depicts a simulation and deployment of an application.

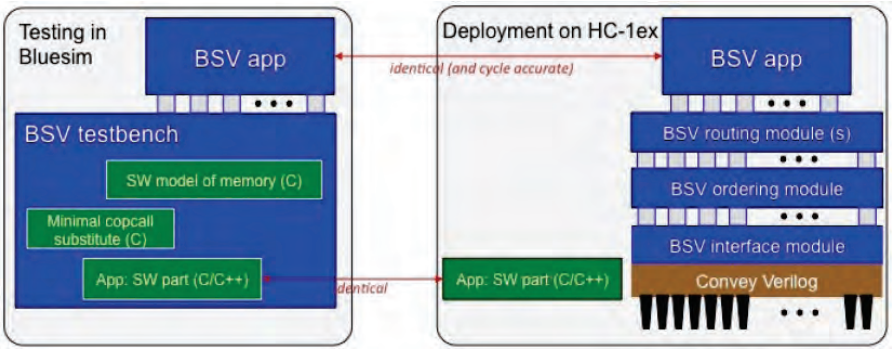


Figure 6. Bluesim simulation of the whole application (C/C++ and BSV)

One could, of course, generate Verilog from BSV code and use Convey's PDK simulation environment described earlier in Section 2.3, in order to debug and test the application. Apart from the disadvantage of also requiring a good Verilog simulator, in our experience this mode of simulation is several orders-of-magnitude slower than one based on Bluesim. For example, a small test on the DNA exact-matcher application ran in about one (1) second using Bluesim, but took about 90 minutes using the Convey PDK with Mentor Graphics' Modelsim, a speed difference of about 5000x.

Since Bluesim is also cycle-accurate, the Modelsim-based Convey PDK simulation did not give us any fundamentally additional detail. As a result, we now use Bluesim almost exclusively, going directly from there to deployment on the hardware.

4.2 Deploying a BSV Application on Convey Hardware

After testing, deployment on the Convey hardware is procedurally simple:

- Compile the C/C++ part of the application (usual C compilation)
- Generate Verilog from BSV using the Bluespec compiler (a single command)
- Synthesize the Verilog to an FPGA bitfile using the Xilinx synthesis tool; the Convey PDK has pre-packaged setup files for this synthesis (a simple "make" command)
- Deploy the application and the bitfile, which is just a matter of moving the compiled files to the appropriate directories on the Convey machine (invoke the application just like any other compiled program in Linux)

All these steps take seconds or minutes, except for Xilinx synthesis, which can take several hours.

Convey's PDK provides clocks and interfaces that run at 150 MHz, and thus it is preferable if one's BSV code also synthesizes at this speed. If Xilinx synthesis is unable to produce a bitfile that meets this speed requirement, one must iterate the development cycle (see Figure 5). In particular, a programmer carries out the following steps:

- Examine the Xilinx synthesis reports to identify the critical paths that fail to meet 150 MHz timing.
- Identify the BSV source structures corresponding to these critical paths. Even though Xilinx synthesis reports are in terms of signal and component names from the BSV-generated Verilog, it is always very easy to relate these to the BSV source code. Thus, we are never tempted to modify the generated Verilog, treating it as a purely machine-generated assembly language.
- Change the BSV source to address the timing issue, and repeat the cycle.

5. Two Examples

5.1 Simple Example: Vector-Add

This example is a good "warm-up" exercise. The Convey PDK includes a simple vector-add example, with the FPGA part written directly in Verilog. Our goal was simply to replace this hand-written Verilog with BSV-generated Verilog (reusing the C++ application driver).

The example performs an element-by-element addition of two vectors V1 and V2 into an output vector V3, and also sums up the results in V3. All elements are 64-bit integers. In other words,

$$\forall_{j=0}^{n-1} \quad V3_j = V1_j + V2_j$$

$$sum = \sum_{j=0}^{n-1} V3_j$$

In the specific example given in the Convey PDK, the input vectors are initialized as follows:

$$\forall_{j=0}^{n-1} \quad V1_j = j \quad \text{and} \quad V2_j = 2j$$

and thus the expected *sum* is 14,850.

By default, the coprocessor memory is configured with a simple binary interleave. We refer to each 512-byte row as a "stripe" across the eight memory controllers (MCs). Bits [8:0] of an address represent its offset within a stripe, bits [8:6] identify the MC it belongs to, and bits [5:0] represent its offset within the 64-byte chunk in that MC.

Figure 7 shows how the three vectors are laid out in this binary interleave. Specifically, note that we assume that all three vectors have the same starting offset within their respective stripes. The actual Convey PDK C++ application code in fact aligns all three vectors to offset 0 within a stripe (i.e., they are all at offset 0 in MC 0), but the BSV code can handle any stripe offset, provided they are the same for all three vectors. Note, the figure shows V1, V2 and V3 at increasing addresses, but this is not assumed. We do assume that V3 does not overlap V1 or V2, so that writes into V3 do not disturb V1 and V2.

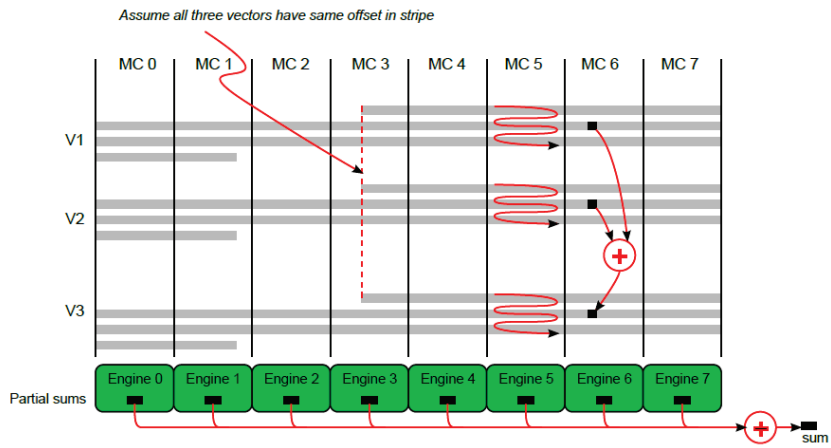


Figure 7. Memory layout of the three vectors in the vector-add example

The computation is performed by eight parallel "engines", each handling the part of the vectors that lie within one MC. Each engine does sequential traversals of its pieces of the vectors, as illustrated by the red squiggly arrows in Figure 7. At each point, it performs the + operation, and accumulates a partial sum. In the end, the partial sums are added to give the final sum.

In steady-state, on each clock, each engine performs two reads and one write to its memory channel. This actually over-subscribes the available bandwidth (two operations per clock), but that is okay, as the natural semantics of BSV rules will stall operations as needed. Reads from V1 and V2 are done on the even and odd ports, respectively. Writes are alternated between the even and odd ports.

The software sets up the hardware computation by communicating the base addresses of the three vectors, and their length, to the hardware. The software then issues a "CAEP0" coprocessor instruction, which is executed by the FPGAs to perform the vector add and sum. The final result sum is retrieved by the software from the FPGAs. (In the Convey PDK, and also in this BSV code, the computation is not divided across the four FPGAs; so this computation may be viewed either as done by just one FPGA, or replicated by all four FPGAs.)

The original Verilog for this application in Convey's PDK is 1,500 or more lines. The BSV code is about 300 lines.

5.2 Example: DNA Exact-Matcher

This example is actually the problem specified in this year's MEMOCODE design contest[15], and is illustrated in Figure 8. We are given a fixed-reference human genome, which is a string of 3 billion characters from the alphabet {A,C,G,T}. This input does not change from run to run, and so we can "pre-process" it to build fast lookup indices, etc. In each run, we are also given many (millions, hundreds of millions) of short "sequence-reads," which are strings of length

100 from the same {A,C,G,T} alphabet. The application's output reports, for each input sequence-read, how many exact matches exist in genome. If there are one or more matches, the application must also report the position in the genome of at least one of them.

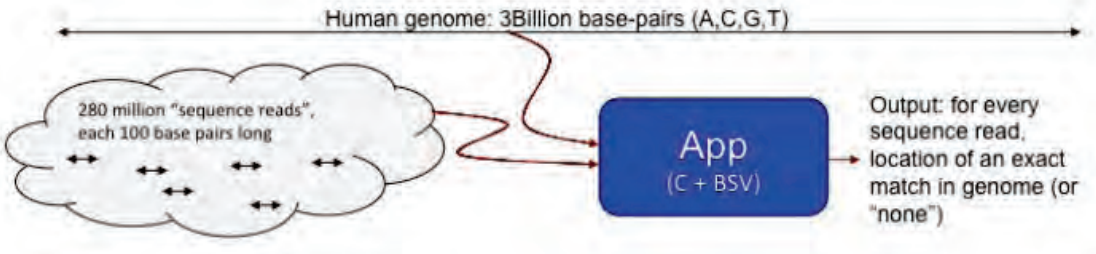


Figure 8. DNA exact-matcher application

5.2.1 Algorithm and Implementation

Langmead, et al., describes Bowtie, a state-of-the-art software DNA matcher[14]. It uses a technique called the Burrows-Wheeler Transform (BWT) for fast indexing into the genome. We used this technique in our matcher as well. In addition to the basic BWT table described in Reference 14, we created some two auxiliary tables to accelerate the algorithm. Overall, these tables occupy 6–7GB in memory.

Figure 9 shows the overall structure of our application. On the left is the C code, consisting of a number of pthreads. One of them simply sends the 64-bit pointer to a parameter block to the hardware. One of them places the 100-long sequence-reads from input files into four in-memory FIFOs feeding the four FPGAs. One of them retrieves match results from four in-memory FIFOs from the four FPGAs.

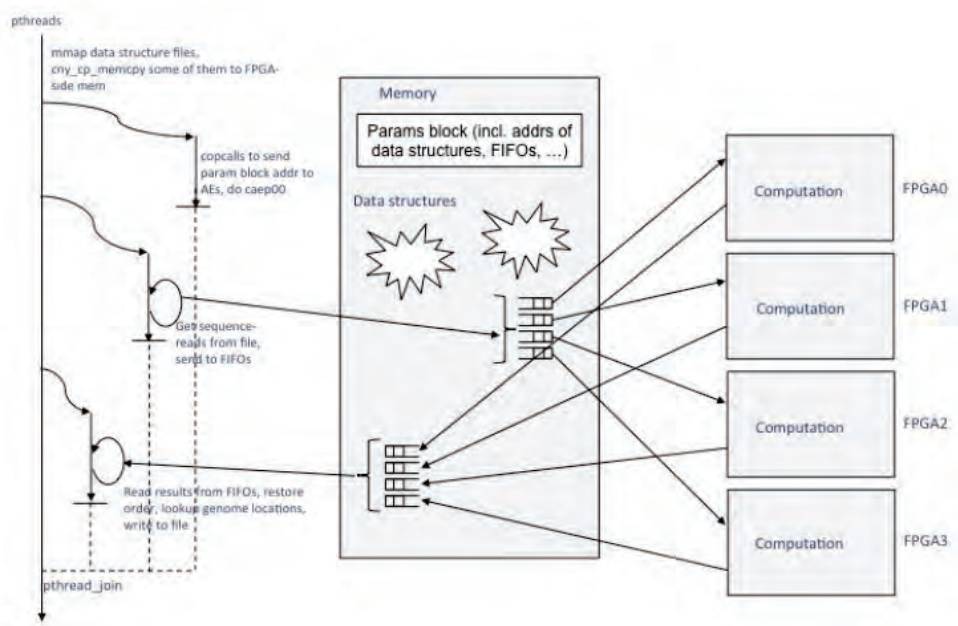


Figure 9. Overall C/BSV structure of the DNA exact-matcher

On the right are the FPGAs that compute the matches. The four FPGAs are loaded with identical programs (but each FPGA knows its identity, and therefore knows to look at specific input and output FIFOs). The internal structure of an FPGA program is shown in Figure 10. At the heart of the code are N identical, replicated "matchers" that perform the core BWT-based computation.

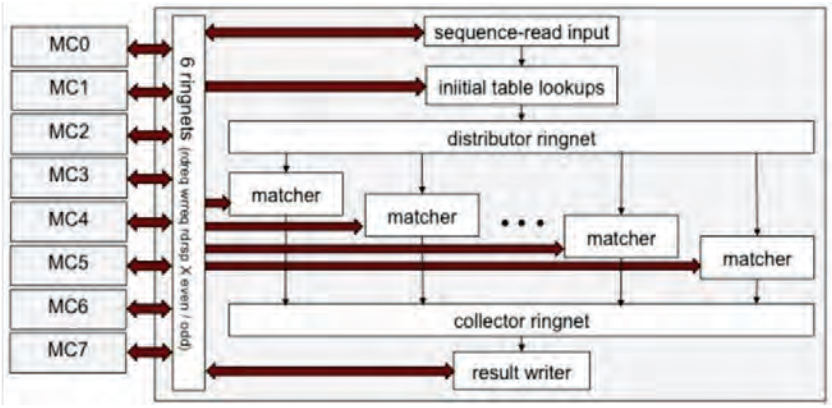


Figure 10. BSV structure in the DNA exact-matcher

Another interesting architectural feature evident in the Figure is the repeated use of ring nets for quite different purposes. Each of these is a little "NoC" (Network-on-a-Chip). The Distributor uses a 1-to-N ring net to distribute input sequence-reads to available matchers. Conversely, the Collector uses an N-to-1 ring net to collect results from the matchers, as they become available. On the left of the figure, we see six ring nets to route memory read- and write-requests from various modules (and various locations within modules) to the eight even and eight odd memory channels, and to route responses back. Each of these ring nets has eight endpoints on the memory channel side, but an application-specific number of endpoints within the application. In this particular case, the number of endpoints also depends on the choice of N, the matcher replication factor.

All these BSV ring nets are written with a single BSV source module, exploiting BSV's powerful parameterization facilities. The module's parameters not only include the number of input and output endpoints but also, borrowing from functional languages like Haskell, a higher-order function, that is the "routing function" for each packet. By simply customizing these, the same ring net can be used for all these instances.

Note that there are many "ordering" issues related to performance in this application. At the algorithmic level, the time taken by each matcher for each sequence-read is highly data-dependent. Thus, the Distributor cannot use a fixed round-robin policy to distribute inputs to matchers; it must be able to opportunistically provide an input to any matcher that becomes free after its previous task. Similarly, the Collector drains results as they become available from any matcher, which may not be in the original order.

In our solution, data are tagged by their original sequence position, and we rely on final C code on the x86 host to reorder outputs to their correct order. This approach allows the matchers to run at full-speed and never wait for any input. Further, the number of data accesses during a matcher task, and the addresses they are directed to, are highly data-dependent. By using dataflow tagging techniques, we allow these memory requests to happen in any order, as the hardware allows.

5.2.2 Results and Experience

Using BSV and the Bluespec tools, we were able to write this application from scratch and have it running in the Bluesim test environment in about 2–3 weeks (about 5 man-weeks). It took a few more weeks to deploy it successfully on an actual Convey machine, but the main roadblock was a Xilinx synthesis bug which was difficult to isolate. We hope that this was an abnormal one-time experience; without this we think the application would have been running in about 6–7 man-weeks.

During this period, we also tried many architectural variants.

- We changed the number of FIFOs feeding the matchers. Initially, we had one FIFO per matcher, but determined that one FIFO per FPGA is enough to feed all the matchers on that FPGA.
- We changed the number of matchers on each FPGA (this is a 1-line parameter definition change, followed by a recompile).
- We tried many variations of pre-fetchers and caches in the matchers
- We tried three variations on the initial Table Lookup module
- We varied the number of memory ring nets, to vary available bandwidth and latency.
- We inserted real-time status reports, monitored by another pthread on the software side, for debugging
- We inserted instrumentation to measure number of memory accesses and average memory-latency

Once running, we were able to run through the entire 284 million sequence-reads in about 47 minutes, i.e., at greater than 100,000 reads/second. This is very fast; even 10,000 reads/second is a respectable number. Our numbers are measured with an "N" replication factor of just two matchers per FPGA. We have enough room on the FPGAs to fit 20 matchers per FPGA, and we are now conducting that experiment, expecting even greater speeds.

The application contains 1,685 lines of C code and 2,400 lines of BSV code. The boilerplate code (non-application specific) includes 578 lines of C code for and 1,725 lines of BSV for the PDK interfaces and transactors. When compiled, the BSV code results in 39,500 lines of Verilog.

When run through Xilinx synthesis for the HC-1 (Virtex 5), the 2-matcher version occupies 61,976 LUTs (29% of available) and 71,006 Slice Registers (34% of available). We make negligible use of BRAM. This includes Convey PDK infrastructure, Bluespec BSV PDK infrastructure, and the BSV application. We did not have much difficulty meeting the target 150 MHz timing. In Section 5.2.1, we discussed the multiple ring nets used in our implementation. This is actually a symptom of a more general observation about parallel memory access. In this and other applications, we find that "memory routing networks" are a fundamental feature of many applications. When an application is implemented in software, all its memory accesses, no matter how separated in the code, ultimately are serviced in the same memory Load/Store unit in the CPU. But when implemented in hardware, the algorithm is spatially laid out on the FPGA fabric. Thus, the different memory accesses originate in different places on the FPGA. These requests must be routed to the correct memory channel port, and responses must be routed back. We make heavy use of dataflow tagging ideas to perform this function, as a result of which we are fully able to exploit the out-of-order, parallel, pipelined memory system on the Convey machine (we have been told that this memory system is a challenge to most other users!). BSV's powerful parameterization facilities make it relatively easy to implement such memory routing structures.

6. Conclusion

In the past, the need to write Verilog programs and do (slow) Verilog simulations has been a major obstacle in exploiting the architectural facilities of the Convey HC-1ex machine. In this paper, we have shown how complex applications can be developed relatively rapidly using the BSV high-level language and Bluespec's tools for fast simulation.

Acknowledgments

Jeff Newbern (Bluespec, Inc.) was an equal contributor in creating the DNA exact-matcher application. Convey answered technical questions about the platform, and provided remote access to Convey machines. Work was performed on the Convey HC1-ex system at NRL-DC under the auspices of the US Department of Defense (DoD) High Performance Computer Modernization Program (HPCMP).

References

1. Chandy, K. and J. Misra, *Parallel Program Design: A Foundation*. Addison Wesley, 1988.

2. Convey Computer, *Convey Reference Manual*, Version 2.0 900-000001-000, 2009.

3. Convey Computer, *Convey Hybrid-Core Computing*, CONV-10-017.1, 2010.

4. Convey Computer, *Convey Programmers Guide*, Version 1.8 901-000003-000, 2010.

5. Convey Computer, *Convey Personality Development Kit Reference Manual*, Version 5.0 900-000002-000, 2011.

6. Coussy, P. and A. Morawiec (eds.), *High Level Synthesis*, Springer, June 2008.

7. Dijkstra, E.W., "Guarded Commands, Nondeterminacy and Formal Derivation of Programs." *Comm. of the ACM*, 18(8), pp. 453–457, August 1975.

8. IEEE, *IEEE Standard VHDL Language Reference Manual*, IEEE Std 1076-1993, 2002.

9. IEEE, IEEE *Standard for System Verilog—Unified Hardware Design, Specification and Verification Language*, IEEE Std 1800-2005, 2005.

10. IEEE, *IEEE Standard SystemC Language Reference Manual*, IEEE Std 1666-2005, 2005.

11. IEEE, *IEEE Standard Verilog (R) Hardware Description Language*, IEEE Std 1364-2005, 2005.

12. Klop, J., *Term Rewriting Systems, volume 2*, Oxford University Press, pp. 1–116, 1992.

13. Lamport, L., *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*, Addison-Wesley Professional (Pearson Education), 2002.

14. Langmead, B., C. Trapnell, M. Pop, and S.L. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." *Genome Biology*, 10(R25), 2009.

15. MEMOCODE, *Design contest*, http://memocode.irisa.fr/2012/MEMOCODE%202012.html, 2012.

16. Metayer, C., J.-R. Abrial, and L. Voisin, *The Event-B Language*, rodin.cs.ncl.ac.uk/deliverables.htm, May 31 2005.

17. Nikhil, R.S., "Abstraction in Hardware System Design." *Communications of the ACM*, 54(10), pp. 36–44, October 2011.

18. Nikhil, R.S. and K.R. Czeck, *BSV by Example*, CreateSpace (Amazon.com), December 2010.

19. Peyton Jones, S. (ed.), *Haskell 98 Language and Libraries: The Revised Report*, Cambridge University Press, haskell.org, 2003.

Computational Analysis for Air/Ship Integration: Final Report

Susan A. Polsky
US Naval Air Warfare Center, Aircraft Division (NAWCAD), Patuxent River, MD
susan.polsky@navy.mil

Abstract

This paper documents the accomplishment of a three-year Grand Challenge Project focusing on the application of computational fluid dynamics (CFD) to predict coupled ship and aircraft aerodynamics. Unstructured overset techniques were used to simulate the coupled ship and aircraft systems. Dynamic aircraft maneuvers were prescribed with the intention of building to simulations with an autopilot in the loop. All simulations were computed in a time-accurate fashion due to the unsteady nature of the flow-field and used the commercial flow solver Cobalt. Analyses for both vertical shipboard landings of the Joint Strike Fighter and rotary-wing aircraft are discussed.

1. Introduction

While many aspects must be taken into consideration to ensure safe shipboard flight operations, a primary factor is evaluation of turbulent airwake effects on aircraft performance and pilot workload. The airwake is a product of wind passing over ship structures creating non-uniform, turbulent air flow. The US Navy conducts shipboard dynamic-interface (DI) testing to evaluate ship airwake effects on aircraft operations. These tests result in wind-over-deck (WOD) flight envelopes that prescribe in what wind conditions an aircraft can or cannot fly. The WOD flight envelopes are part of the operating procedures for all ship-based aircraft. Testing is required to generate WOD envelopes for each model of air vehicle operating from a given ship. The DI tests are performed at-sea, typically over the course of several weeks.

Application of computational fluid dynamics (CFD) methods to predict the turbulent ship airwake has been studied in the past with considerable success [References 1–4]. This has resulted in the use of CFD as an analysis tool to "diagnose" airwake structures that may impact air operations for both current and future ship designs. These diagnoses are accomplished by linking stored CFD generated airwake data with offline aircraft models controlled by either a pilot model or some other autonomous controller. For this "one-way-coupled" approach, the airwake data is imposed on the aircraft model; however, the presence of the aircraft does not feedback into the airwake data. While this approach has proven very useful, there are limitations to its applicability. When employing CFD data generated from a ship in isolation, the underlying assumption is that the presence of the aircraft will not affect the airwake from the ship structures. In the case of a small uninhabited air vehicle (UAV) this is likely a valid assumption; however, for an aircraft that produces a large wake of its own (such as a helicopter), this assumption becomes less and less valid as the aircraft comes in closer proximity to ship structures. The interaction of the ship airwake and the aircraft wake is generally referred to as ship/aircraft coupling. Aerodynamic coupling is a concern for both fixed-wing and rotary-wing shipboard operations.

Over the three years of the present Grand Challenge Project, efforts in the rotary-wing area were concentrated on analysis of a DDG destroyer and H-60 helicopter combination [Figure 1]. The motivation for the DDG/H-60 analysis is to understand where (in regards to proximity to a ship) aerodynamic coupling becomes important for rotary-wing vehicles. This knowledge, along with the coupled DDG/H-60 CFD data will be used to develop methods to account for aerodynamic coupling suitable for man-in-the-loop simulations (i.e., methods that run in real-time).

Fixed-wing research concentrated on Joint Strike Fighter (JSF) vertical landing and short takeoff from LHD-class ships [Figure 2]. Analyses were required to determine the impact of JSF outwash during vertical landings (VL) on critical (and costly) ship systems, either due to exposure to hot jet exhaust or due to the force of the jet exhaust. During an approach to a landing spot, the aircraft core nozzle passes near many large hull-mounted systems, such as radars and weaponry, and passes directly over the catwalk and flight deck. Although sub-scale experimental studies of the Short Take-Off/Vertical Landing (STOVL) outwash field were conducted, these experiments were for zero ambient wind speed with exhaust impingement on

flat plates. It was recognized that the air flow patterns would likely be affected by surrounding ship structures and prevailing wind speed and direction. Therefore, the CFD analyses included the ship hull, island superstructure, and many of the larger ship systems in the area of concern [Figure 3]. In the first two years of the Grand Challenge Project, the CFD analyses were used to help determine whether steps should be taken to move or shield deck edge equipment and replace it with instrumentation to collect data during flight tests at sea. The final year of the Grand Challenge concentrated on validating the CFD analyses against pressure and temperature data collected during at-sea testing conducted at the beginning of FY12.

Figure 1. An H-60 on approach to DDG-class ship

Figure 2. JSF VL approach to LHD-1

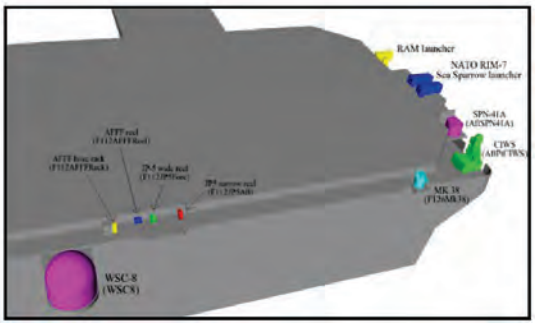


Figure 3. Stern & port mounted ship systems included in the initial CFD surface model

2. Technical Approach

The commercial CFD solver Cobalt, which was initially developed by the US Air Force under a CHSSI project, was used for this work. The solver employs a finite-volume, cell-centered scheme with flux vector splitting and unstructured, overset grid technology for bodies in relative motion [Reference 5]. All cases were run with second-order spatial and temporal accuracy. Past work has established that employing a monotone integrated large-eddy simulation (MILES) turbulence modeling approach is well-suited for the prediction of the large-scale turbulence found in ship airwake flow-fields. This approach was applied for the DDG/H-60 analysis discussed in this work. However, the MILES approach is not well-suited for JSF analysis due to the fine-scale turbulence which dominates the jet flow. Previous to this work, a validation study was completed comparing core nozzle exit-flow to sub-scale experimental temperature and pressure measurements at various distances below the nozzle exit. The results of this study indicated that applying a shear stress transport (SST) model was required to capture the core nozzle exhaust-flow with required accuracy. This analysis approach was extended to prediction of the lift-fan flow in the 2nd year and was found to accurately predict the thrust for the lift-fan. For coupled ship cases with no wind, using an SST model is not an issue. However, application of an SST model to aircraft/ship analyses with ambient wind presents a dilemma as it has been established that application of turbulence models to ship airwake flows damps out much of the higher frequency turbulence [Reference 2]. While it is thought that this higher frequency turbulence will not significantly affect the JSF outwash, more analysis is needed to confirm this.

The gridding and turbulence modeling approaches established by the core nozzle validation study were applied when modeling the complete JSF configuration including the core nozzle, lift-fan and roll nozzles. In the first year of this project, the lift-fan and roll jets were modeled using user prescribed boundary conditions at the nozzle exits. The boundary conditions were derived from high-fidelity CFD solutions of the isolated nozzles provided by Lockheed Martin.

Verification of the multiple nozzle flow interaction against sub-scale experimental data using this configuration was conducted and showed that the CFD was in the "ballpark" of the sub-scale data. However, unsteady characteristics in the CFD analysis pointed to issues with application of the user prescribed boundary conditions for the lift fan in particular. In the 2nd year of the project, internal components of the lift-fan and roll nozzles were included in the CFD model eliminating the prescribed exit boundary conditions. The improved geometric representation of the isolated lift-fan and roll nozzles provided accurate thrust perditions and were thus included in the complete JSF model. Verification against full-scale ship board data began in November of 2011 and is on-going.

Both the H-60 and JSF calculations employed overset-grids. All grids were generated using the NASA GridTool/VGRID system. Grid sizes for the DDG/H-60 analysis were on the order of 11.4 million and 3 million cells, respectively. The rotors (main and tail) were modeled as actuator disks and in some cases as blade elements. In the first year grid sizes for the LHD/JSF outwash analysis were on the order of 29 million and 3 million cells, respectively. With the addition of the lift-fan internal geometry, the grid sizes for the JSF have grown to over 15 million cells. Because aircraft performance predictions were not required, a relatively simple fuselage model was used for the JSF aircraft. This allowed for construction of much smaller grids than would be required if the detailed surface geometry was modeled. The JSF analysis employed overset-grids to model motion of the aircraft relative to the ship while the H-60 analysis employed overset- grids to integrate individual blade segment aerodynamic information generated by an external blade element model.

3. Accomplishments

3.1 DDG/H60

The first year of the Grand Challenge Project produced a database of over 88 coupled DDG/H-60 solutions [Figure 4]. The time-accurate data was saved and used as a database to explore the feasibility of applying proper orthogonal decomposition (POD) neural networks techniques to model the coupled aerodynamic environment in real-time, piloted simulations. This work is on-going and includes experimental elements in addition to the computational efforts.

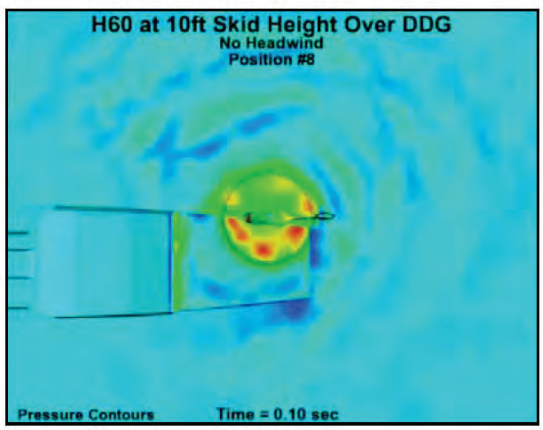


Figure 4. Example of simulation at one rotor location and wind speed (snapshot in time)

The second year of rotary-wing work concentrated on linking aircraft simulation models used at the US Naval Air Systems Command (NAVAIR) Manned Flight Simulator (MFS) at Patuxent River, Maryland with the Cobalt CFD flow solver, with the goal of "flying" the aircraft through the CFD simulation using an auto pilot. The coupled helicopter/ship analysis described above employed actuator disk modeling for the rotor. However, the real-time aircraft simulation model employs a blade element model. Overset-grid methods [Figure 5] were developed to track the blade locations predicted by the aircraft simulation model and fed to the flow solver. In addition to blade locations, pressures at each blade element were also provided by the aircraft simulation model to the flow solver [Figure 6]. Several test cases were completed [Figures 7 and 8] to verify that the method was sound, and all data was provided and read correctly by the flow solver. For these test cases, the aircraft simulation model was run locally (at Pax River) to produce a data file with the input data required for the flow solver boundary condition and overset-grid positioning. In the third year of the project, the aircraft simulation model has been ported to Raptor at the US Air Force Research Laboratory (AFRL). Currently, efforts are underway to enable the flow solver and A/C simulation model to pass data back and forth dynamically during a batch run.

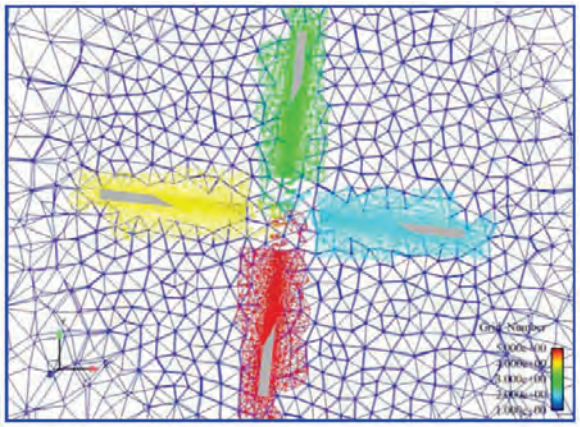


Figure 5. Blade element overset grids (red, cyan, green, yellow) with blue background grid

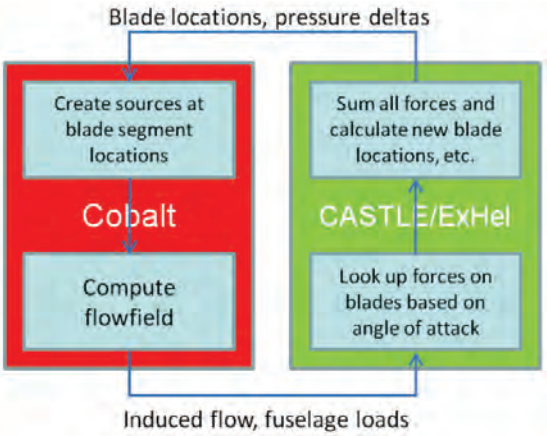


Figure 6. Block diagram showing Cobalt and CASTLE data paths

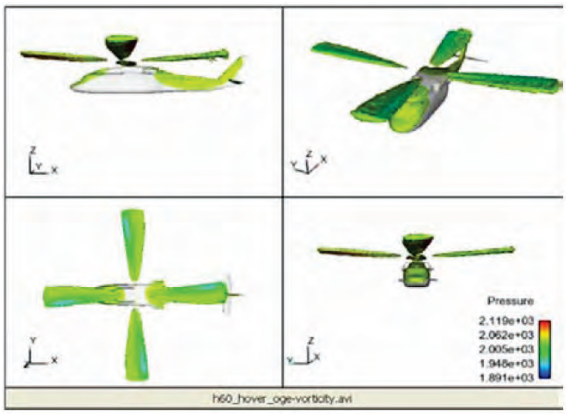


Figure 7. Iso-surface of vorticity at an early time step of Cobalt analysis using ExHel blade element data

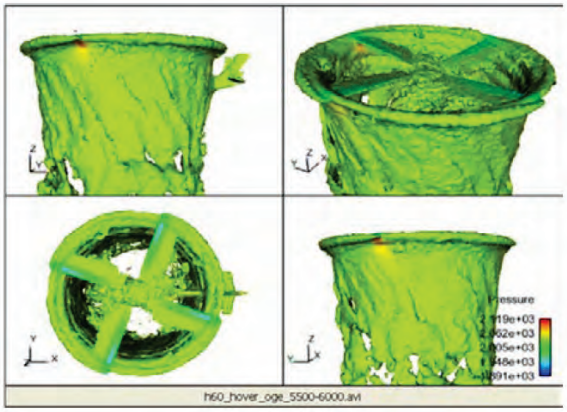


Figure 8. Iso-surface of vorticity at a later time step of Cobalt analysis using ExHel blade element data

Although a generic blade element model (ExHel) was used as the demonstration case for Cobalt/CASTLE integration, the ultimate goal is to use the integrated approach for a real air vehicle. A major difficulty in linking an aircraft like an H-60 in a non-real-time simulation is the need for a pilot to fly the simulation model. This requirement also exists for ExHel. In the 2nd year of this project, an autopilot was created to fly the ExHel simulation model. The pilot model was tested using the standard one-way airwake coupling approach. It is anticipated that the first fully-coupled auto-piloted CFD analysis will be completed by the end of the fiscal year.

The Fire Scout VTUAV [Figure 9] provides a unique opportunity because the same autonomous control system that flies the aircraft can be integrated in the simulation, obviating the need for a pilot model. Because a pilot is not required in the loop, the simulation need not run in real-time. A Fire Scout CFD fuselage surface model [Figure 10] was created in the 2nd year of the project, and predictions of fuselage drag have been compared to available wind tunnel data for multiple angle-of-attack and yaw configurations.

Figure 9. Fire Scout VTUAV above the flight deck of a DDG destroyer

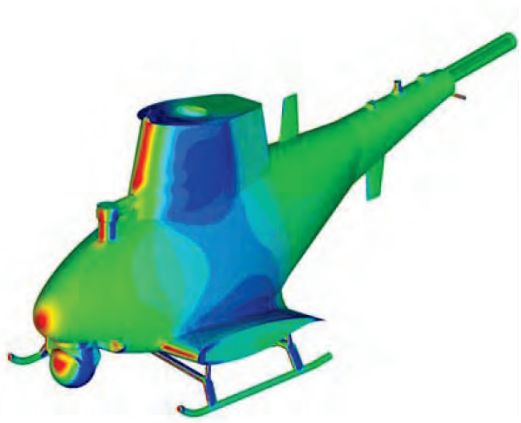


Figure 10. Example of CFD results for Fire Scout fuselage model

3.2 JSF/LHD

The first 2 years of the JSF work concentrated on prediction of A/C outwash during vertical landings and, to a lesser extent, short takeoffs. All dynamic approach simulations were conducted with prescribed approach paths and included a 90° approach (shown in red in Figure 11) and a 45° approach, and the deck roll portion of a short takeoff. Two landing spots were examined: spot 7 (forward spot shown in Figure 11) and spot 9 aft of spot 7. In the first year of the project, a generic LHD ship was used. However, in the 2nd year, a ship surface model specific to LHD-1 was created, and considerable detail added to the port-side deck edge [Figure 12]. Outwash data was recorded and provided to ship system subject matter experts as input to ship design/change decisions.

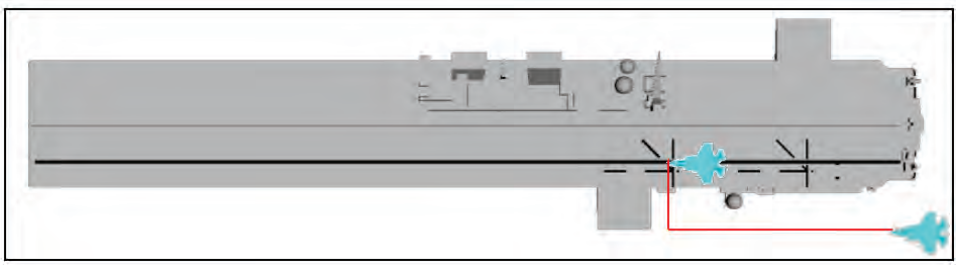


Figure 11. Example of a vertical landing approach path (red) on generic LHD ship model

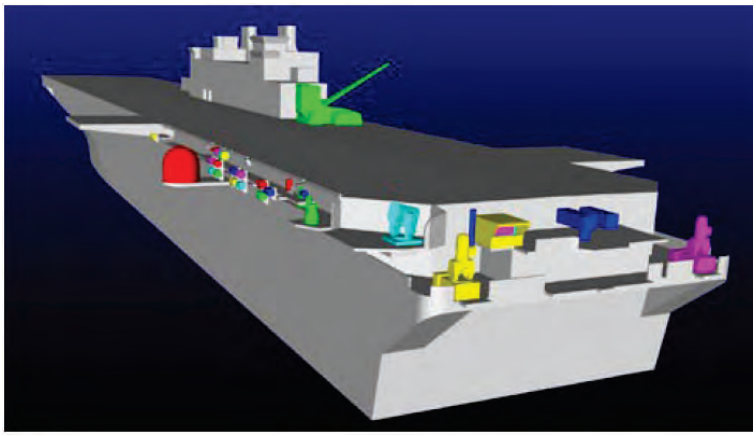


Figure 12. Refined LHD surface model

Although a typical STO consists of three distinct phases: deck roll, rotation, and fly away, only the deck roll portion has been modeled thus far. The motion is again prescribed and models the typical aircraft accelerations and speeds. A snapshot from typical STO case is shown in Figure 13. The jet exhaust is visualized by surfaces of constant velocity.

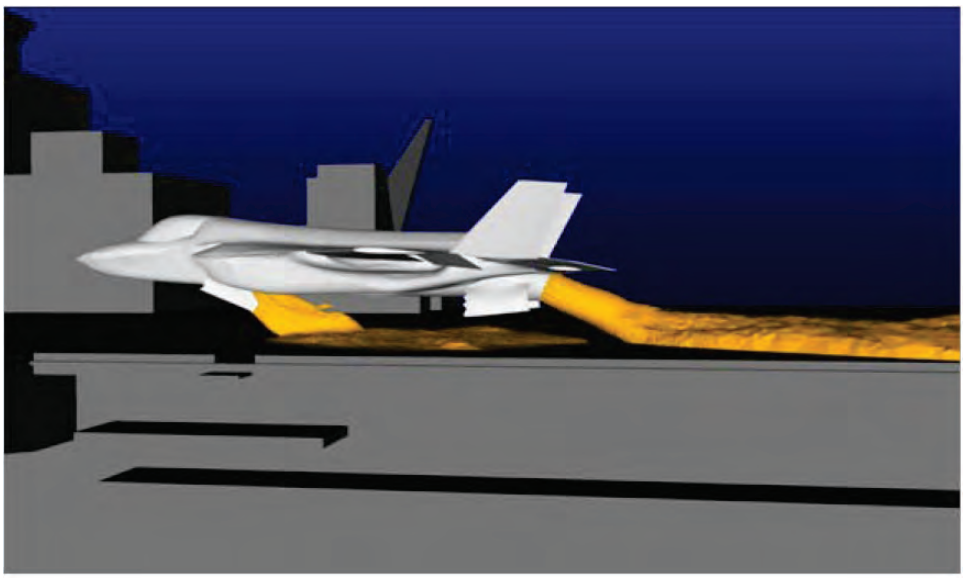


Figure 13. A snapshot in time of short take-off simulation. Jet exhaust and outwash visualized by velocity iso-surface.

In the 3rd year, the ship model was again refined to be representative of the specific configuration of LHD-1 during JSF DT testing [Figure 14]. Aircraft motion data recorded during at-sea testing was used to drive the JSF CFD model relative to the ship. Ambient wind conditions recorded at-sea for each of over 15 vertical landings was applied as boundary conditions for the CFD simulations. Typically, thirty seconds of the approach and landing were modeled; however, some cases required either more or less time. Most landings were modeled using the approach developed during the first two years of the project. However, discrepancies found when comparing the CFD results to recorded temperature and pressure [Figure 15] time histories prompted a study of time-step size and turbulence modeling. While final results cannot yet be presented, the time-step and turbulence model study demonstrated the need to tailor the modeling approach depending on the dynamic nature of the flow-field. It is hoped that final results from this study may be published in the near future.

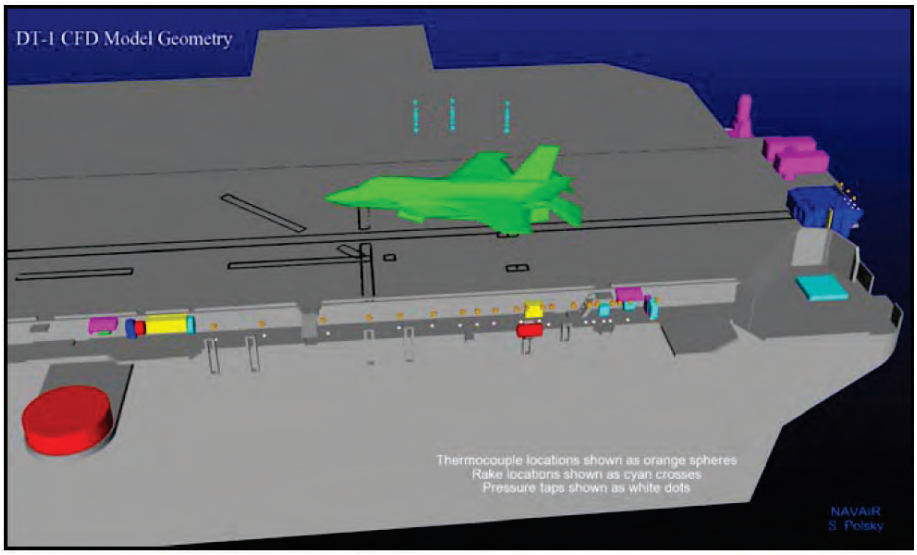


Figure 14. Pressure and temperature experimental instrumentation locations for CFD validation data

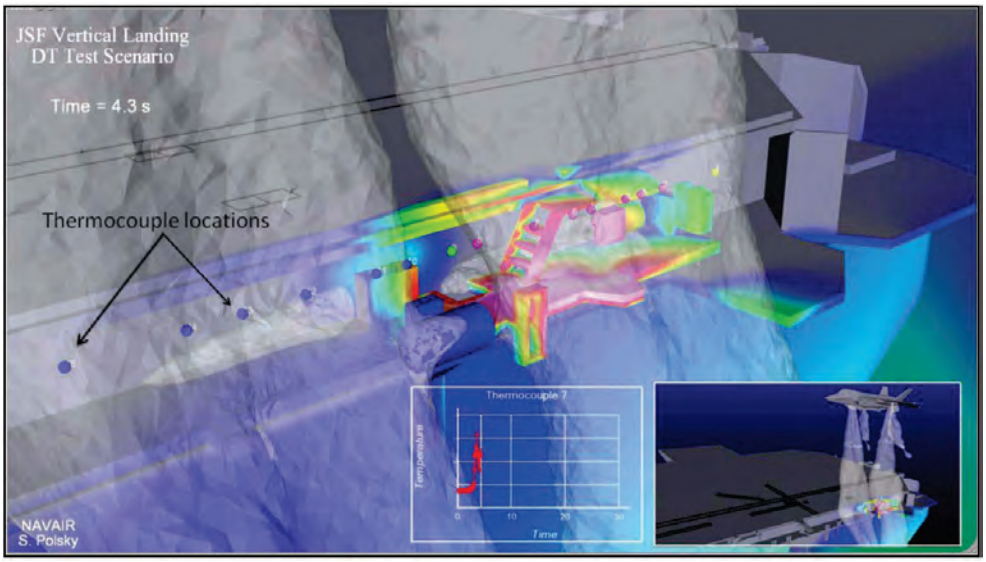


Figure 15. A snapshot in time of dynamic approach simulation. Jet exhaust and outwash visualized by velocity iso-surface. Thermocouple locations shown as spheres colored by temperature.

Over the three years, JSF simulations were run on AFRL Hawk & Raptor, MHPCC Mana, and ARL Harold. Calculations typically utilized 256 processors and required approximately 20,000–50,000 CPU-hours each depending on the grid-fidelity and length of dynamic maneuver.

4. Significance to the DoD

Rotary-wing ship integration modeling and simulation research funded through the Office of Naval Research CASSIA program is paving the future in dynamic interface modeling and simulation. Results from this work will directly contribute to improving ship design, ship-based aircraft control system design, and flight control design for piloted and autonomous air vehicles. Numerical algorithms and physical models developed through this program will be transitioned to the CREATE program to benefit the CFD community throughout the DoD.

Techniques and methodologies developed through this project will be directly applicable to the JSF program during developmental testing (DT), operational testing (OT), and fleet support to help ensure safe operation of the aircraft aboard ship. Successful application will require the development, validation, and application of CFD techniques related to coupled aircraft and ship airwake prediction. Inherent in the development and validation of this technology are important science and technology areas such as vortex modeling, large-eddy simulation, grid-generation, parallel code development, CFD boundary condition development, and temporal accuracy issues. These developments will also benefit other DoD aircraft and ship programs, such as V-22, VTUAV, CVN-78, and DDG-1000.

The military advantages gained by exploiting HPC capability are related to design and testing. This analysis will enable ship integration of the JSF system in such a way that minimizes potential damage to ship structure and systems and helps to ensure safety of personnel.

Acknowledgements

The author would like to thank: the Office of Naval Research and the Joint Strike Fighter Program for their financial and professional support and the High Performance Computing Modernization Program for their generous allocation of computer time, without which this work would not have been possible.

References

1. Polsky, S. and Bruner, C.W., "Time-Accurate Computational Simulations of an LHA Ship Airwake." *AIAA Paper 2000-4126*, 2000.

2. Polsky, S., "Computational Study of Unsteady Ship Wake." *AIAA Paper 2002-1002*, Jan. 2002.

3. Polsky, S., "CFD Prediction of Airwake Flowfields for Ship Experiencing Beam Winds." *AIAA Paper 2003-3657*, Aug. 2003.

4. Woodson, S. and Ghee, T., "A Computational and Experimental Determination of the Air Flow Around the Landing Deck of a U.S. Navy Destroyer (DDG)." *AIAA Paper AIAA-2005-4958*, June 2005.

5. Strang, W.Z., Tomaro, R.F., and Grismer, M.J., "The Defining Methods of Cobalt_{60}: A Parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver." *AIAA-99-0786*, January 1999.

Effect of Nozzle-Exit Turbulence Level on the Flow and Acoustic Properties of Imperfectly Expanded Supersonic Jets

J. Liu, K. Kailasanath, and J.P. Boris
US Naval Research Laboratory (NRL-DC), Washington, DC
{jhliu, kailas, boris}@nrl.navy.mil

N. Heeb, D. Monday, and E. Gutmark
University of Cincinnati, Cincinnati, OH
{heebns, mundaydd, gutmarej}@ucmail.uc.edu

Abstract

The impact of the turbulence level at the nozzle exit on the jet flow and far-field noise level has been investigated using large-eddy simulations (LES) and the Ffowcs Williams & Hawkings (FW-H) surface integral method. The jet exit condition is slightly under-expanded. Both the random perturbations inside the nozzle and the nozzle surface roughness are used to increase the turbulence level. Increasing the turbulence level at the nozzle exit increases the shear-layer spreading, reduces the screech intensity and also decreases slightly the far-field noise level. The LES predictions of the shock-cell structures and the far-field predictions agree well with the measurement data. In addition, the impact of grid resolution used in both the jet flow and the near-field acoustic propagation region has also been investigated. It is found that the grid resolution used in the shear layer impacts the numerical predictions more than those used in the other regions, and an adequate amount of turbulence level at the nozzle exit greatly improves the predictions using a coarser grid resolution in the shear layer. This work involves a large number of simulations, which are very CPU-intensive because of the large number of grid points and the large number of time steps needed for simulating and processing the acoustic properties. Thus, the computational resources provided by the High Performance Computing Modernization Program Challenge Project C5G were critical for completion of this research.

1. Introduction

There is a growing need to reduce significantly the noise generated by high performance, supersonic military aircraft. The noise generated during takeoff and landing on aircraft carriers has direct impact on shipboard health and safety issues. It is important to understand the mechanism of the noise generation under these operating conditions to develop effective noise reduction techniques. The exhaust from the engines tends to be imperfectly (under/over) expanded during takeoff and landing, and the nozzle exit conditions may directly or indirectly impact the jet noise generation.

Instability waves originating from a nozzle lip in a jet flow roll up into coherent structures in the shear layer, impacting shear-layer spreading, jet core length, peak turbulence level and associated noise generation. Since the nozzle exit conditions impact both the initial instability waves and the coherent structures, differences in the initial conditions may result in a large scatter of measurement data[1–4]. Among the variables that define the nozzle exit conditions, the initial turbulence level is found to have a dramatic effect, and may have been the primary reason for discrepancies reported in literature[2]. Increasing the turbulence level at the nozzle exit is found to increase shear-layer spreading and weaken the coherent vortex paring[4]. In imperfectly expanded supersonic jet flows, screech tones with large intensities also dramatically impact the jet flow[5]. The turbulence level at the nozzle exit can inhibit the formation of the screech tones and also alter the level of shock-cell associated noise[3]. Therefore, it is important to take into account the nozzle exit conditions, especially the initial turbulence level, to assess data taken from different facilities, such as laboratory model tests or full-scale tests, and data taken from different approaches, such as numerical simulations or experimental measurements.

In this paper, we report on an investigation of the impact of the turbulence level at the nozzle exit on the jet flow development and on the noise characteristics of a slightly under-expanded jet flow using large-eddy simulations (LES). In the past, we have studied the flow and noise properties of imperfectly expanded jets emanating from a converging-diverging (C-D) nozzle that is similar to those used in military aircraft. The low inflow turbulence level at the nozzle exit

in the simulations caused a slower shear-layer spreading[7]. To reduce the differences of initial conditions between simulations and measurements, techniques controlling the initial turbulence levels are sought. Randomly perturbing pressure or velocity field has been used to increase the turbulence level at the nozzle exit in simulations of jet flows[6,9–11]. For example, randomly perturbed pressure has been used by Bogey and Bailly in their simulations of subsonic jets[4] and Cacqueray, et al. in their simulations of an over-expanded heated jet[8]. Nozzle geometries were not included in those simulations, so analytical velocity and temperature profiles were used at the nozzle exit. In addition, randomly perturbed velocity field with some physical constraints have also been used in simulations of subsonic jet flows[9,10]. In this paper, we present results using random perturbations of the energy field inside the nozzle as well as using nozzle surface roughness to control the turbulence level at the nozzle exit. The advantage of the first approach is its simplicity, and the advantage of the second approach is that the generation of turbulence is a solution of the flow equations.

2. Nozzle Geometry and Methodology

The unsteady three-dimensional, compressible flow equations are solved with a finite element code FEFLO on unstructured tetrahedral meshes. The Flux-Corrected Transport algorithm[11] is used for the spatial discretization and a second order Taylor-Galerkin scheme is used for time integration. A non-clipping flux limiter synchronizing the minimum of the limiters obtained for the density and energy is used[12]. No explicit subgrid scale model is used and thus the present simulations are in the category of monotonically integrated large-eddy simulations (MILES). FEFLO is a well-validated and fully parallelized production code, and the parallelism either uses the pure Message Passing Interface (MPI) or the mixed MPI/Open MP options. Table 1 shows the speedup of a simulation using a mesh of 348 million elements with the mixed MPI/Open MP options on a distributed memory quad-core Intel Xeon Nehalem-EP cluster (diamond) at the US Army Engineer Research and Development Center (ERDC). The computational domain is split to 128 sub-domains, and 128 nodes are used. It can be seen that scalability is satisfactory.

Table 1. Speedup study of FEFLO with a mesh of 348 millions elements

CPU core#	CPU time (s)/per core/per step/per million element	Speedup	Element per core (million)
128	0.0149	1.0	2.72
256	0.00535	2.79	1.36
512	0.00300	5.05	0.68
1024	0.00187	8.0	0.34

The nozzle geometry shown in Figure 1 is the same C-D nozzle geometry reported in Reference 7. It is representative of practical military engine nozzles, which do not have smoothly varying contours designed by the method of characteristics to produce shock-free jet flows at the design condition. The design Mach number of this nozzle is 1.5 and the fully expanded nozzle pressure ratio is 3.7. The inflow total temperature is the same as the ambient temperature, and the nozzle pressure ratio presented in this paper is 4.0, which is slightly under-expanded. Since the Reynolds numbers in the current simulations are high ($>10^6$), the nozzle boundary layer is very thin and the direct simulation of such thin boundary layers using an explicit temporal scheme would require a very small time step. To circumvent this problem, the nozzle boundary layer is not directly implemented and the free slip boundary conditions are used on the wall. This may result in a very low turbulence level at the nozzle exit, but it allows us to study the impact of the turbulence level at the nozzle exit from a very low level to the maximum level that we would like to investigate. The roughness pattern used in simulations is also shown in Figure 1. These bumps are randomly distributed and the height is less than 0.2% of the nozzle exit diameter. In addition, Ffowcs Williams & Hawkings (FW-H) surface integral method is used to predict the far-field noise[14].

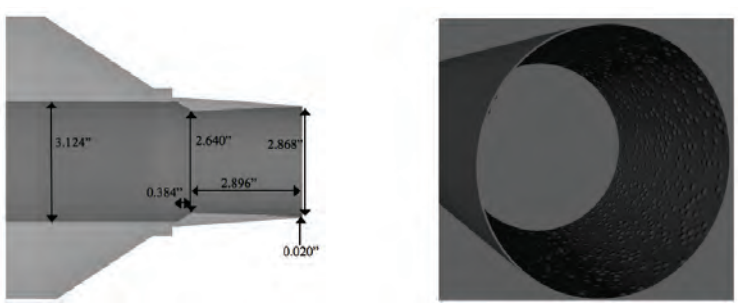


Figure 1. The nozzle geometry is shown on the left, and the surface roughness pattern is shown on the right

Grid resolution tests have been carried out in three major computational regions, which include the shear layer, the downstream jet wake region, and the near-field acoustic propagation region. The cell sizes and the number of grid points of each mesh are shown in Table 2. Meshes I, II, and III are used to test the grid resolution in the flow region, and Meshes II, IIb, and IIc are used to test the grid resolution in the region around the FW-H surfaces. Two cell sizes are tested in the shear layer and the downstream jet wake region, whereas three cell sizes are tested in the near-field acoustic propagation region. The axial variations of the cell sizes from the shear layer to the downstream are shown in Figure 2. In the radial direction, the jet-wake grid resolution of either D/30.0 or D/45.0 extends to 1.1D above the jet center at the nozzle exit and gradually extends to 1.8D at x=10.0D and maintains this constant radius up to 25.0D downstream. The mesh size then increases slowly to the size used for the near-field region, beyond which the size increases rapidly to a very large size in the far field.

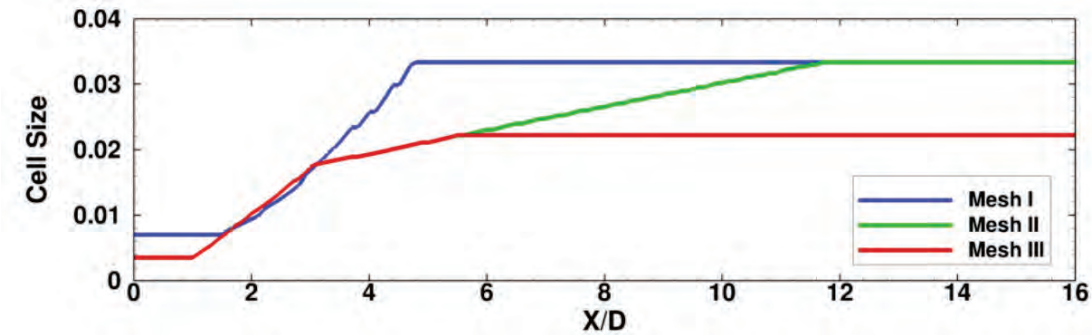


Figure 2. Mesh size distributions along the axial direction for meshes I-III shown in Table 2

Table 2. Cell sizes and numbers of grid points of meshes (D is the nozzle exit diameter)

Mesh	Shear layer	Jet wake	Near field	Number of grid points	Number of Elements
Mesh I	D/143.0	D/30.0	D/15.0	28M	165M
Mesh II	D/286.0	D/30.0	D/15.0	42M	240M
Mesh IIb	D/286.0	D/30.0	D/20.0	59M (larger FW-H surface)	348M
Mesh IIc	D/286.0	D/30.0	D/30.0	70M	389M
Mesh III	D/286.0	D/45.0	D/20.0	66M	410M

3. Results and Discussions

Because it is easier to change the turbulence level by adjusting the perturbation level inside the nozzle than by adjusting the magnitude of the wall surface roughness, the impact of the initial turbulence level has been studied mostly through the first approach, in which perturbations of a small percentage of the local energy are added near the nozzle expansion surface. These perturbations are random in time and in space. A simulation using the wall roughness has also been carried out using Mesh II for comparison.

Table 3. Turbulence levels near the nozzle lip and shear-layer momentum thicknesses

	Mesh I				Mesh II					Exp.[15]
Pert. Level	0.0%	0.5%	1.0%	2.0%	0.0%	0.25%	0.5%	1.0%	Roughness	
TKE/U_j	0.4%	0.85%	2.1%	3.0%	0.85%	1.2%	3.2%	7.0%	2.0%	
θ/D@x=0.0D	0.010	0.0095	0.012	0.012	0.0087	0.0087	0.0088	0.0115	0.0088	
θ/D @x=0.72D	0.019	0.019	0.02	0.024	0.017	0.02	0.025	0.034	0.02	0.02
θ/D @x=1.92D	0.027	0.034	0.051	0.054	0.047	0.050	0.053	0.07	0.053	0.049

3.1 Results of the Flow Field

The turbulence levels near the nozzle lip introduced by these two approaches are shown in Table 3 along with the shear-layer momentum thicknesses at several axial locations. TKE is the turbulent kinetic energy, U_j the fully expanded jet velocity and θ the momentum thickness. The measurement results with the same nozzle size and similar operating conditions have been previously published in Reference 15. It can be seen that the turbulence level increases with the

perturbation level, and the coarser grid resolution requires more perturbations to get the same level of turbulence. The turbulence level introduced by the wall roughness is between the perturbation levels of 0.25% and 0.5%.

Figures 3–4 show the instantaneous density and pressure distributions using Mesh IIb, with a small inflow perturbation level of 0.25%. It can be seen that large pressure waves radiate from the end of the potential core, which is located roughly at x=9.0D~10.0D, as shown in Figure 3. In addition, some pressure waves with finer scales also radiate from the nozzle exit. The locations of three FW-H surfaces are also plotted in Figure 4. These surfaces are angled at 5.3°, 7.4°, and 9.5°, respectively, to the axial axis after the nozzle exit and they extend to 2.0D upstream of the nozzle exit to 27.0D downstream. The radius at the nozzle exit and upstream is 1.5D above the jet center. Mesh IIb is used to study the impact of the location of FW-H surfaces, so the region with the near-field grid resolution covers the largest FW-H surface. However, to reduce the number of grid points and to speed up the test, the near-field region of other four meshes only covers the smallest FW-H surface.

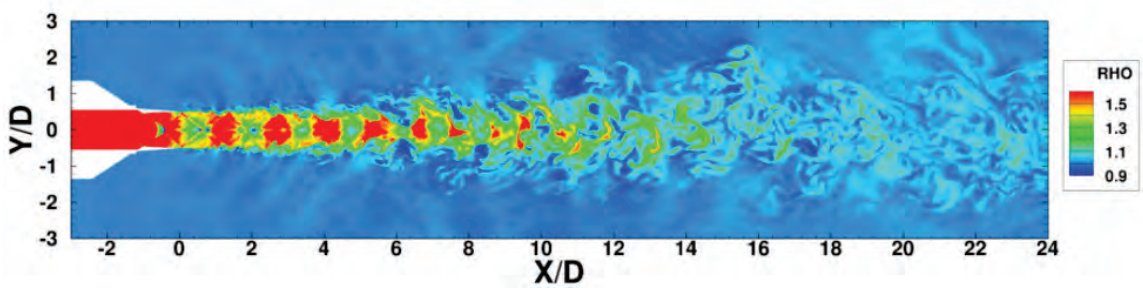


Figure 3. Instantaneous density distribution of a slightly under-expanded jet condition (small random perturbations with 0.25% of the local energy values are added to the energy along the diverging section near the nozzle surface)

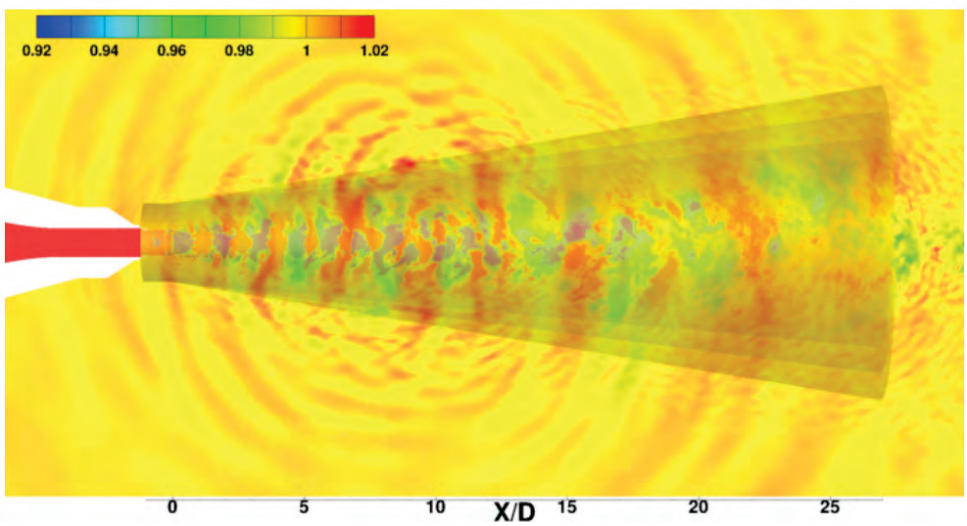


Figure 4. An instantaneous pressure distribution at the jet condition mentioned above. The three surfaces shown in this figures are the FW-H surfaces used in this paper. These three FW-H surfaces are at an angle of 5.3°, 7.4°, and 9.5°, respectively, to the axial axis downstream of the nozzle exit, and the axial range is from 2.0D upstream of the nozzle exit to 27.0D downstream. The radius at the nozzle exit is 1.5D for all the surfaces, and the region with this radius extends to 2.0D upstream of the nozzle exit.

The time step size is roughly 0.0006 D/U_j for Mesh I and 0.0003 D/U_j for the other four meshes. After the flow has reached the fully developed status, it requires at least 500,000 time steps for Mesh I and 1,000,000 time steps for other four meshes to cover a time length of roughly 300 D/U_j to provides a minimum number of time samples for computing the near- and far-field sound pressure levels. Therefore, Mesh I would require around 100, 000 CPU hours on a quad-core Xeon Nehalem cluster to obtain a minimum number of time samples for the evaluation of the sound pressure levels, whereas, Mesh II would require twice more CPU hours because of the increased number of grid points and a reduced time step size. The CPU hours needed for other three meshes would be even greater. In addition, since each mesh may involve several simulations to test the impact of the turbulence levels at the nozzle exit, a large number of expensive simulations are involved in this work, and we greatly appreciate the computational resources provided by the High Performance Computing Modernization Program (HPCMP) Challenge Project C5G.

Figure 5 presents comparisons of axial velocities between PIV measurement[15] and LES predictions using the finer shear-layer grid resolution (Mesh II) with either the inflow perturbation level of 0.25% or the wall roughness. It can be seen that the agreement of the shock-cell structure is excellent and the shear-layer spreading is also comparable in both cases. The responses of the momentum thickness to the turbulence level using this mesh and mesh I that has a coarser grid resolution in the shear layer can be seen in Figure 6 and also in Table 3. It can be seen that increasing the turbulence level at the nozzle exit increases the shear layer spreading, but the coarser grid resolution (Mesh I) needs more perturbations to increase the turbulence level at the nozzle exit, and also needs more turbulence to produce a similar shear layer growth. The coarser grid resolution, however, is able to produce a shear-layer growth rate similar to what is observed in the measurement at a perturbation level of 1.0% or above. On the other hand, the finer grid resolution (Mesh II) can produce a similar shear-layer growth rate without adding perturbations. This finer grid resolution should have captured more fine turbulence scales and thus increases the turbulence level in the shear layer. Overall, the prediction with a small percentage of perturbations (0.25%) has the best agreement with the measurement, and the results using the wall roughness also agree well. It should also be pointed out that the correlation between the perturbation level and the turbulence level at the nozzle exit may vary with how the simulation is performed because it not only depends on the grid resolution, as shown in Figure 6 and Table 3, but also depends on the methodology used in flow solvers and the computational approaches used inside the nozzle. Thus, the magnitude of the perturbation level is not critical, but how much turbulence level it generates is more important.

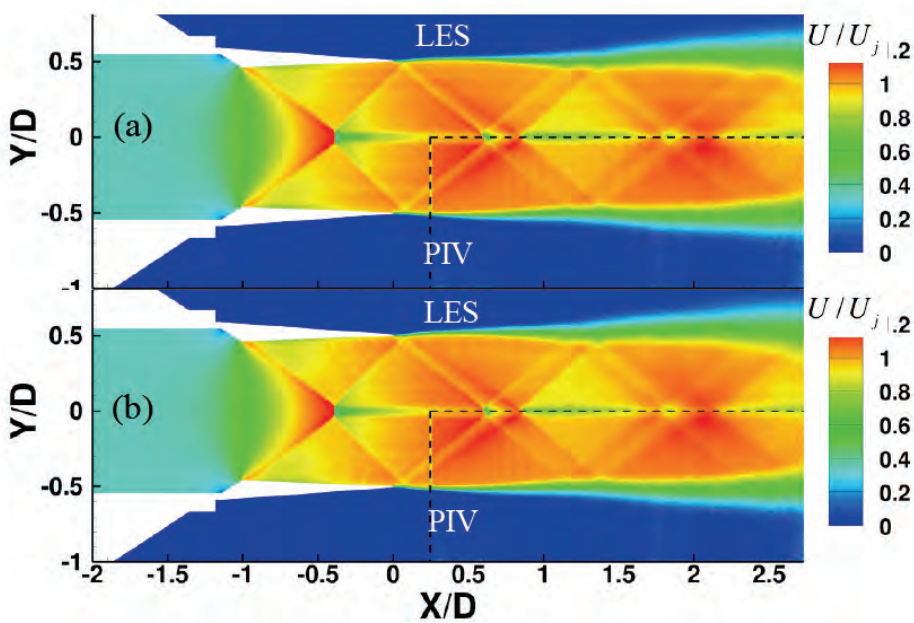


Figure 5. Axial velocity comparison between LES using the finer grid resolution (Mesh II) and PIV measurement[15]; (a) Perturbations at the level of 0.25% are added; (b) Wall roughness is used

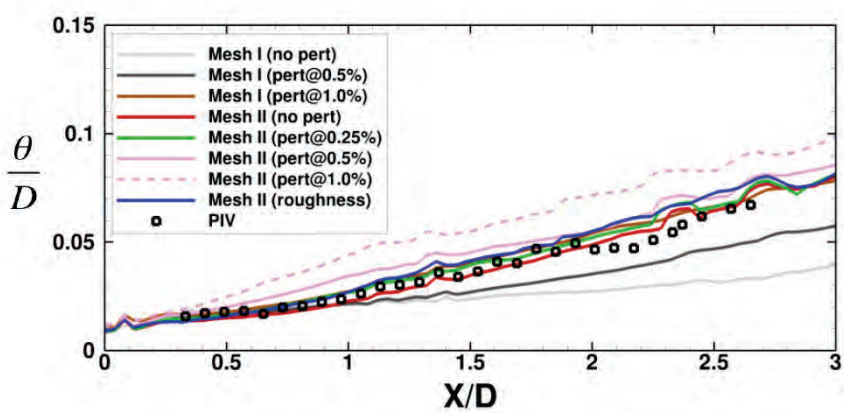


Figure 6. Impact of the turbulence level at the nozzle exit on momentum thicknesses with two grid resolutions in the shear layer

The impact of the turbulence level at the nozzle exit on the non-dimensionalized power spectral densities of the axial turbulence u' along the lip line using Mesh I and II are shown in Figure 7. The coarser grid resolution has a turbulence level roughly an order of magnitude lower inside the shear layer than that predicted by using the finer grid resolution except a strong screech tone that can be seen even at x=5.0D. Perturbing the flow inside nozzle or increasing the grid resolution increases the turbulence level in the shear layer, and the increased grid resolution produces a turbulence level slightly higher than that given by the coarser mesh simulation with a perturbation level of 0.5%. It can be seen that this strong screech tone is suppressed rather effectively by a small increase of the turbulence level at this mildly under-expanded jet condition. The screech tone, however, can be still seen in the coarser mesh simulation with a perturbation level of 0.5%, but it is not visible when the turbulence level is further increased. It should be mentioned that this strong screech tone might not be observed in experiments at this jet condition, because the turbulence level there should be always higher than what is shown in Figure 6(a). However, for some over-expanded jet conditions that have much stronger screech tones, screech tones may not be so easily suppressed and much higher turbulence levels may be needed to control the screech intensity, as shown in Reference 16. Furthermore, it appears that the random perturbation method increases more high-frequency components if the finer grid resolution is used, whereas the increase is more even over the entire frequency range if the surface roughness is used. Although there are large variations of the power spectral density distributions near the nozzle lip, the profiles at x=5.0D remain similar and all of them display an inertial range with the power law exponent of −5/3 around St=1.0, where $St=fD/Uj$ (f is the frequency). In addition, as it would be expected, the finer grid resolution in the shear layer resolves more high-frequency components than the coarser one.

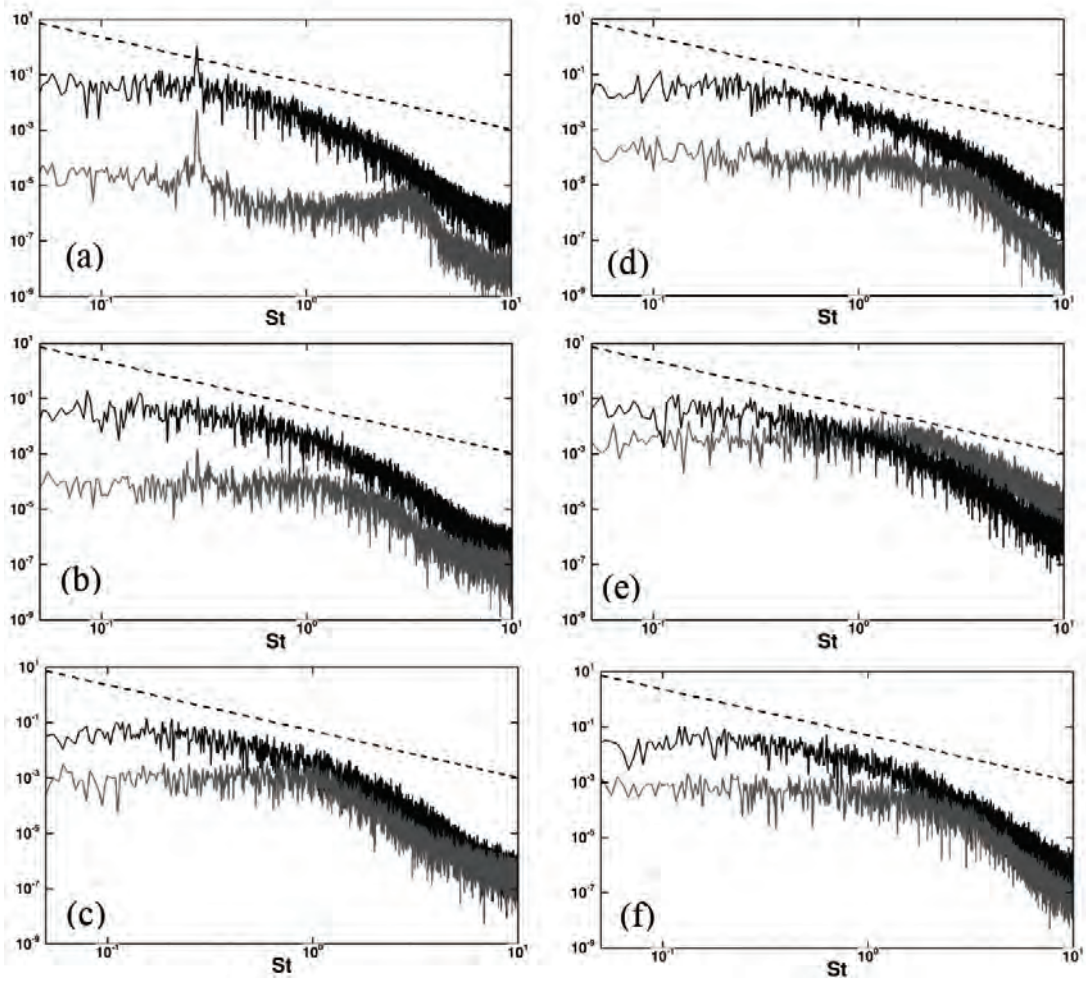


Figure 7. Impact of the turbulence level at the nozzle exit and the grid resolution of the shear layer on the non-dimensionalized power spectral density of the axial turbulence *u'*. Figures on the left are predictions using the coarser grid resolution in the shear layer (mesh I); a) No perturbations; b) Perturbation level at 0.5%; and c) Perturbation level at 1.0%. Figures on the right are predictions using the finer grid resolution in the shear layer (mesh II); d). No perturbations; e) Perturbation level at 0.5%; and f) Wall roughness. Gray lines: x=0.5D and y=0.5D. Black lines: x=5.0D and y=0.5D. The straight dashed lines have a slope of -5/3.

3.2 Far-Field Noise Predictions

Figure 8 shows the far-field noise predictions using these three FW-H surfaces without the contribution of the end cap. The perturbation level is 0.25% and the turbulence kinetic energy at the nozzle exit is 1.2% of the jet velocity. The axial length of these surfaces spans from 2.0D upstream to 27.0D downstream, roughly 2.5~3.0 times of the jet potential core length. The differences among the three predictions are smaller at the upstream angles than those at aft angles, and the largest difference between the smallest surface and the largest surface is roughly 1.0dB, but the difference between these two larger surfaces is only within the range of 0.2dB~0.3dB. The overall agreement between the measurement data and the LES predictions are good and the prediction of the largest FW-H surface agree better with the measurement data. In addition, the contribution from the end cap is found small, as shown in Figure 8, but its calculation is sensitive to the mesh size used in the integration. If the mesh size were too large compared to the cell size used in the simulation, large errors would be found in high-frequency components, but reducing the mesh size will keep this high-frequency contamination under control. Thus, we should either save a large number of points on the end-cap disk to process correctly its contribution to the far-field noise level or neglect its contribution altogether since it is small.

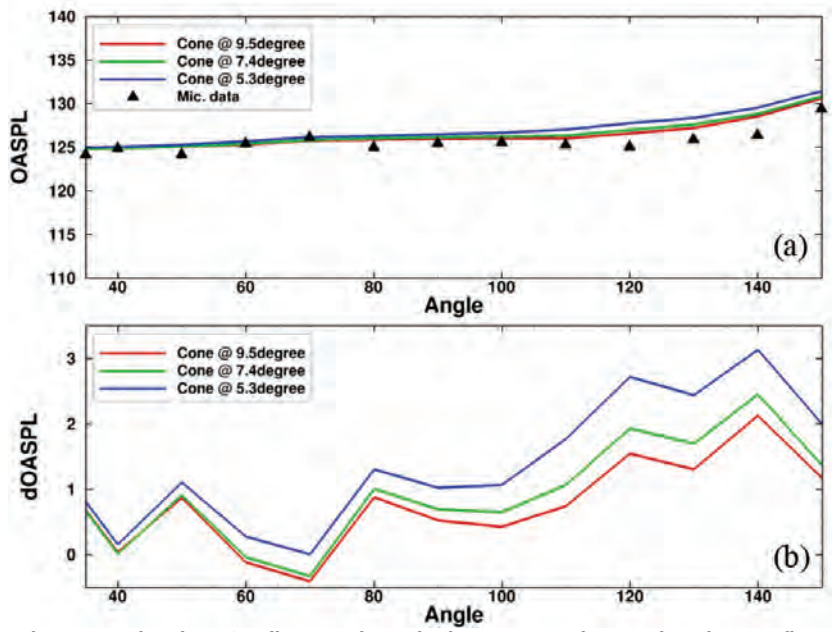


Figure 8. Overall sound pressure level at 50D diameter from the jet center at the nozzle exit. Top figure: The overall sound pressure level using three FW-H surfaces. Bottom figure: the differences between numerical predictions and measurement. The angle is 90° above the nozzle exit and it increases at downstream locations and decreases at upstream locations.

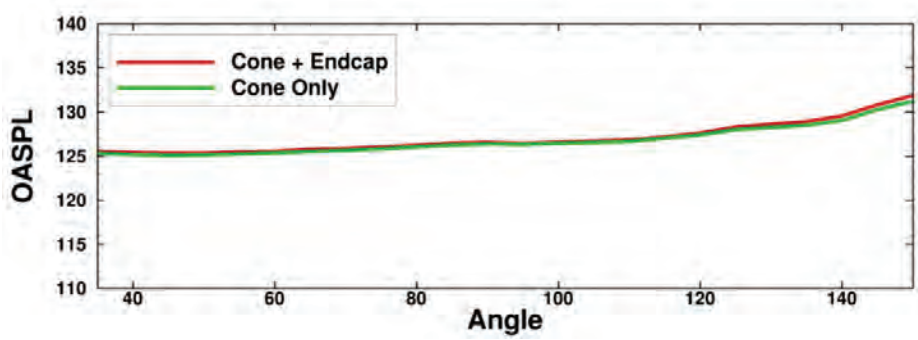


Figure 9. The impact of the end-cap contribution to the far-field noise predictions (the FW-H surface at the angle of 5.3° is used)

The impact of the turbulence level at the nozzle exit on the far-field noise predictions using both Mesh I and Mesh II is shown in Figure 10. It can be seen that, if the turbulence level at the nozzle exit is very low, the coarse grid resolution produces 2.0dB~3.0dB more noise at both the upstream and aft angles because of the presence of the strong screech tone as shown in Figure 6. Increasing the turbulence level decreases the far-field noise level, and the reduction is slightly more at

the aft angles with lower turbulence levels, but the reduction at upstream angles increases with the turbulence level. The increased turbulence level may have enhanced the mixing and thus reduced the mixing noise slightly, and it may also have altered the shock-associated noise to some extent.

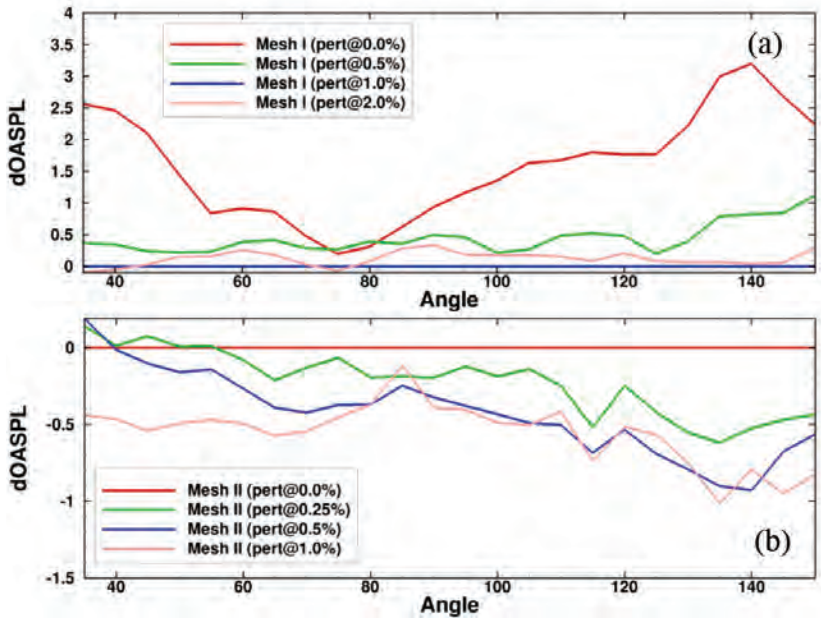


Figure 10. The impact of the inflow turbulence level on the far-field noise prediction using two grid resolutions in the shear layer (the FW-H surface at the angle of 5.3o is used); a) Mesh I (a coarser grid resolution); and b) Mesh II (a finer grid resolution)

Figure 11 shows the impact of the grid resolution of the shear layer, the downstream jet wake region and the near field region on the far-field noise predictions. It can be seen that increasing the grid resolution in both the downstream jet wake region and the near-field region only has a small impact on the overall sound pressure level, but increasing the grid resolution in the shear layer from D/143.0 to D/286.0 increases the overall sound pressure level by roughly 2.0dB at most of the far-field angles. We have found that increasing the grid resolution in shear layer improves both the mid- and high-frequency components, whereas increasing grid resolution in the downstream jet wake region and the near-field region only improves the high-frequency components. Since the magnitude of the high-frequency components is smaller than that of the mid-frequency components, changes made in the high-frequency range should impact the overall sound pressure level less.

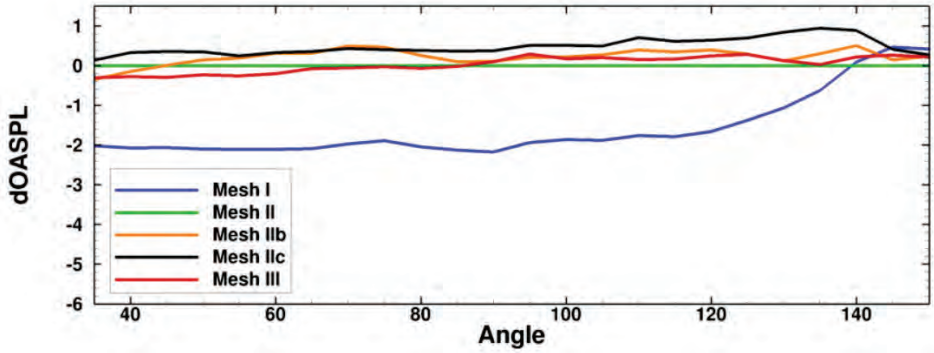


Figure 11. The impact of the grid resolutions used in the shear layer, downstream jet wake region and near-field acoustic propagation region on the far-field noise predictions (the FW-H surface at the angle of 5.3° is used). Inflow perturbations are added inside the nozzle to increase the turbulence levels for all cases.

4. Conclusion

The impact of the turbulence level at the nozzle exit on the flow development and the far-field noise predictions has been investigated in a slightly under-expanded jet flow using large-eddy simulations and the Ffowcs Williams & Hawking surface integral method. Random perturbations inside the nozzle and the nozzle surface roughness are used to increase the turbulence level at the nozzle exit. Three FW-H surfaces are used to study the impact of the integral surface location on the far-field noise prediction.

Increasing the turbulence level at the nozzle exit increases the shear layer spreading, suppresses the screech tone, and also reduces slightly the far-field noise level. The LES predictions of the shock-cell structures agree well with the PIV measurement, and good agreement is also found between the LES far-field predictions and the measurement data. The contribution of the end cap is small, but it is sensitive to the mesh size used for the integration and a mesh size not comparable to the cell size used in the simulation is needed to avoid high-frequency contamination.

The impact of grid resolution in the shear layer, the downstream jet wake region and the region near the FW-H surface has also been investigated. It is found that the grid resolution used in the shear layer impacts the flow development and the far-field noise level more than those used in the other two regions. The prediction using a coarser grid resolution in the shear layer is more sensitive to the turbulence level near the nozzle lip than those using finer grid resolutions, and an adequate amount of turbulence level near the nozzle lip greatly improves the coarser grid simulations.

Acknowledgements

This research has been sponsored by the US Office of Naval Research through the Jet Noise Reduction Project under the Noise Induced Hearing Loss program, as well as the US Naval Research Laboratory (NRL) 6.1 Computational Physics Task Area. The authors would like to thank Prof. Rainald Lohner from the George Mason University for his significant help with the code, FEFLO. Computing resources have been provided by the DoD High Performance Computing Modernization Program under the HPCMP Challenge Project C5G. The computing time was made available at DoD Supercomputing Resource Centers: the US Army Engineer Research and Development Center (diamond and garnet), the US Army Research Laboratory (harold), the US Air Force Research Laboratory (raptor), and the US Naval Research Laboratory (polar).

References

1. Gutmark, E. and Ho, C., "Preferred modes and the spreading rates of jets." *Phys. Fluids*, 26 (10), October 1983

2. Hussain, A.K.M.F. and Zedan, M.F., "Effects of the initial condition on the axisymmetric free shear layer: Effects of the initial fluctuation level." *Phys. Fluids*, Vol. 21, No. 9, pp. 1475–1481, 1978.

3. Long, D., McDonald, T. and Maye, P., "Effect of Inlet Flow Conditions on Noise and Performance of Supersonic Nozzles." *AIAA 2010-3920*

4. Bogey, C. and Bailly, C., "Influence of the nozzle-exit boundary-layer thickness on the flow and acoustic fields of initially laminar jets." *J. Fluid Mech.*, Vol. 663, pp. 507–538, 2010.

5. Glass, R., Effects of acoustic feedback on the spread and decay of supersonic jets." *AIAA Journal*, 6, pp. 1890–1897, 1968.

6. Bridges, J. and Wernet, M., "Turbulence Associated with Broadband Shock Noise in Hot Jets." *AIAA-2008-2835*.

7. Liu, J., Kailasanath, K., Ramamurti, R., Munday, D., Gutmark, E., and Lohner, R., "Large-Eddy Simulations of a Supersonic Jet and Its Near-Field Acoustic Properties." *AIAA Journal*, Vol. 47, Issue 8, pp. 1849–1864

8. Cacqueray, N., Bogey, C., and Bailly, C., "Direct noise computation of a shocked and heated jet at a Mach number of 3.30." *AIAA 2010-3732*.

9. Bogey, C., Marsden, O., and Bailly, C., "Large-eddy simulation of the flow and acoustic fields of a Reynolds number 10^5 subsonic jet with tripped exit boundary layers." *Phys. Fluids*, 23, 035104, 2011.

10. Pokora, C.D., McMullan, W.A., Page, G.J., and McGuirk, J.J., "Influence of a Numerical Boundary Layer Trip on Spatial-Temporal Correlations within LES of a Subsonic Jet." *AIAA-2011-2920*.

11. Boris, J.P. and Book, D.L., "Flux-corrected Transport I: SHASTA a fluid transport algorithm that works." *Journal of Computational Physics*, Volume 11, Issue 1, pp. 38–69, 1973.

12. Lohner, R. and Baum, J., "30 Years of FCT: Status and Directions," *Flux-Corrected Transport: Principles, Algorithms, and Applications*, Eds., D. Kuzmin, R. Lohner, and S. Turek, Springer, pp. 79–104., 2005; *Communications in Applied Numerical Methods*, Vol. 4, Issue 6, pp. 717–729, 1988.

13. Ffowcs Williams, J.E. and Hawkings, D.L., "Sound Generation by Turbulence and Surfaces in Arbitrary Motion." *Philosophical Transactions of the Royal Society of London*, Vol. 264A, pp. 321–342, May 1969.

14. Lyrintzis, A., "Surface integral methods in computational aeroacoustics—From the (CFD) near-field to the (Acoustic) far-field." *International Journal of Aeroacoustics*, Volume 2, Number 2, pp. 95–128, April 2003.

15. Munday, D., Gutmark, E., Liu, J., and Kailasanath, K., "Flow Structure and Acoustics of Supersonic Jets from Conical Convergent-Divergent Nozzles." *Physics of Fluids*, Vol. 23, No. 11, 2011.

16. Liu, J., Kailasanath, K., Ramamurti, R., Heeb, N., Munday, D., and Gutmark. E., "Numerical Investigation of Fluidic Enhanced Chevrons for Noise Reduction in Supersonic Jets." *AIAA-2012-9-502*.

General Purpose Computing on Graphics Processing Units: Optimization Strategy

Henry Au and Gregory Lum
US Space and Naval Warfare Systems Center Pacific (SSC Pacific), Pearl City, HI
{henry.au, gregory.lum}@navy.mil

Abstract

This paper describes the optimization strategies when porting traditional C/C++ algorithms which run on CPU's to parallel processing architectures found on graphics processing units (GPUs). The Compute Unified Device Architecture (CUDA) parallel programming architecture is also explored through the use of NVIDIA's Visual Profiler for performance analysis. Real-time video feeds, such as from onshore surveillance cameras, offer limited visibility when fog, haze, smoke, or dust clouds are present. In order to enhance the video, image processing algorithms such as the Adaptive Linear Filter (ALF) are performed. However, algorithms such as the ALF require large computational time, thus limiting the picture quality, size of the video, or number of video feeds being processed concurrently in real-time. The GPU's parallel processing computational power is exploited to attain speed-ups so that image processing can be performed on-the-fly in real-time. Thus, surveillance is enhanced by providing visual improvement for detection and classification of objects in low-visibility conditions using the ALF. The ALF was selected to provide an image processing context for algorithm optimization on GPUs. The optimization strategies being explored will be CUDA Host memory allocations, streams, and asynchronous memory transfers. Performance results of the ALF running on the GPU and the GPU after optimization will also be reported. As well, GPU limitations will also be briefly discussed in this paper, as not every algorithm will benefit from execution on parallel processing architectures.

1. Introduction

Graphics processing units (GPUs) have been in use in one form or another to display information to users since the 1980s. GPUs continued to evolve from simple shape accelerators to performing more complex computations such as three-dimensional (3D) rendering. However, only as recently as 2007 did General-Purpose Computing on Graphics Processing Units (GPGPU) become a viable option for high performance computing. This availability is due to NVIDIA's Compute Unified Device Architecture (CUDA). CUDA has provided a lot of the back-end coordination required for managing the hundreds of parallel cores found on their GPUs. As well, an added benefit of GPGPU is the ease with which GPUs can be added or upgraded to a standalone desktop machine for increased performance.

Using the large number of cores available on a single GPU, a desktop computer or even laptop can become a mobile HPC device. This makes it ideal for military applications where mobility, package size, and energy requirements are important factors. Remote drones or unmanned aerial vehicles (UAV) suddenly become possible applications. With a GPU installed on a UAV, data can be processed in near-real-time on the aircraft instead of post-processed at a remote site when time-sensitive information is required.

Figure 1 show how a set of data is segmented and processed in parallel using blocks and threads. A thread is a set of operations that processes data independent of order, thus allowing for parallel execution. Multiple threads create a block, and multiple GPU cores process multiple blocks at the same time. With this architecture, it can be easily seen that additional cores result in more data being processed in parallel. Thus, overall computational time is reduced. This makes it more efficient than a CPU that processes data sequentially. However, there are limitations associated with GPGPU due to the fundamental differences between CPU and GPU cores. The CPU core is much more robust and faster, enabling it to handle a wider range of tasks when compared to a GPU core. However, since CPU processors have orders-of-magnitude fewer cores than GPUs, when dealing with highly-parallel computations, the GPU outperforms the CPU in floating point operations per second (FLOPS). As will be discussed later, GPGPUs do have limitations.

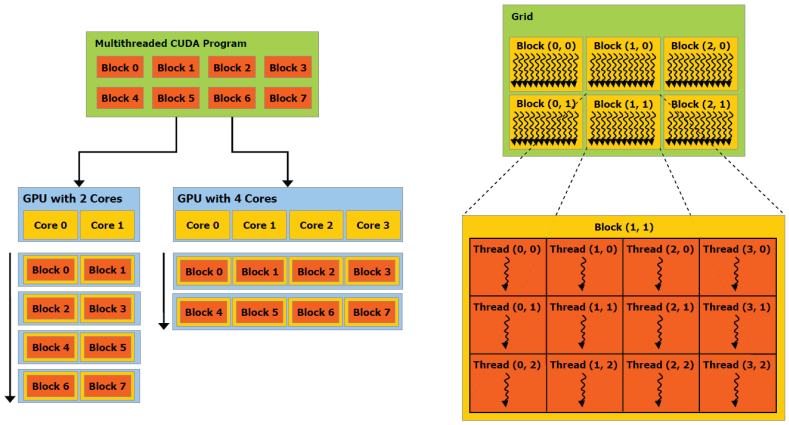


Figure 1. Parallel computation of data

2. Methodology

In order to utilize the computational power of GPUs, a standalone desktop was built with the following hardware: an Intel Core i7, 8GB of DDR3 RAM, 7200 RPM hard drive, 1000W power supply, two NVIDIA GTX 460 graphics cards (one for computer display and the other dedicated for processing). Software utilized included Windows 7 64bit Pro, CUDA Toolkit 4.2, University of Oregon's Tuning and Analysis Utilities (TAU) 2.20 timing software, NVIDIA's Visual Profiler, and Microsoft Visual Studio 2008 Professional. Figure 2 describes the CUDA-enabled NVIDIA GTX 460 graphics card used for parallel processing. It should be noted that NVIDIA offers much higher performance GPUs dedicated for general-purpose computing, such as the Fermi-based GPGPUs. The GTX 460 was selected due to its lower price-point. It can be seen that the GTX 460 supports concurrent copy and execute with one (1) copy engine. Other enterprise-level cards have two (2) copy engines, allowing for greater host-to-device and device-to-host memory copying performance.

```
Device 0: "GeForce GTX 460"
  CUDA Driver Version / Runtime Version          4.2 / 4.2
  CUDA Capability Major/Minor version number:    2.1
  Total amount of global memory:                 1024 MBytes (1073414144 bytes)
  ( 7) Multiprocessors x ( 48) CUDA Cores/MP:    336 CUDA Cores
  GPU Clock rate:                                1440 MHz (1.44 GHz)
  Memory Clock rate:                             1800 Mhz
  Memory Bus Width:                              256-bit
  L2 Cache Size:                                 524288 bytes
  Max Texture Dimension Size (x,y,z)             1D=(65536), 2D=(65536,65535), 3D=(2048,2048,2048)
  Max Layered Texture Size (dim) x layers        1D=(16384) x 2048, 2D=(16384,16384) x 2048
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 32768
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1536
  Maximum number of threads per block:           1024
  Maximum sizes of each dimension of a block:    1024 x 1024 x 64
  Maximum sizes of each dimension of a grid:     65535 x 65535 x 65535
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and execution:                 Yes with 1 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Concurrent kernel execution:                   Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support enabled:                No
  Device is using TCC driver mode:               No
  Device supports Unified Addressing (UVA):      No
  Device PCI Bus ID / PCI location ID:           2 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >
```

Figure 2. NVIDIA GTX 460 GPU specifications

The ALF algorithm was profiled using TAU 2.20, with special consideration taken to ensure that functions which occur repeatedly, such as *for loops*, were profiled. The timing information from the ALF TAU profile was then used to develop a decomposition strategy. It is important to keep in mind that profiling a system that currently operates at near-real-time through CPU processing may not reveal bottlenecks based only on timing. It is therefore important to understand the algorithm being profiled, as well as to understand the timing profile created using TAU. Figure 3 depicts the ALF algorithm processing a 720×480 resolution image.

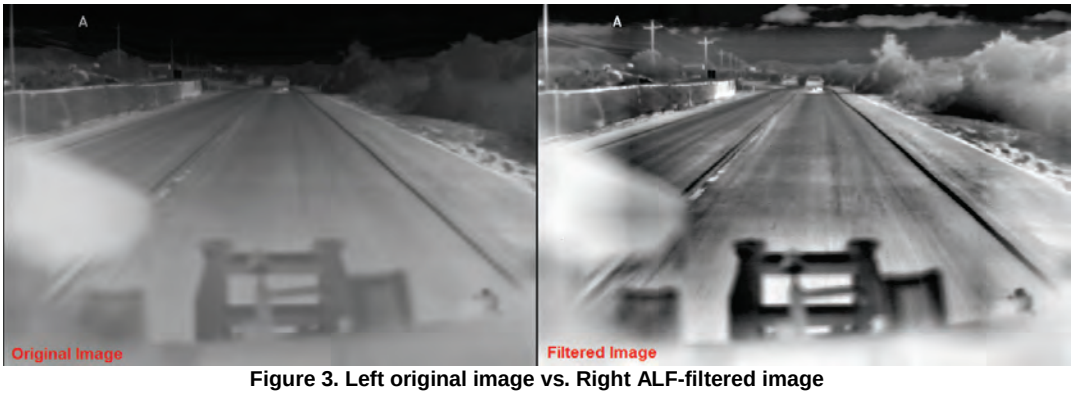


Figure 3. Left original image vs. Right ALF-filtered image

The parallelizable code is then ported and another timing profile is conducted to determine the increased performance. As well, some portions of serial code are also ported to the GPU and processed serially, in order to reduce memory transfers between the host and device. Determining whether transferring the data or processing the data directly on the GPU, requires the timing profile and a clear understanding of the GPU's performance. Thus, it is important to benchmark the performance of the system before optimizing the host and device code.

3. Results and Discussion

3.1 Memory Declarations CPU and GPU

When integrating CUDA C into existing C/C++ code there are three available methods to allocate host memory: new, malloc, and cudaHostAlloc. The cudHostAlloc is similar to malloc, except that it allocates a buffer of page-locked host memory. Thus, the memory will not be paged out to disk and the GPU can utilize its direct memory access (DMA) engine to copy data to and from the host. This is especially important for performance enhancement, because many memory copies between the host and device will result in decreased performance. Figure 4 depicts the processing speed-up achieved by using cudaHostAlloc to page-lock memory for data transfers between the host and device. However, overuse of page-locked memory can cause decreased system performance.

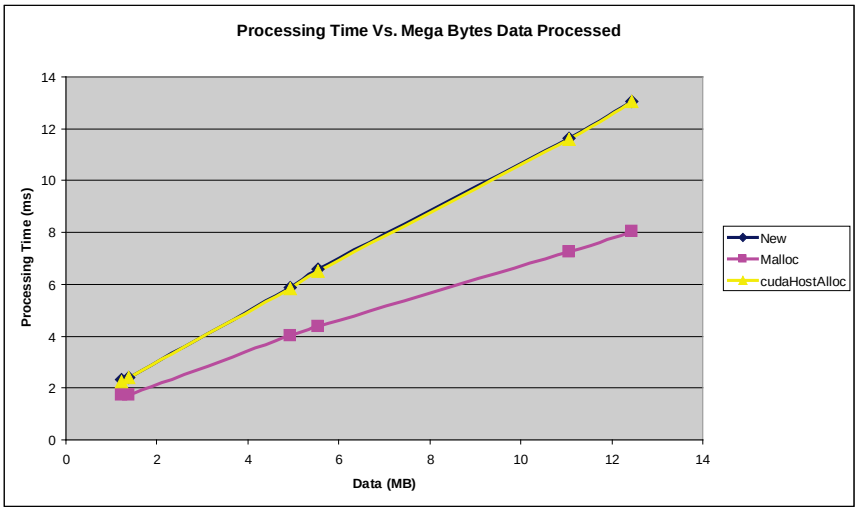


Figure 4. Data processing using: New vs. Malloc vs. cudaHostAlloc for memory allocation

3.2 Asynchronous Memory Transfers and Execution using Streams

NVIDIA's programming guide recommends that streams be created to properly queue device instructions to utilize the asynchronous memory copy and kernel executions. Asynchronous memory transfers and kernel executions require a GPU with concurrent copy and execution available, as well as the memory being copied between host and device to be page-locked (host memory allocated using cudaHostalloc). It is important to remember that asynchronous memory transfers and computations should be utilized for portions of code which are independent of other memory transfers and computations. Figure 5 provides an example of an improperly queued stream which does not utilize the concurrent copy and execute available on the GTX 460. Highlighted in red are instructions which have been queued for stream 0 and instructions queued on stream 1 in black.

```
cudaMemcpyAsync(dataA0, stream0, HostToDevice)
cudaMemcpyAsync(dataB0, stream0, HostToDevice)
kernel<<< blocks, threads, stream0>>>(result0, dataA0, dataB0)
cudaMemcpyAsync(result0, stream0, DeviceToHost)
cudaMemcpyAsync(dataA1, stream1, HostToDevice)
cudaMemcpyAsync(dataB1, stream1, HostToDevice)
kernel<<<blocks, threads, stream1>>>(result1, dataA1, dataB1)
cudaMemcpyAsync(result1, stream1, DeviceToHost)
```

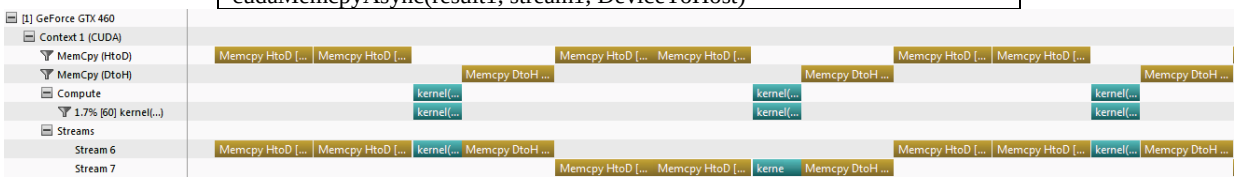


Figure 5. NVIDIA Visual Profiler results for version 1 stream queuing (time taken 49.5ms)

It can be seen in Figure 5 that by queuing the device instructions of stream 0 and stream 1 in a serial manner, the memory copies (tan) and computations (teal) are not overlapped and no performance increase is achieved. The proper CUDA stream queue structure would be interleaving the two streams, so that as memory is being copied, the kernel can be instructed to execute. Figure 6 demonstrates the proper use of CUDA streams.

```
cudaMemcpyAsync(dataA0, stream0, HostToDevice)
cudaMemcpyAsync(dataA1, stream1, HostToDevice)
cudaMemcpyAsync(dataB0, stream0, HostToDevice)
cudaMemcpyAsync(dataB1, stream1, HostToDevice)
kernel<<< blocks, threads, stream0>>>(result0, dataA0, dataB0)
kernel<<<blocks, threads, stream1>>>(result1, dataA1, dataB1)
cudaMemcpyAsync(result0, stream0, DeviceToHost)
cudaMemcpyAsync(result1, stream1, DeviceToHost)
```

Figure 6. NVIDIA Visual Profiler results for version 2 stream queuing (time taken 49.4ms)

However, when using the visual profiler we see that after the first asynchronous memory copy and transfer the subsequent calls result in a serial behavior. Through testing it was found that when the kernel execution of stream 0 is immediately followed by the stream 0 memory copy, and similarly for stream 1, that full asynchronous copy and execution is achieved.

In Figure 7 it can be seen that asynchronous memory copy and kernel executions overlap when queuing device instructions in this form. Thus, a portion of the memory transfer overhead is hidden during the kernel execution. For the GTX 460 only one copy engine is available. If two copy engines are available performance would increase even more as less overhead is required for memory copies.

```
cudaMemcpyAsync(dataA0, stream0, HostToDevice)
cudaMemcpyAsync(dataA1, stream1, HostToDevice)
cudaMemcpyAsync(dataB0, stream0, HostToDevice)
cudaMemcpyAsync(dataB1, stream1, HostToDevice)
kernel<<< blocks, threads, stream0>>>(result0, dataA0, dataB0)
cudaMemcpyAsync(result0, stream0, DeviceToHost)
kernel<<<blocks, threads, stream1>>>(result1, dataA1, dataB1)
cudaMemcpyAsync(result1, stream1, DeviceToHost)
```

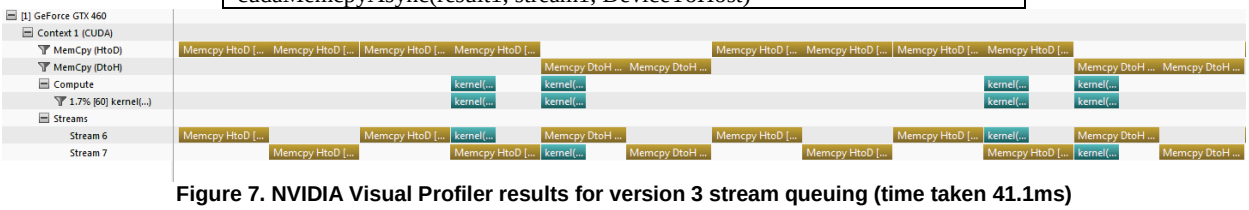


Figure 7. NVIDIA Visual Profiler results for version 3 stream queuing (time taken 41.1ms)

Figure 8 depicts the increase in performance when comparing proper use of streams with no streams (streams queued in serial). As more data is processed, the effect of proper stream queuing becomes more evident. Thus, utilizing streams lends to scalability and performance increase in the future for larger data structures.

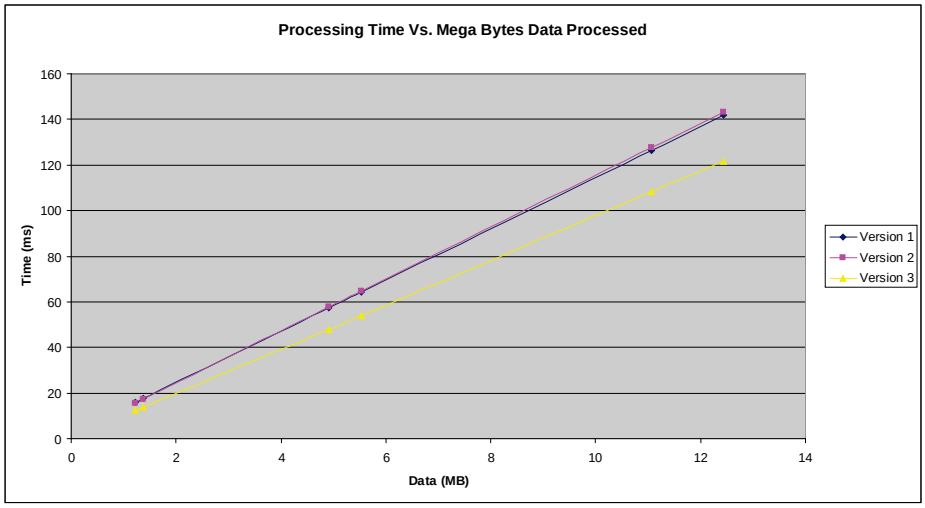


Figure 8. No Stream, stream, and modified stream queue structure processing results

However, asynchronous memory copies and concurrent kernel executions utilized by proper device stream queuing does not always result in a performance increase. This is the result of the overhead associated with creating, synchronizing, and destroying streams. Thus, it is important to profile the device code to ensure that a sufficient amount of processing time occurs during computations of the kernels and not during the memory copying process. Figure 9 depicts these results, revealing that when computation time is low in comparison to copying time, concurrent copy and computation results in degraded performance.

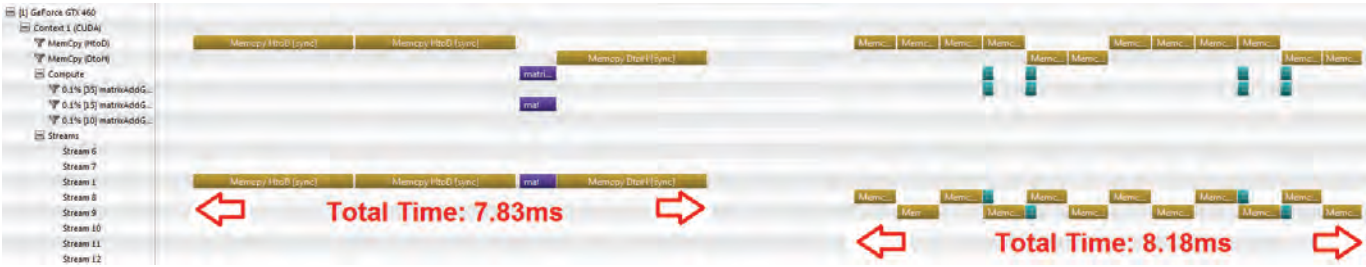


Figure 9. Visual Profile of serial device copy-compute-copy and concurrent copy-compute

3.3 GPU Processing Results

The ALF algorithm was ported to the GPU and profiled. Using the timing profile of the CPU and GPU processing times, the code was then optimized by determining which data structures should be page-locked and thus able to be streamed for asynchronous memory copy and kernel execution. Table 1 presents the ALF algorithm processing time for a single frame of varying image resolutions. It can be seen that after optimization, an increase in frames per second (FPS) is achieved.

Table 1. Complete ALF algorithm execution time

Resolution	ALF: GPU+CPU (FPS)	ALF: GPU+CPU Optimized (FPS)	Speed Up (FPS)
720×480	7.59	17.81	10.22
1440×960	5.06	8.78	3.72
2160×1440	3.24	4.52	1.28

4. Conclusion

This paper has demonstrated that simple optimization strategies such as using page-locked host memory, asynchronous memory copies and concurrent kernel executions, and proper stream coordination result in a decrease in processing time. However, it has also been demonstrated that utilizing streams to hide the latency from device-to-host and host-to-device memory copies does not always result in increased performance. Hiding the memory copy latency works best when the computation times are comparable to memory copy times so that the stream creation, synchronization, and destruction overhead do not add more latency than it is able to hide.

Thus, developing a decomposition strategy in conjunction with benchmarked CUDA overhead requirements will help develop an optimization strategy to utilize the full computational power available through GPUs. This paper has shown that a speed-up of ~2x is achieved for the ALF algorithm operating on an 720×480 resolution image.

As described previously, the GPUs have their limitations, such as the large overhead associated with memory copies between host and device and vice versa. However, for image processing, GPUs can be an asset. The GPUS' size-to-performance also makes it the ideal technology for a range of military applications, such as UAV onboard data processing and mobile high performance computers for forward-operating bases.

Acknowledgements

This work has been funded by SPAWAR System Center Pacific Internal Applied Research (IAR) program (Dave Rees, Program Manager). We would like to thank David Buck for allowing us to use his ALF algorithm. As well, if not for the support from our Department Manager, George McCarty; Code H Business Deputy, Neal Miyake; H5600 Division Head, Alan Umeda; and H56D0 Branch Head, Justin Lee, we would not have had this great opportunity. Last, and definitely not least, we would like to thank Nick Kamin and Dr. Randy Shimabukuro for mentoring us throughout the course of this applied research.

References

CUDA C Best Practices Guide Ver 4.0, 5/2011.

NVIDIA CUDA Programming Guide Ver 4.0, 5/6/2011.

Sanders, Jason, and Edward Kandrot, *CUDA by Example, An Introduction to General-Purpose GPU Programming*, Addison-Wesley, Copyright NVIDIA Corporation, 2011.

High-Resolution and Probabilistic Tropical Cyclone Track and Intensity Prediction

James Doyle, Carolyn Reynolds, Sue Chen, Yi Jin, Hao Jin, Chi-Sann Liou, Justin McLay, Jon Moskaitis, and James Ridout
US Naval Research Laboratory (NRL-MRY), Monterey, CA
{james.doyle, carolyn.reynolds, sue.chen, yi.jin, hao.jin, chi-sann.liou, justin.mclay, jon.moskaitis, james.ridout}@nrlmry.navy.mil

Richard Hodur
Science Applications International Corporation, Monterey, CA
richard.hodur@nrlmry.navy.mil

Abstract

New capabilities to predict tropical cyclone track, intensity, and structure have been developed and extensively tested using the Navy's regional and global modeling systems. A high-resolution tropical cyclone prediction capability has been developed for the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS-TC™) and tested in real-time over multiple regions of the globe. The results show that COAMPS-TC was the most skillful dynamical model for tropical cyclone intensity forecasts in the W. Atlantic basin for the 2010-2011 seasons. The model has superior intensity skill relative to the current Navy's operational regional tropical cyclone model, and comparable tropical cyclone track skill. An example is shown for Hurricane Irene, in which the COAMPS-TC provides very skillful predictions. For the global atmospheric ensemble prediction system, the impact of parameter variations on the Navy Operational Global Atmospheric Prediction System ensemble performance is examined. In the tropics, the parameter variations significantly increase ensemble spread, and significantly improve probabilistic prediction of low-level wind-speed. There are also small but significant improvements in the ensemble mean tropical cyclone track forecasts. In addition, an increase in horizontal resolution, transitioned to operations in September 2011, has been shown to significantly improve tropical cyclone track forecasts. Real-time high-resolution ensembles have been run for the 2010 and 2011 tropical cyclone seasons, and products are being developed to convey uncertainty information contained within the ensembles. Work is continuing on improving ensemble performance through inclusion of model uncertainty in ensemble design and further increases in the resolution of the ensemble members, and it is anticipated that these improvements will be transitioned to Navy operations within the coming year.

1. Introduction

A dramatic scenario played out during August 2011 as Hurricane Irene threatened many communities along the US Eastern Seaboard, from Florida to New England. Basic questions such as where Irene would track and how strong it would become had profound implications for the millions of people in its path and billions of dollars that were vulnerable. The potential impact of tropical cyclones on military operations can also be enormous. An extreme example is the infamous Typhoon Cobra, also known as Halsey's Typhoon after Admiral William Halsey, which struck the US Navy's Pacific Fleet in December 1944 during World War II. Three destroyers were lost, and a total of 790 sailors perished. More recently during Irene, the decision to sortie Navy assets from Norfolk, VA and other ports along the Eastern Seaboard many days in advance of the storm was critically dependent on forecasts of Irene's track, intensity (maximum sustained wind-speed at the surface), and storm structure (such as the size of the storm or radius of key wind-speed thresholds). In the N.W. Pacific, the Navy Pacific Fleet in the Philippine Sea was impacted recently by Typhoon Nanmadol, which exhibited erratic movement and was poorly forecasted.

The demand for more accurate tropical cyclone forecasts with longer lead-times is greater than ever due to the enormous economic and societal impact. There has been spectacular improvement of tropical cyclone (TC) track prediction; a three-day hurricane track forecast today is as skillful as a one-day forecast was just 30 years ago. However, there has been almost no progress in improving TC intensity and structure forecasts due to a variety of reasons, ranging from a lack of critical observations under high-wind conditions and in the TC environment to inaccurate representations of TC physical processes in numerical weather prediction (NWP) models. Advances in high-resolution TC modeling and data assimilation are thought to be necessary to significantly improve the performance of intensity and structure prediction. To this end, the US Naval Research Laboratory (NRL) in Monterey, CA, has developed the Coupled Ocean/Atmosphere Mesoscale Prediction System for Tropical Cyclones (COAMPS-TC), a new version of COAMPS designed specifically for high-resolution tropical cyclone prediction. Additionally, new probabilistic prediction capabilities for the Navy Global Atmospheric Prediction System (NOGAPS) are being developed, as well. Research continues to provide evidence that increased resolution and inclusion of model uncertainty in the ensemble design increases the skill of tropical cyclone track forecasts provided by the NOGAPS ensembles. A recent upgrade to the operational ensemble system should result in improved operational tropical cyclone track forecasts, and additional transitions are anticipated for next year.

2. Overview of COAMPS-TC

A new version of the Coupled Ocean/Atmosphere Mesoscale Prediction System for Tropical Cyclones (COAMPS-TC™) has been developed specifically for forecasting tropical cyclone track, structure, and intensity. The COAMPS-TC has been tested in real-time in both coupled and uncoupled modes over the past several tropical cyclone seasons in the Pacific and Atlantic basins at a horizontal grid-spacing of 5 km. An evaluation of a large sample of real-time forecasts for the 2010 and 2011 seasons in the Atlantic basin reveals that the COAMPS-TC predictions have smaller intensity errors than other real-time dynamical models for forecasts beyond the 30 h time. Real-time forecasts for Hurricane Irene (2011) illustrate the capability of the model to capture both the intensity and the fine-scale features (e.g., eyewall, rainbands), in agreement with observations. The results of this research highlight the promise of high-resolution deterministic and ensemble-based approaches for tropical cyclone prediction using COAMPS-TC.

3. COAMPS-TC Description

The COAMPS-TC system is comprised of data quality control, analysis, initialization, and forecast model sub-components. The Navy Variational Data Assimilation System (NAVDAS) is used to blend observations of winds, temperature, moisture, and pressure from a plethora of sources such as radiosondes, pilot balloons, satellites, surface measurements, ships, buoys, and aircraft. Enhancements to the NAVDAS system for COAMPS-TC include the addition of synthetic observations that define the TC structure and intensity, based on the TC reports in real-time from the National Hurricane Center (NHC) and the Joint Typhoon Warning Center (JTWC). Also, as part of the TC analysis procedure, the pre-existing circulation in the COAMPS-TC first-guess fields is relocated to allow for an accurate representation of the TC position during the analysis. Following this step, the analyzed fields are initialized to reduce the generation of spurious, high-frequency atmospheric gravity waves. The sea surface temperature is analyzed directly on the model computational grid using the Navy Coastal Ocean Data Assimilation (NCODA) system, which makes use of all available satellite, ship, float, and buoy observations. Both the NCODA and NAVDAS systems are applied using a data assimilation cycle in which the first-guess from the analysis is derived from the previous short-term forecast.

The COAMPS-TC atmospheric model uses the non-hydrostatic and compressible form of the dynamics, and has prognostic variables for the three components of the wind (two horizontal wind components, and the vertical wind), the perturbation pressure, potential temperature, water vapor, cloud droplets, raindrops, ice crystals, snowflakes, graupel, and turbulent kinetic energy. Physical parameterizations include representations of cloud microphysical processes, convection, radiation, boundary layer processes, and surface layer fluxes. The COAMPS-TC model contains a representation of dissipative heating near the ocean surface, which has been found to be important for tropical cyclone intensity forecasts. The model also contains an optional advective scheme for scalars that preserves monotonicity and is positive definite. The COAMPS-TC system uses a flexible nesting design that has proven useful when more than one storm is present in a basin at a given time, as well as special options for moving nested grid families that independently follow individual tropical cyclone centers of interest.

The COAMPS-TC system has the capability to operate in a fully-coupled air-sea interaction mode[11]. The atmospheric module within COAMPS-TC is coupled to the NRL-developed Navy Coastal Ocean Model (NCOM) to represent air-sea interaction processes. The COAMPS-TC system has an option to predict ocean surface waves, and the

interactions between the atmosphere, ocean circulation, and waves using the Simulating WAves Nearshore (SWAN) model. A sea-spray parameterization can be used to represent the injection of droplets into the atmospheric boundary-layer due to ocean surface wave-breaking and shearing.

4. Evaluation of COAMPS-TC Forecasts

COAMPS-TC has been tested in real-time in both coupled and uncoupled modes over the past several tropical cyclone seasons in the Pacific and Atlantic basins. These real-time tests have been conducted in conjunction with the National Oceanic and Atmospheric Administration (NOAA)-sponsored Hurricane Forecast Improvement Project (HFIP), which is focused on accelerating the improvement in hurricane intensity forecasts. In these real-time applications, the atmospheric portion of the COAMPS-TC system makes use of horizontally-nested grids with grid-spacing of 45, 15, and 5 km. The 15- and 5-km grid-spacing meshes track the TC center, which enables the TC convection to be more realistically represented on the finest mesh in an efficient manner. The forecasts make use of the Navy or the NOAA global model for lateral boundary conditions.

The model is typically run four times daily for the W. Atlantic, E. Pacific, and W. Pacific regions, and is triggered by the NHC and JTWC warning message (which contains observational estimates of the storm position and intensity) when a storm reaches a 30 knot intensity. The forecasts are routinely disseminated in real-time to NHC, JTWC, and HFIP researchers. The forecast graphics are also available in real-time at http://www.nrlmry.navy.mil/coamps-web/web/tc.

4a. COAMPS-TC Atmospheric Model Forecasts

Real-time COAMPS-TC forecasts have been conducted using US Department of Defense (DoD) high performance computing (HPC) platforms over the past several years. An example of the intensity forecast performance of COAMPS-TC for a large number of cases (more than 450 cases at the 24-h forecast-time) in the W. Atlantic region for the 2010 and 2011 seasons is shown in Figure 1 (for a homogeneous statistical sample). The COAMPS-TC model had the lowest intensity error of any dynamical model for the 36–120 h forecast-times, which is an important period for forecasters and decision makers. Other numerical models included in this analysis are operational models run by NOAA (HWRF, GFDL), and the Navy's current operational limited-area model (GFDN). This promising performance is a result of a large effort devoted to developing and improving COAMPS-TC over the past several years. Key aspects of the COAMPS-TC system that have contributed most to the forecast-intensity skill improvement (as deduced from sensitivity tests) include: i) new synthetic observations and data assimilation system modifications; ii) modified turbulence parameterization (in particular the mixing length) in and above the boundary layer; iii) surface drag, heat, and moisture exchange consistent with recent field campaigns; and iv) fidelity of the microphysical parameterization.

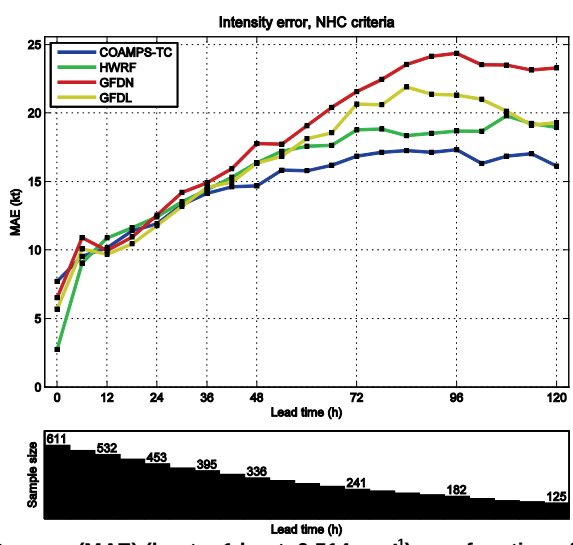


Figure 1. Wind-speed mean absolute error (MAE) (knots; 1 knot=0.514 m s^{-1}) as a function of forecast time for the 2010 and 2011 seasons in the Atlantic basin for a homogeneous statistical sample. The numerical models included in this analysis are the Navy's COAMPS-TC, operational models run by NOAA (HWRF, GFDL), and the Navy's current operational limited-area model (GFDN). The number of cases is shown at the bottom.

An example of a real-time COAMPS-TC forecast for the recent Hurricane Irene is shown in Figure 2. The composite National Weather Service radar reflectivity is shown in the top panel near the time of landfall in North Carolina at 1148 UTC 27 August 2011, and the COAMPS-TC predicted radar reflectivity at 36-h valid at 1200 UTC is shown in the bottom panel. The COAMPS-TC forecast shown in Figure 2 is for the model second grid-mesh (15 km horizontal resolution). The model prediction was accurate in the track (skill similar to other dynamical models), eventual landfall location (Figure 2), storm intensity (Figure 4), as well as the structure and size (from a qualitative perspective), an especially important characteristic of this particular storm in such close proximity to the US East Coast. One noteworthy aspect of Irene was its large size, with tropical storm force winds (34 knot) extending radially outward from the eye for nearly 300 km. The large size of Irene is also apparent in the observed radar reflectivity in Figure 2. The COAMPS-TC prediction captures the large areal extent of the precipitation field, as well as its asymmetry about the TC center (most of the precipitation is north and east of the center). This large shield of heavy precipitation caused severe river flooding as it slowly moved north through the mid-Atlantic and Northeast US. The simulated radar reflectivity for the COAMPS-TC grid mesh 3 (5 km horizontal resolution), shown in Figure 3, illustrates the capability of the model to capture the finer-scale features, such as the eyewall and rainbands, in generally good agreement with the observed reflectivity. Sensitivity tests (not shown) indicate that the synthetic observations and TC-related data assimilation system modifications, along with the in-cloud turbulent mixing representation, are important for the proper simulation of Irene's structure. It should also be noted that COAMPS-TC makes use of a 1.5 order closure turbulence parameterization that predicts turbulent kinetic energy (TKE) and includes advection of TKE as well. We hypothesize that the advection of TKE may be important for the proper representation of the turbulence evolution within the hurricane boundary-layer beneath and near the eyewall.

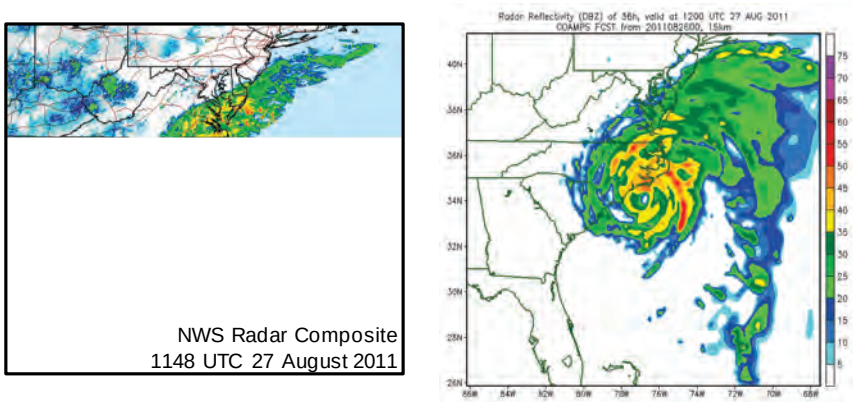


Figure 2. The NWS composite radar reflectivity (from NOAA) valid at 1148 UTC 27 August 2011 (left) (source NOAA), and the COAMPS-TC 36-h forecast radar reflectivity performed in real-time and valid at 1200 UTC 27 August (right) for Hurricane Irene. The COAMPS-TC reflectivity is shown for the second grid-mesh, which has a horizontal resolution of 15 km.

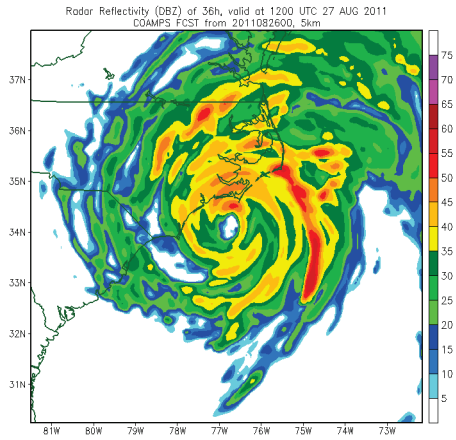


Figure 3. The COAMPS-TC 36-h forecast radar reflectivity valid at 1200 UTC 27 August for the third grid-mesh, which has a horizontal resolution of 5 km

Overall, the Navy's COAMPS-TC real-time intensity predictions of Hurricane Irene were more skillful than the other leading operational governmental forecast models, as shown in Figure 4. All of the available models except for COAMPS-TC had a tendency to over-intensify Irene, often by a full storm category or more. These real-time COAMPS-TC forecasts were used by forecasters at the National Hurricane Center as part of an experimental HFIP multi-model ensemble. The COAMPS-TC consistently provided accurate real-time intensity forecasts during the period 23–28 August 2011, when critical decisions were made by forecasters and emergency managers, including evacuations.

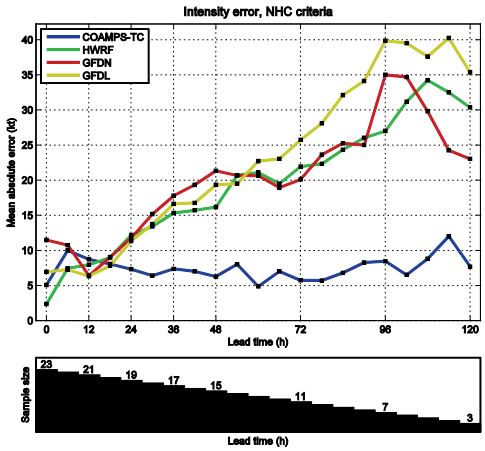


Figure 4. Wind-speed mean absolute error (MAE) (knots) as a function of forecast time for Hurricane Irene for a homogeneous statistical sample. The numerical models included in this analysis are the Navy's COAMPS-TC, operational models run by NOAA (HWRF, GFDL), and the Navy's current operational limited-area model (GFDN). The number of cases is shown at the bottom. Only forecasts after Irene has moved away from Hispaniola are shown here.

4b. COAMPS-TC Ensemble Forecasts

While research is ongoing to improve deterministic atmospheric forecasts through advancements to the forecast model and more accurate estimates of the initial state, simultaneously there has been interest in obtaining probabilistic information derived from ensemble forecasts. An ensemble of forecasts from equally-plausible initial states and model formulations offer a computationally-feasible way of addressing inevitable forecast uncertainties; offering improved forecasts through ensemble statistics such as mean quantities, as well as quantitative estimates of forecast error and variance. Although the concept of ensemble modeling is relatively simple, the performance of an ensemble forecast system is very sensitive to the basic ensemble architecture. At NRL, we are designing new ensemble methods for both the global and mesoscale atmospheric forecast systems. Because of the high computational demands associated with ensemble development and verification (especially when one is interested in severe or rare events), exceptional computational resources are necessary to perform this research.

A new COAMPS-TC ensemble system that is capable of providing probabilistic forecasts of TC track, intensity, and structure has been developed by scientists at NRL in Monterey, CA. This system makes use of the community-based Data Assimilation Research Test-bed (DART) capability developed at the National Center for Atmospheric Research, which includes various options for Ensemble Kalman Filter (EnKF) data assimilation. The COAMPS-TC DART system constitutes a next-generation data assimilation system for tropical cyclones that uses flow-dependent statistics from the ensemble to assimilate observational information on the mesoscale.

A real-time COAMPS-TC ensemble system was run in a demonstration mode in 2011 for the W. Atlantic, E. Pacific, and W. Pacific regions. The system comprised an 80-member COAMPS-TC cycling data assimilation ensemble on three nested-grids with horizontal spacing of 45, 15, and 5 km. For each new warning message issued by either NHC or JTWC, the COAMPS-TC ensemble was initialized by interpolating global forecast fields from the 80-member GFS-EnKF system[14] to the three nested-grids, which were centered on the storm. Six-hour forecasts were made four times daily to provide background estimates for the assimilation of observations from surface stations, ship data, radiosonde ascents, cloud-track wind retrievals, and aircraft data with the DART EnKF. In addition to these conventional data sets, the NHC and JTWC TC position estimates were directly assimilated with the EnKF system. Under-sampling issues in the data assimilation procedure associated with the finite ensemble size were controlled by limiting the spatial influence of observations with a static localization radius of 1,000 km, as well as applying a spatially- and temporall-varying inflation factor[15] to the prior ensemble perturbations. To most effectively use the high-resolution capability of COAMPS-TC, a

two-way interactive data assimilation procedure was implemented. In this algorithm, the innovations were defined using the highest resolution nest that contained the observation. These innovations were used to update the COAMPS-TC fields on all three grids. Furthermore, observations contained outside of a nest were allowed to update the fields within the nest.

Ten-member forecasts were performed twice daily to five days using the same three nested-grid configuration as the data assimilation ensemble. The first 10 members of the data assimilation ensemble were used to define the forecast ensemble. Lateral boundary conditions for the forecast ensemble were drawn from the GFS-EnKF ensemble forecast. Examples of probabilistic products for Hurricane Irene are shown in Figure 5 for both track (top panel) and intensity (bottom panel). This is a real-time forecast initialized at 1200 UTC 23 August, which is four days prior to landfall. The probabilistic track product shows the TC position from the individual ensemble members every 24-h and ellipses that encompass 1/3 and 2/3 of the ensemble member forecast positions. Note that the observed landfall location of the eye (see Figure 2) was within the ensemble distribution, although the ensemble mean landfall was approximately 12-h later than observed. The probabilistic intensity product (lower panel) shows a considerable spread among the members, particularly beyond 84-h, just prior to landfall. These products can be extremely valuable to assess the uncertainty in both track and intensity forecasts, and NRL is currently developing these capabilities and products further.

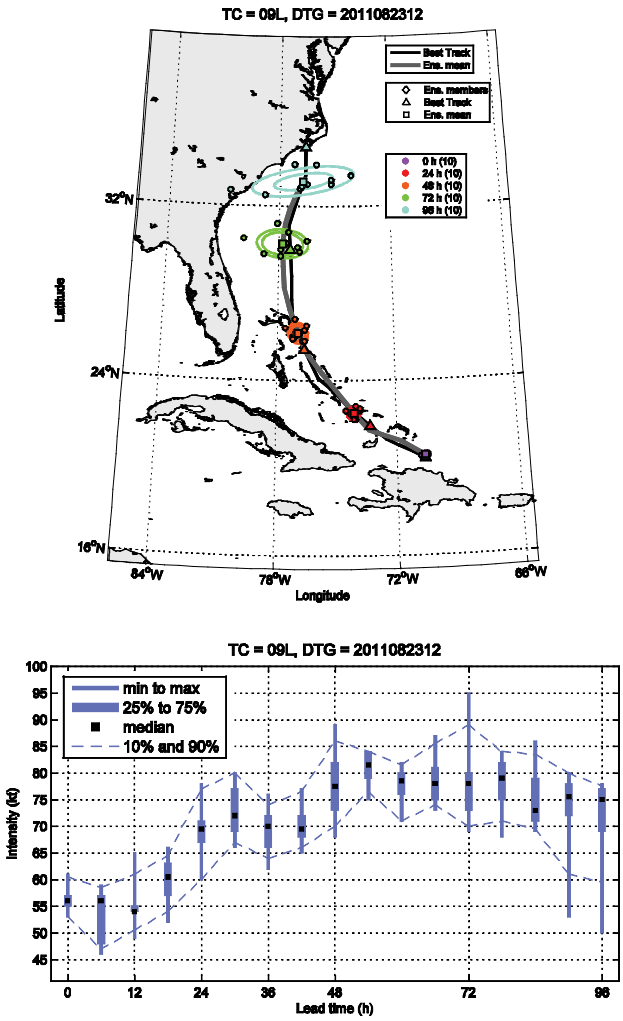


Figure 5. Probabilistic products from the COAMPS-TC ensemble for Hurricane Irene corresponding to the track (top panel) and hurricane intensity (bottom panel). This real-time forecast was initialized at 1200 UTC 23 August, which is approximately four days prior to landfall. The probabilistic track product shows the TC position from the individual ensemble members every 24-h and ellipses that encompass 1/3 and 2/3 of the ensemble members. The intensity (knots) distribution is shown as a function of forecast lead-time (hours) with the minimum value, maximum value and various quantiles of the ensemble distribution shown as denoted by the legend.

4c. Coupled COAMPS-TC Forecasts

The COAMPS-TC system was run in a fully-coupled mode, interactive with NCOM, during the Office of Naval Research sponsored Interaction of Typhoon and Ocean Project (ITOP) during the summer and fall of 2010. An example of a fully-coupled COAMPS-TC forecast for Typhoon Fanapi performed in real-time is shown in Figure 6. The NCOM ocean model was applied using a 10-km horizontal grid-increment in this example, and the finest mesh for the atmospheric model used a 5-km grid-increment. The atmosphere and ocean fluxes were exchanged every ocean model time-step. The COAMPS-TC predicted track (red) from a 90-h real-time forecast valid at 0600 UTC 19 September 2010, is quite close to the observed or best track (black). The sea surface temperature, shown in color shading, indicates significant cooling was predicted by COAMPS-TC during the passage of Fanapi due to enhanced mixing by the strong near-surface winds of the typhoon. The predicted cooling of the sea surface temperatures of 2-4°C is in agreement with estimates from *in situ* and remote sensing observations in this region. The negative impact of the ocean cooling underneath the tropical cyclone can reduce the TC intensity and broaden the tropical cyclone secondary circulation, which underscores the importance of properly representing these air-sea interaction processes.

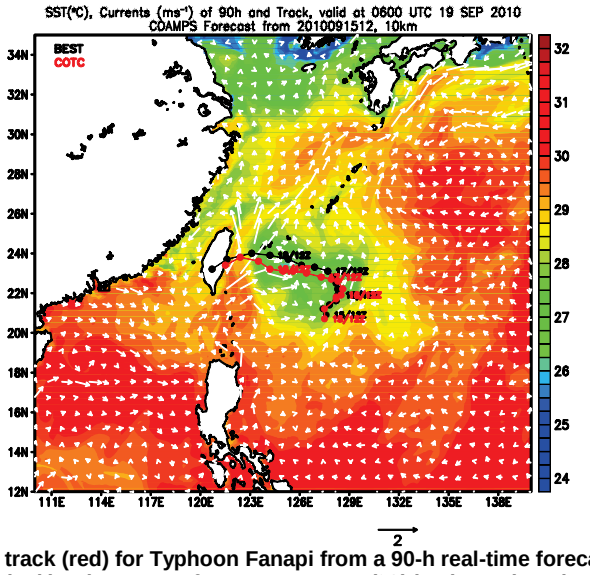


Figure 6. The COAMPS-TC predicted track (red) for Typhoon Fanapi from a 90-h real-time forecast valid at 0600 UTC 19 September 2010 and the observed best track (black). The sea surface temperature (°C) is shown in color shading and indicates significant cooling during the passage of Fanapi. The surface currents are shown by the white vectors.

A joint Navy/Air Force Hurricane Hunter program was in its demonstration phase in 2011 with Airborne Expendable Bathythermographs (AXBTs) being deployed from WC-130J hurricane reconnaissance aircraft in order to improve the initialization and validation of coupled models such as COAMPS-TC. Over 30 AXBTs were deployed from Air Force Hurricane Hunter aircraft in Irene as it approached landfall on Cape Hatteras. Assimilation studies are currently being performed to assess the impact of these AXBTs on the coupled model skill. These new observations supplement previous studies[4,11] of hurricane-induced cooling and may further help document the existence of ocean mixing or advective processes (and cooling of the sea surface) along the storm track and in coastal regions that may have slowed the intensification of Irene and other storms.

5. NOGAPS Ensemble Forecasts

This Challenge Project has been instrumental in the development, testing and transition of improvements to the global ensemble forecasting system run operationally at FNMOC. These transitions have not only produced significant increases in skill from the operational ensemble system, but have also facilitated participation of the Navy global ensembles in the North American Ensemble Forecasting System (NAEFS), and the National Unified Operational Prediction Capability (NUOPC) ensemble. Up until now, the global ensemble has been based on the Navy Operational Global Atmospheric Prediction System (NOGAPS). However, the challenge project is currently supporting the implementation of the next-generation global forecasting system, the Navy Global Environmental Model (NAVGEM) into the ensemble system.

5a. The Banded Ensemble Transform: Operational Transition in May 2010

Before 2010, the method to produce initial perturbations for the operational global ensemble system was based on the global ensemble transform (ET), in which short-term (6-h) ensemble forecasts are transformed globally to be consistent with estimates of analysis error variance. However, there were aspects of the practical application of the global ET that needed improvement. Primarily, given a finite number of ensemble members, the initial ensemble perturbations were too small in the tropics and too large in the mid-latitudes, when compared to the analysis error variance estimate produced by the NRL Atmospheric Variational Data Assimilation System (NAVDAS). Using HPC challenge resources, a new formulation of the ET was developed in which the transform is performed locally within latitude bands. This resulted in a better match to the initial analysis error variance estimate, as well as enhanced ensemble performance over a variety of metrics in the tropics and the extra-tropics (McLay, 2010). In addition to the improved initial perturbation formulation, the inclusion of stochastic forcing to account for model error was also tested (Reynolds, et al., 2011a) and shown to improve performance in the tropics. As an example, Figure 7 shows the Brier score, which is a probabilistic error score, for the ensemble forecast of 10-m wind-speeds exceeding a 5 m s^{-1} threshold in the tropics, for T159 ensemble forecasts averaged over the period 26 June through 15 October 2008.

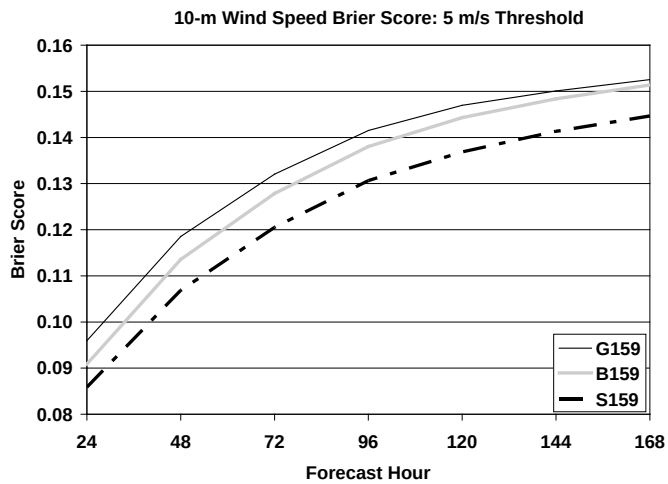


Figure 7. Brier score for the global ET (G159, thin solid black), Banded ET (B159, thick solid grey) and Banded ET with stochastic forcing (S159, thick dash-dot) ensembles for the 5 m s^{-1} wind-speed threshold

As indicated in Figure 7, the banded ET results in lower (better) probabilistic predictions of low-level wind-speeds in the tropics, with improvements ranging between 6 and 12 hours in forecast skill. The addition of stochastic forcing improves things even further. Based on these tests and others, the banded ET was transitioned to operations at FNMOC in February 2010.

5b. Increased Ensemble Resolution: Operational Transition in September 2011

The Challenge Project was also used to investigate the impact of resolution on ensemble performance. It was found that increasing the resolution from T119 (about 110 km horizontal resolution) to T159 (about 82 km horizontal resolution) resulted in significant improvement to ensemble performance, particularly in regards to tropical cyclones (Reynolds, et al., 2011a). The Northern Hemisphere ensemble mean tropical cyclone track forecast errors for the 2008 season are shown in Figure 8.

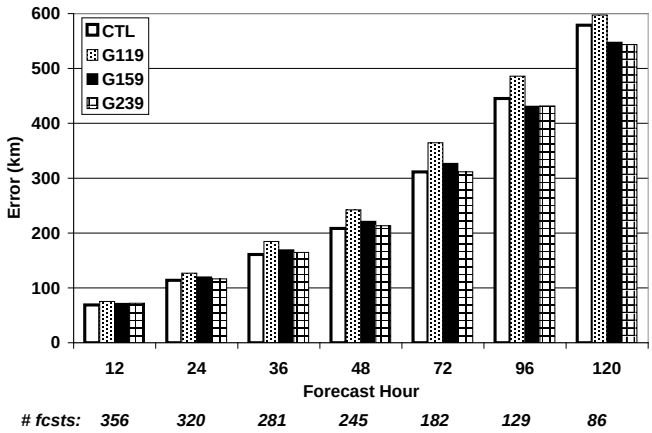


Figure 8. Homogenous NH TC track forecast error (km), for T119 (G119), T159 (G159) and T239 (G239) ensemble mean tracks as denoted in key. Also shown is the average forecast error of the T239L30 NOGAPS operational deterministic forecast (CTL). The numbers of verifying forecasts are shown below the x-axis. The differences between G119 and G159 are statistically-significant at the 95% level out to 96 hours. The differences between G159 and G239 are statistically-significant at 48 and 72 hours.

The biggest improvement comes from increasing the resolution from T119 to T159, with further improvements gained by increasing resolution to T239 at 48 and 72 hours. Based on these results and others using HPC Challenge resources, the resolution of the operational forecast system at FNMOC was increased from T119 to T159 in September 2011.

5c. Inclusion of Model Uncertainty in Ensemble Design

In addition to the stochastic forcing mentioned in Section 5a, other ways to account for model design in ensemble formulation have been developed and tested using HPC Challenge resources. One method is the inclusion of a simple sea surface temperature (SST) model coupled to NOGAPS to account for diurnal SST variations. Currently, in the operational atmospheric forecasts, the SST is held fixed through the forecast integration. This simple diurnal temperature model is showing significant progress for improved forecasts of low-level wind-fields (McLay, et al., 2012). Another method to account for model uncertainty in ensemble design is through variation of parameter values in the parameterization of sub-gridscale processes. Reynolds, et al. (2011b) has shown that varying parameters within the ensemble can improve ensemble performance in a number of measures over the tropics. Figure 9 shows the impact of parameter variations on ensemble-mean tropical cyclone track errors as a function of forecast time for the 2009 season.

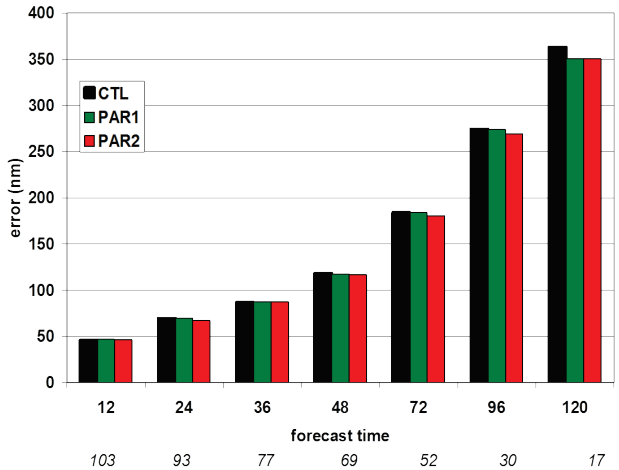


Figure 9. Homogenous NH ensemble-mean TC track forecast error (nm) for the control ensemble (without model uncertainty, CTL, black), an ensemble with parameter variations in the convection scheme only (PAR1, green), and parameter variations in the convection and boundary-layer schemes (PAR2, red). The numbers of verifying forecasts are shown below the x-axis.

The improvements to tropical cyclone track forecasts are small, but statistically-significant between 24 and 96 hours, with the inclusion of parameter variations in the convection and boundary layer parameterizations. Research on the inclusion of model uncertainty in ensemble design is ongoing, and it is anticipated that model uncertainty methods will be transitioned to the operational ensemble at FNMOC in FY13.

5d. Real-Time High-Resolution Ensemble Forecasts for the Hurricane Forecast Improvement Program

In addition to research and development, the HPC Challenge Project was also used to support real-time high-resolution (T319, or 42 km) 20-member NOGAPS ensembles during the 2010 and 2011 tropical cyclone seasons in support of the Hurricane Forecast Improvement Project. It was found that the ensemble-mean tropical cyclone track errors were approximately 20 nm better than the control deterministic operational forecast at the 3–5 day range. Research on the development of ensemble probabilistic products is also ongoing. An example of a product in development is shown in Figure 10 for Hurricane Danielle.

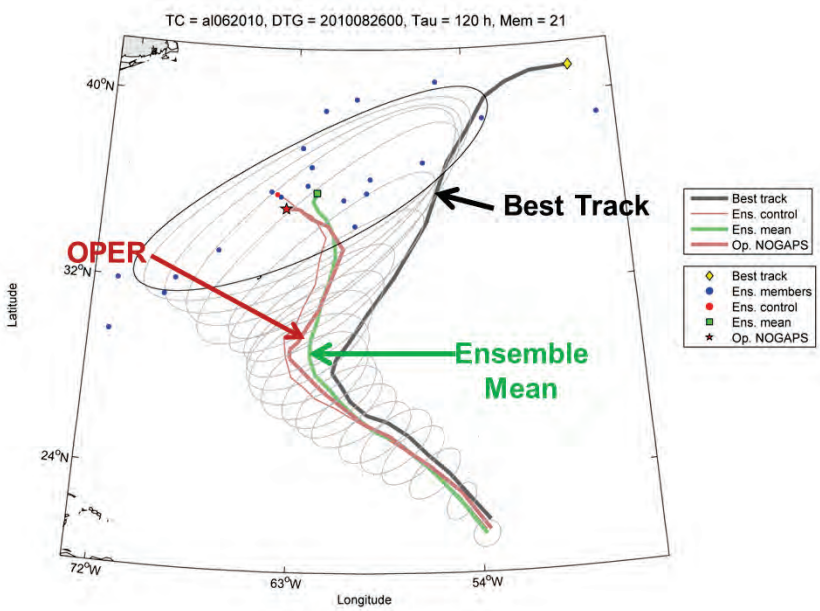


Figure 10. Ensemble forecasts for Hurricane Danielle from 26 August 2010. The observed "best track" is shown in black. The positions of the hurricane in each of the 20-member ensembles in the 120-h forecasts are shown by the blue dots. The operational deterministic NOGAPS forecast is indicated by the thick brown line. The ensemble-mean track forecast is given by the green line. Ellipses are drawn in gray at 6-h intervals to encompass two-thirds of the ensemble-member storms.

The ellipses in Figure 10 have been developed as a way to convey ensemble spread and forecast uncertainty information to the forecaster. Note that the ellipses are relatively small for early forecast times, when the ensemble mean and operational tracks are close to the verifying best track. When Danielle recurves towards the northwest, the operational deterministic forecast and ensemble-mean tracks diverge from the observed track, however, the ensemble does capture a significant increase in uncertainty in the forecast track at this time, particularly along the southwest-northeast direction (e.g., the ellipses became considerably larger). This indicates that, for this forecast, the ensemble divergence was capable of "warning" the forecaster that there was considerable uncertainty and potential error after the mid-point in the forecast. Ensemble-based forecast products and diagnostics are in continued development, and are also being used to provide feedback to model developers on forecast error characteristics.

6. Conclusion

In this paper, we have summarized recent progress in the development and testing of the Navy's next-generation tropical cyclone prediction system for the mesoscale, COAMPS-TC, and new capabilities for the Navy's global ensemble system using NOGAPS. An improved version of the COAMPS-TC has been developed in the past year and executed in real-time for multiple regions of the globe, including the W. Atlantic, W. Pacific, E. Pacific, C. Pacific, and S. Hemisphere

basins. Evaluation of these real-time forecasts performed on the DoD HPC reveals that COAMPS-TC was the most skillful dynamical model for tropical cyclone intensity forecasts in the W. Atlantic basin for the 2010–2011 seasons. The COAMPS-TC intensity skill is superior to other dynamical models for 36-h forecasts and beyond to 5 days. COAMPS-TC has comparable TC track skill as the current Navy's regional tropical cyclone model, GFDN.

Future advancements to COAMPS-TC are currently being tested. These improvements include new methods to initialize the tropical cyclone vortex, improved representations of crucial physical process such as turbulence and clouds, and new methods to evaluate COAMPS-TC forecasts to better diagnose systematic tropical cyclone forecast errors. New methods for data assimilation using Ensemble Kalman Filter and four-dimensional variational assimilation methods are currently being developed, and are expected to provide significant improvements to both track and intensity forecast skill for COAMPS-TC.

Several advances have been made in the global ensemble forecast system. Significant improvement is obtainable in the NOGAPS ensemble performance in the tropics for a variety of metrics through the inclusion of parameter variations, which reflect components of model error. In addition, significant improvement in ensemble-mean tropical cyclone tracks have been demonstrated through increased horizontal resolution, which was transitioned to operations in September 2011. Ongoing work on inclusion of a simple model of the diurnal cycle of sea surface temperature, and in methods for stochastic forcing, also show promise. It is anticipated that further upgrades to the global ensemble system (e.g., higher-resolution or inclusion of model uncertainty) will be transitioned to operations within the next year.

Acknowledgements

We acknowledge support through the Office of Naval Research's Program Element 0601153N. We also appreciate support for computational resources through a grant of high performance computing time from the HPCMP DoD Supercomputing Resource Centers at Stennis, MS and in Vicksburg, MS. COAMPS® is a registered trademark of the US Naval Research Laboratory.

References

McLay, J.G., C.H. Bishop, and C.A Reynolds, "A local formulation of the ensemble transform analysis perturbation scheme." *Weather and Forecasting*, 25, pp. 885–993, 2010.

McLay, J., M. Flatau, C.A. Reynolds, and T. Hogan, "Inclusion of prognostic sea surface temperature in the ensemble transform (ET) global ensemble prediction system of the U. S. Navy." *J. Geophysical Research*, 2012 (submitted).

Reynolds, C.A., J.A. Ridout, and J.G. McLay, "Examination of parameter variations in the U.S. Navy Global Ensemble." *Tellus*, 63A, pp. 841–857, 2011b.

Reynolds, C.A., J.G. McLay, J.S. Goerss, E.A. Serra, D. Hodyss, and C.R. Sampson, "The impact of resolution and design on ensemble performance in the tropics." *Mon. Wea. Rev.*, 139, pp. 2145–2155, 2011a.

Numerical Flow Analysis of Planning Boats

Kyle A. Brucker, Thomas T. O'Shea, and Douglas G. Dommermuth
Science Applications International Corporation, Naval Hydrodynamics Division, San Diego, CA
{kyle.a.brucker, thomas.t.o'shea, douglas.g.dommermuth}@saic.com

Abstract

The focus of this paper is to describe and document the recent effort to validate the computer code Numerical Flow Analysis (NFA) for the prediction of hydrodynamic forces and moments associated with deep-V planing craft. This detailed validation effort was composed of three parts. The first part focuses on assessing NFA's ability to predict pressures on the surface of a 10 degree dead-rise wedge during impact with an undisturbed free surface. Detailed comparisons to pressure gauges are presented for two different drop heights, 6 inches and 10 inches. Results show NFA accurately predicted pressures during the slamming event. The second part examines NFA's ability to match sinkage, trim and resistance from experiments performed on constant dead-rise planing hulls, detailed in Fridsma (1969). Simulations were preformed on two 20 degree dead-rise hull-forms of varying length to beam ratios (4 and 5) over a range of speed-length ratios (2, 3, 4, 5, and 6 knots-ft$^{-1/2}$). Results show good agreement with experimentally measured values, as well as values calculated using Savitsky's parametric equations, detailed in Savistsky (1964). The final part of the validation study focused on assessing how well NFA was able to accurately model the complex multiphase flow associated with high Froude number flows, specifically the formation of the spray sheet. NFA simulations of a planing hull fixed at various angles-of-roll (0 degrees, 10 degrees, 20 degrees, and 30 degrees) were compared to experiments from Judge (2012). Comparisons to underwater photographs illustrate NFA's ability to model the formation of the spray sheet and the free surface turbulence associated with planing boat hydrodynamics.

1. Introduction

Over the past few years, much progress has been made in the ability of computational fluid dynamics (CFD) to accurately simulate the hydrodynamics associated with a deep-V mono-hull planing craft, (Fu, et al., 2010; Broglio & Iafarati, 2010; Fu, et al., 2011). This paper describes a recent effort to assess NFA for the prediction of deep-V planing craft hydrodynamic forces and moments and evaluate how well it models the complex multiphase flows associated with high-Froude number flows, specifically the formation of the spray sheet. This detailed validation effort was composed of three parts. The first part focused on assessing NFA's ability to predict pressures on the surface of a 10 degree dead-rise wedge during impact with an undisturbed free surface. The second part examines NFA's ability to match sinkage, trim and resistance from experiments performed on constant dead-rise planing hulls, detailed in Fridsma (1969), and the final part of the validation study is focused on assessing how well NFA was able to accurately model the complex multiphase flow associated with high-Froude number flows, specifically the formation of the spray sheet. The format of this paper is as follows: Section 2 provides details and references for the NFA code; Section 3 details the Wedge Impact study; Section 4 covers the steady forward speed comparisons to the experiments of Fridsma (1969); Section 5 discusses the fixed-roll comparisons to Judge (2012); and Section 6 details the principle conclusions, followed by a summary of their significance to the Department of Defense (DoD).

2. Numerical Approach

The Numerical Flow Analysis (NFA) computer code is employed in this study. NFA solves the Navier-Stokes equations utilizing a cut-cell, Cartesian-grid formulation with interface-capturing to model the unsteady flow of air and water around moving bodies. The interface-capturing of the free surface uses a second-order-accurate volume of fluid (VOF) technique. NFA uses an implicit subgrid-scale model that is built into the treatment of the convective terms in the

momentum equations (Rottman, et al., 2010). A panelized surface representation of the ship hull (body) is all that is required as input in terms of body geometry. Domain decomposition is used to distribute portions of the grid over a large number of processors. The algorithm is implemented on parallel computers using FORTRAN 2003 and the Message Passing Interface (MPI). The interested reader is referred to Dommermuth, et al. (2007); O'Shea, et al. (2008); Brucker, et al. (2010), for a detailed description of the numerical algorithm and of its implementation on distributed memory high performance computing (HPC) platforms.

3. Wedge Drop Validation

The structural response of a planing boat during a slamming event is of interest to designers. Specifically, the fatigue life of the structural members of existing planing boat designs are observed to be shorter than would be expected, given current design tools. During a slamming event, the bottom of a planing hull experiences a severe pressure pulse as the spray root moves outboard from the keel. To quantify this high-speed, high-pressure pulse, a wedge drop experiment was performed, the details of which can be found in Lesar, et al. (2012). The wedge was instrumented with, among other things, an array of high-frequency pressure gauges. During the impact, these gauges captured the pressure over time at various distances from the keel.

To assess NFA's ability to accurately predict these pressures, simulations of the experiments were performed. Care was taken to insure the initial conditions and boundary conditions were matched to the experimental setup. The wedge was dropped from rest at 6 inches and 10 inches above the undisturbed free surface. The model tank and the NFA domain were 15 feet wide by 24 feet long by 14 feet 6 inches deep. The wedge weighed 165.2 pounds, and was 36 inches by 23.75 inches, with a dead-rise angle of 10.23 degrees. Figure 1 shows the bottom of the instrumented wedge alongside the NFA simulation's geometry and domain extents. Two symmetric pressure probes were located at 1.5 inches from the keel. A four-by-four array of pressure probes had lines of four probes at 5.385, 6.795, 8.205, and 9.615 inches from the keel. Each probe in the array was 1.41 inches from its neighbor longitudinally and transversely. For both drop heights, the NFA simulations used wall boundary conditions in each Cartesian direction. The number of grid points in the x, y and z directions was 1,536, 1,024, and 512, resulting in 805 million cells. Grid spacing near the wedge was 0.001L, where L is the wedge length or 0.035 inches. The non-dimensional time-step taken was 0.00005, resulting in an output frequency of 37.8 KHz. The simulations were run on Jade, a Cray® XT4 (Cray, Inc.), on 768 processors, and took 12 hours for the 6-inch, running for 2,000 time-steps, and 32 hours for the 10-inch drop height, running 6,000 time-steps. Figures 2 and 4 show image sequences from the 6- and 10-inch drop height simulations, respectively. The camera is looking up at the wedge from underneath the water surface. The wedge is painted with contours representing the pressure that NFA has calculated. Blue represents zero pressure while red indicates 15 psi. The pressure pulse can be seen moving outboard from the keel over the pressure probes indicated by white dots on the bottom of the wedge on the right side. This pulse is concentrated at the spray root, which can be seen as the translucent surface that begins to break up later in the simulation.

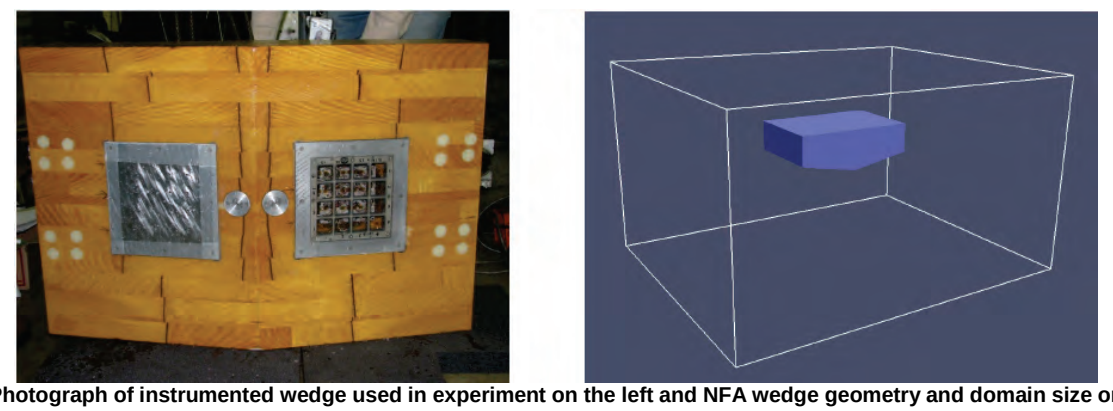


Figure 1. Photograph of instrumented wedge used in experiment on the left and NFA wedge geometry and domain size on the right

The output from the pressure probes is plotted against the pressure sampled from NFA in Figures 3 and 5, for the 6- and 10-inch drop heights, respectively. The x-axis is time in seconds, and the y-axis is gauge pressure in psi. Pressure-time histories from three completely separate drops are plotted together, with the peak from the first row of gauges aligned. The dashed black lines show five peaks in time, the first from the symmetric keel gauges and the remaining four from the four rows of pressure gauges in the array. Since the pressure pulse is quasi-two-dimensional, each row of pressure gauges in the keel-wise direction shows approximately the same time history. The time histories exhibit variability associated with minor

experimental irregularities. The blue error bars represent the minimum and maximum of the experimental peaks for the three drops. The sampling frequency of the pressure probes was 19.2 KHz, while NFA pressures were extracted at a rate of 37.8 KHz.

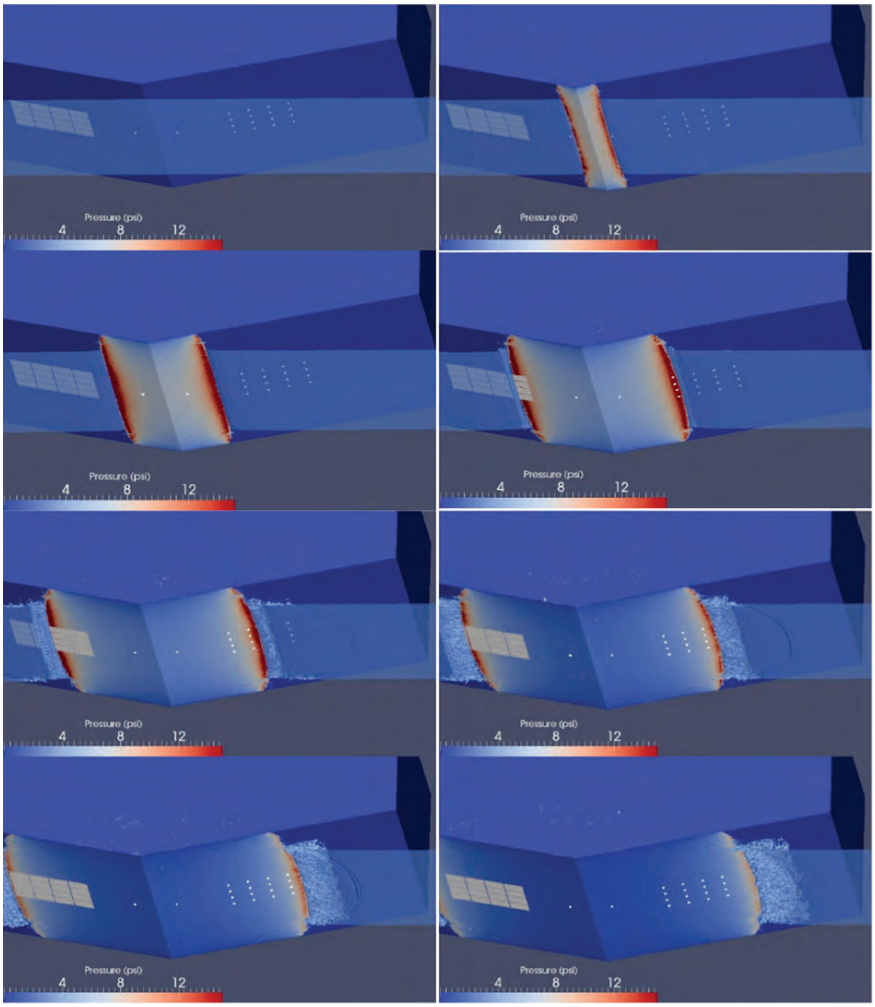


Figure 2. Sequences of frames from an animation of the 6-inch drop height wedge impacting the free surface. Pressure is interpolated onto the wedge and displayed, blue indicating 0 pressure, and red indicating 15 psi. Note the pressure wave from the spray root expanding transversely over the pressure gauges, represented by white circles.

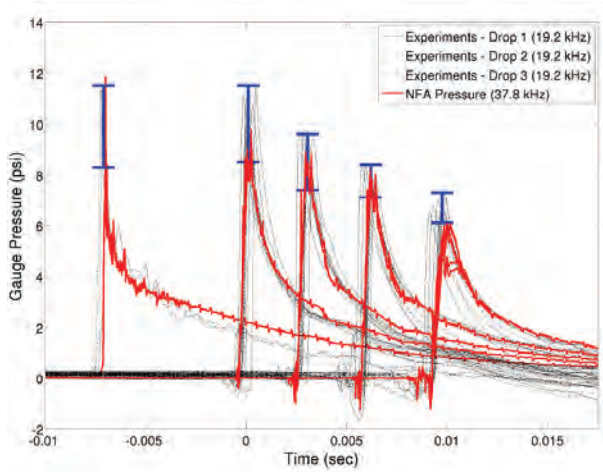


Figure 3. Six-inch drop height case: NFA pressure predictions plotted together with pressure gauge output from three separate drops

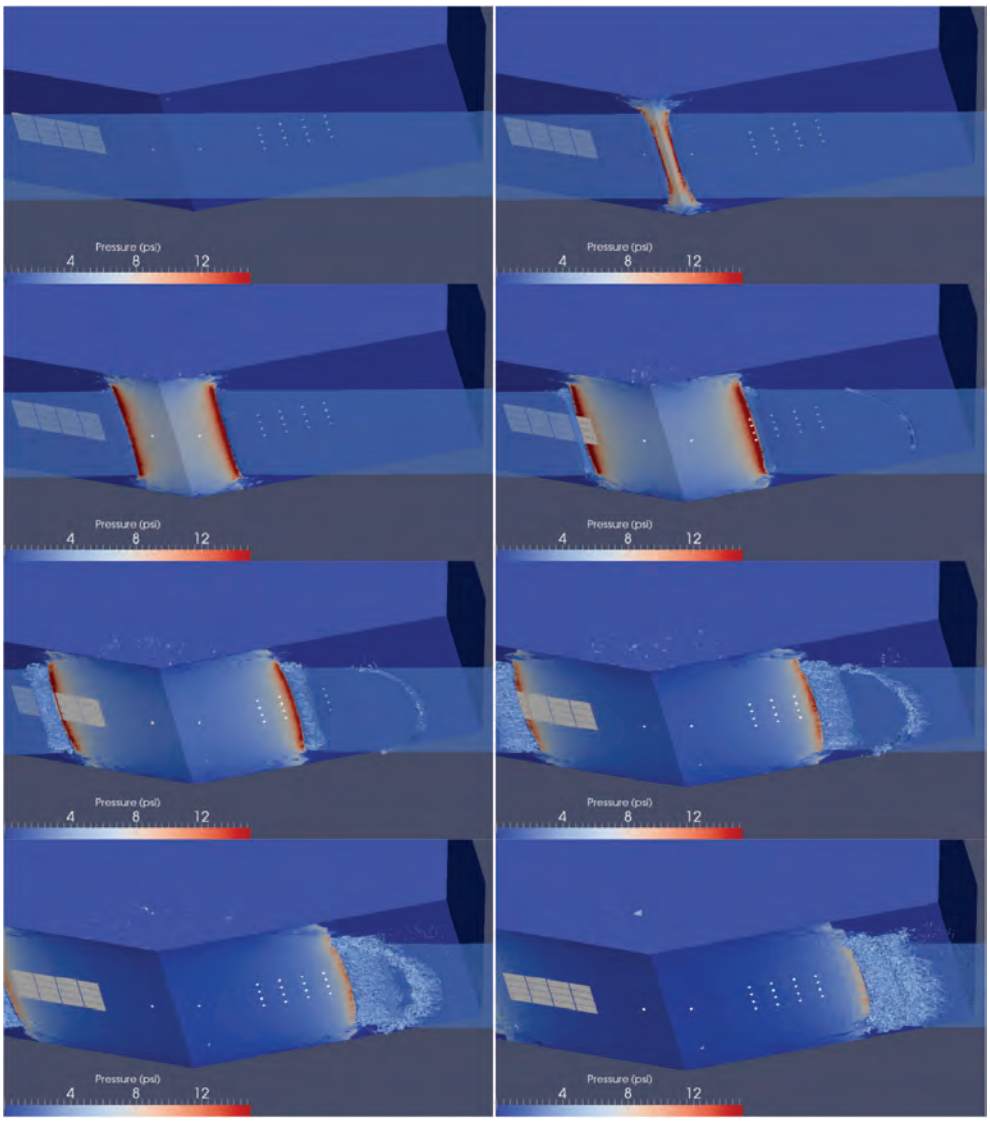


Figure 4. Sequences of frames from an animation of the 10-inch drop height wedge impacting the free surface. Pressure is interpolated onto the wedge and displayed, blue indicating 0 pressure, and red indicating 15 psi. Note the pressure wave from the spray root expanding transversely over the pressure gauges, represented by white circles.

The peak from NFA's first row of gauges was aligned to the peak from the experiments and plotted in red. NFA shows excellent agreement for peak pressure magnitudes and pulse duration. The spray root takes approximately 10 milliseconds (ms) to travel across the array of gauges, and each peak pressure is maintained for less than 1 ms. The propagation in time of the pressure pulse from the spray root is slowed by the deceleration of the wedge due to the impact with the free surface. The length of time between the final two is, therefore, greater than that of the preceding peaks. NFA correctly predicts this momentum transfer and thus the position in time of all four pressure peaks. NFA is within experimental variability for peak pressure and timing. It is notable that NFA also predicts the brief negative pressures seen in the experiments before the peak pressure due to the turbulent breakup of the expanding spray root.

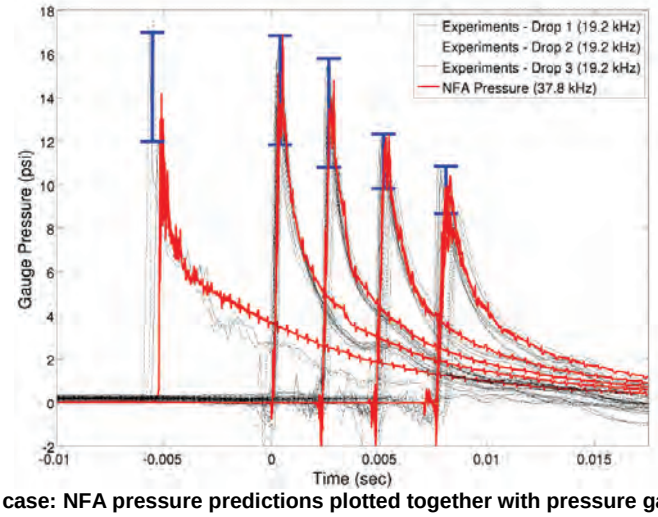


Figure 5. Ten-inch drop height case: NFA pressure predictions plotted together with pressure gauge output from three separate drops.

4. Steady Forward Speed Comparisons

The planing hull model used in the experiments of Fridsma (1969) is employed for this study of constant dead-rise planing hull-forms, in order to verify numerical prediction of sinkage, trim and resistance. Three configurations with different hull length (L) to beam (b) ratios, L/b=4 and L/b=5 (shown on the left in Figure 6), and light/heavy displacements are investigated. In all simulations, the model is of unit length with the bow of the model is fixed at x=0, and the flow is from positive x. The model was free to sink and trim about the center of gravity (CG). The vertical center of gravity was fixed at 0.294b above the keel, while the longitudinal center of gravity (LCG) was fixed at 0.6, 0.655, or 0.7L from the bow. Inflow and outflow boundary conditions are used in the stream-wise (x) direction, and free-slip conditions are used in the span-wise (y) and cross-stream (z) directions. The inflow condition is a free-stream current, and the outflow condition is one-dimensional, non-reflective, Orlanski-type boundary condition. The Froude number is $Fr=U(gL)^{-1/2}$, where U and L are, respectively, the speed and length of the planing hull model, and g is the acceleration of gravity, resulting in Froude numbers between 0.6 and 1.8. The extents of the computational domain, shown in Figure 6 (right), in the stream-wise (x), span-wise (y), and cross-stream (z) directions are respectively [-3.0L, 1.0L], [-1.6L, 1.6L], and [-1.25L, 1.0L]. These dimensions were chosen to match the Davidson Laboratory's Tank 3 dimensions (Bruno 1993). The number of grid points [nx, ny, nz] are [768, 512, 384]. The grid is stretched with nearly uniform spacing around the hull where the grid spacing is [0.00304L, 0.00306L, 0.00305L]. The maximum grid spacing far away from the ship along the Cartesian axes is [0.016L, 0.0204L, 0.0152L]. The grid points are distributed in 64×64×64 blocks over 576 cores, and the time-step is Δt=0.0005. All simulations have been run on the Cray® XE6 (Cray, Inc.), Raptor, platform located at the US Army Engineering Research and Development Center (ERDC), and run by the US Air Force Research Laboratory (AFRL).

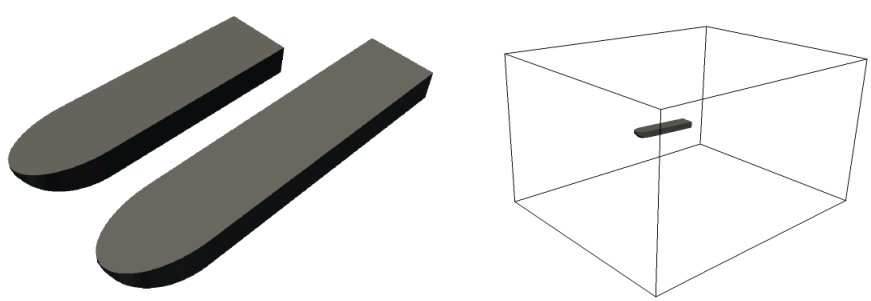


Figure 6. Setup L/b=4 and L/b=5 hull geometries (left) and L/b=5 model in computational domain (right)

Figures 7 through 9 are comparisons of the rise at the CG, or sinkage, on the left, the trim in the middle, and the ratio of the resistance to the displacement on the right. A ▲ denotes the experimental measurements made by Fridsma (Fridsma, 1969); the ● denotes the predicted value according to Savitsky (Savitsky , 1964); and the ■ denotes the value predicted by the NFA simulations. The NFA simulations were averaged for four boat-lengths, from non-dimensional time

of four to eight for Figures 7 and 8, and over the last two boat-lengths for Figure 9. The error bars show the minimum and maximum values over the averaging period for the rise at the CG and trim, and ± the *r.m.s.* for the resistance. The comparison of interest here is between the experiments and the NFA simulations; the Savitsky predictions are simply provided for comparison. Figure 7 shows the results of the comparison for C_Δ=0.608, L/b=4, and β=20°, with the LCG at 0.6L from the bow, where C_Δ is the load coefficient defined as $C_\Delta=\Delta/(wb^3)$, Δ is the hull displacement, w is the specific weight of water, b is the beam of the hull, L is the length of the hull, and β is the dead-rise angle. The rise at the CG, shown in units of hull length, is shown on the left at the slower speed ratios, $VL^{-1/2}$ [knots-ft$^{-1/2}$], of 2 and 3, the results of the simulations are identical to those recorded during the experiments. At higher speed ratios of 4, 5, and 6, the rise at the CG is under-predicted by the simulations. The trim, shown in the middle frame, also shows good agreement with the experiments, although again at the higher speeds the simulations under-predict the final trim angle by around 10 percent. It is noted that the general shape of the curve matches the shape of the curve recorded in the experiments and also the predictions from Savitsky. The resistance, shown on the right, is under-predicted for all but the highest-speed case, where it is over-predicted. In all three frames of Figure 8, the error bars are small, meaning that the simulations predicted a nearly constant value for the quantities of interest. No porpoising was observed.

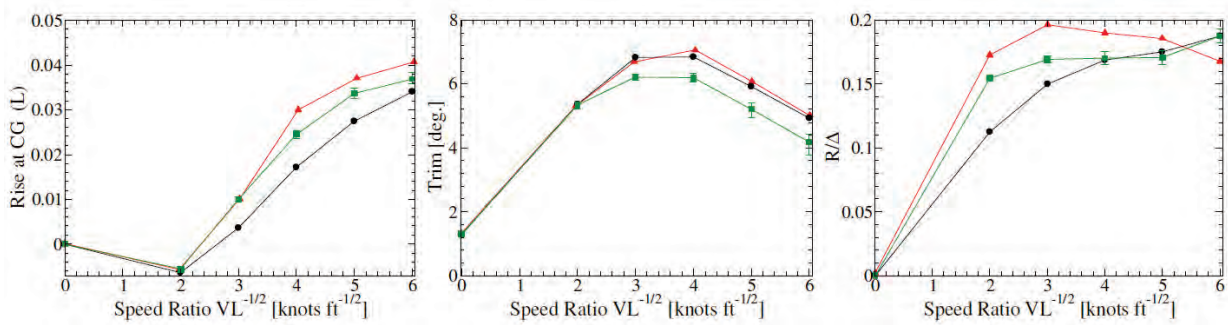


Figure 7. Fridsma comparison for C_Δ=0.608, L/b=4 and β=20°. ▲, Fridsma experiments (Fridsma, 1969); ●, Savitsky predictions (Savitsky, 1964); ■, NFA simulations. Rise at the CG (left), trim (middle), and resistance (right).

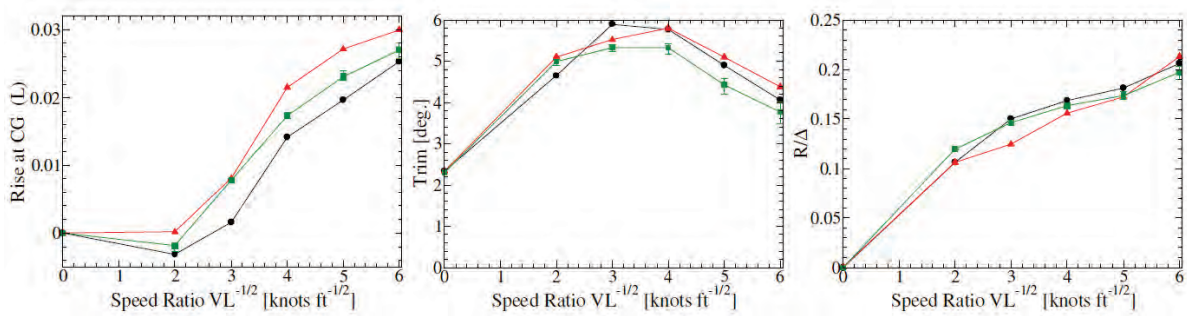


Figure 8. Fridsma comparison for C_Δ=0.608, L/b=5 and β=20°. ▲, Fridsma experiments (Fridsma, 1969); ●, Savitsky predictions (Savitsky, 1964); ■, NFA simulations. Rise at the CG (left), trim (middle), and resistance (right).

Figure 8 shows the results of the comparison for C_Δ=0.608, L/b=5 and β=20°, with the LCG at 0.655L from the bow. The rise at the CG is well-predicted by the numerical simulations, with the agreement being nearly identical at the lower-speeds. There is good qualitative agreement at the higher-speeds, but again the numerical simulations under-predict the rise at the CG at the higher-speeds. The trim is well-predicted by the numerical simulations, being nearly identical to the experiments for speed ratios of 2 and 3. For the higher-speed ratios, those above 3, the trim is under-predicted but the shape of the curve is similar to the one recorded in the experiments and the one predicted by Savitsky. The resistance displays the opposite trend to the L/b=4 case. Here, it is over-predicted by the numerical simulations for speed ratios between 2 and 5, and then under-predicted for a speed ratio of 6. No porpoising was observed.

Figure 9 shows the lighter displacement case with for C_Δ=0.304, L/b=4 and β=20°, with the LCG at 0.7L from the bow. The rise at the CG is under-predicted over the entire speed range. Note the large error bars at the speed ratio of 5, the speed at which the model displayed evidence of porpoising in the experiments, and no data was recorded. The trim again shows excellent agreement with the value recorded in the experiments for the lowest speed ratio but is under-predicted for speed ratios of 3 and 4. The resistance, shown on the right, is nearly identical to the values recorded in the experiments. Porpoising was observed at a speed ratio of 5, which was consistent with the experiments.

Figure 10 shows the trim as a function of time. For speed ratios of 2 and 3, a steady trim is achieved after only two boat lengths. The speed ratio of 4 achieves a quasi-steady trim after about five boat lengths. However, the speed ratio of 5 does not achieve a steady trim, and furthermore, the amplitude of the oscillations does not decay with time.

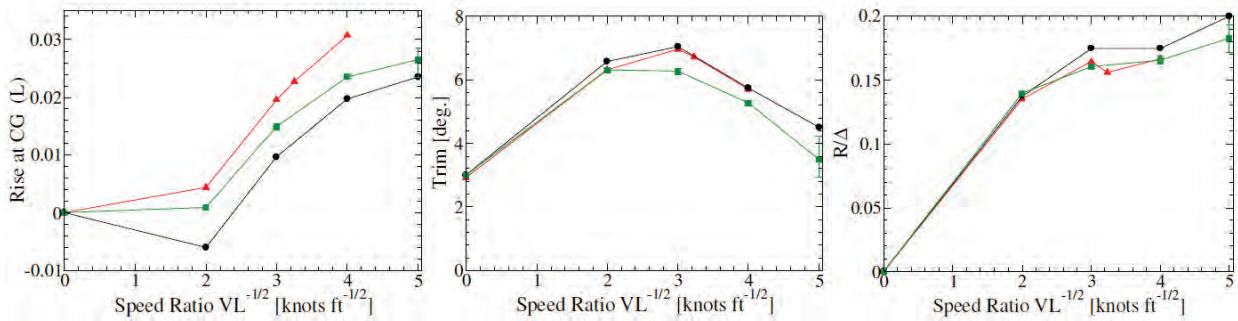


Figure 9. Fridsma comparison for C_Δ=0.304, L/b=4 and β=20°. ▲, Fridsma experiments (Fridsma, 1969); ●, Savitsky predictions (Savitsky, 1964); ■, NFA simulations. Rise at the CG (left), trim (middle), and resistance (right).

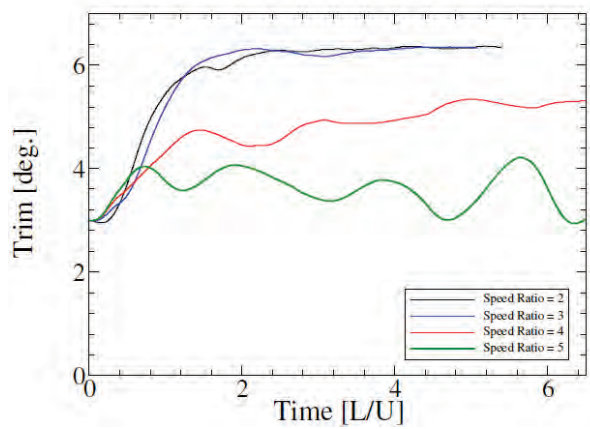


Figure 10. Trim versus time for C_Δ=0.304, L/b=4 and β=20°. Note the oscillatory behavior at a speed ratio of 5 (green line), indicative of porpoising. [Speed ratio is in knots-ft-$^{1/2}$]

Figure 11 shows instantaneous pressures on the bottom of the hull for the C_Δ=0.608, L/b=4 and β=20°, with the LCG at 0.6L from the bow for speed ratios of 2 through 6 moving from left to right. The transition from displacement to planing is evidenced by the sharpening of the spray root and lack of hydrostatic pressure aft of the spray root, which can be observed by comparing the leftmost and rightmost frames. The hot spots are transitory in nature and would not be observed if time-averaged pressures were shown. Given their transitory nature, they do not significantly contribute to the force balance which determines the position of the hull.

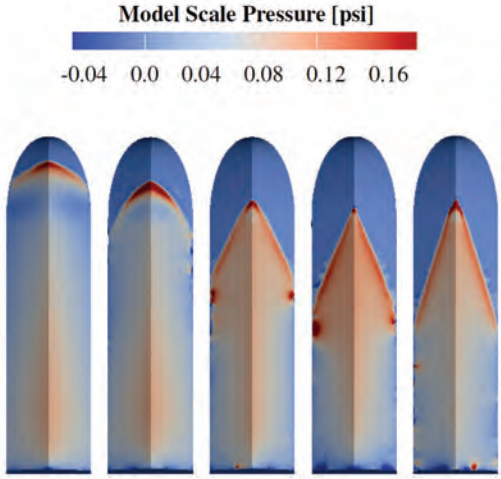


Figure 11. Pressures on hull for C_Δ=0.608, L/b=4 and β=20°; Speed ratio=2, 3, 4, 5, 6 (from left to right)

6. Fixed Roll Comparisons

Planing boats at high-speed in a developed seaway may experience many types of dynamic instability, such as porpoising, chine-walking, steady heeling, and bow diving. To better understand the dynamic stability and hydrodynamics of planing boats, a series of forced roll experiments were performed, the details of which are presented in Judge (2012). The first section of this test focused on steady forward speed, at fixed sinkage and trim, at a series of different roll angles. NFA simulations were performed at these conditions and comparisons to the underwater photographs are presented in this section.

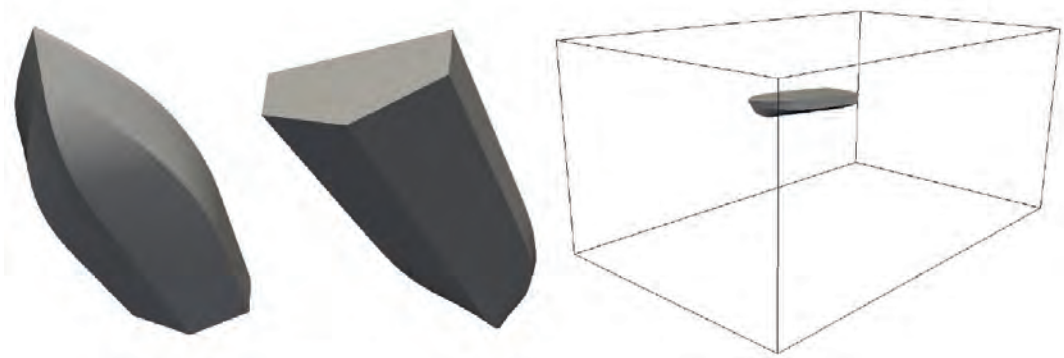


Figure 12. USNA constant deadrise hull-form (left) and NFA setup with geometry and domain extents (right)

The United States Naval Academy (USNA) geometry and the corresponding NFA domain can be seen in Figure 12. This constant dead-rise hull-form had a length overall (LOA) of 5 feet and a beam of 1.48 feet. Four cases of varying roll angles (0 degrees, 10 degrees, 20 degrees, and 30 degrees) at 20 feet per second were investigated. The geometry was positioned at a trim of 4 degrees and a draft at the transom of 3.574 inches, and then rolled around an axis parallel to the keel at the vertical center of gravity (VCG) of 5.29 inches. The experiments were performed at the Naval Academy's Hydromechanics Laboratory 380-foot Tow Tank, which are 16 feet deep and 26 feet wide. The simulation was run with a width of two boat lengths, or 10 feet, and a depth of one boat length, or 5 feet. NFA's domain had to be smaller than the towing tank to cluster cells near the body without stretching the grid too heavily. This smaller domain may have resulted in some blockage, and further simulations need to be performed to investigate the effect of domain size. The number of cells in x, y, and z was 1,280, 896, and 448, resulting in 514 million cells in the simulation. Spacing near the body was 0.0009L or 0.054 inches, necessitating a non-dimensional time-step of 0.00023. Simulations were run for 13,000 time-steps, or three body lengths, which was sufficient to reach steady-state, on Garnet, a Cray® XT6 (Cray, Inc.), at ERDC, and took approximately 24 hours to complete using 576 processors.

Figures 13 through 16 show the underwater photographs taken during the experiments compared to a similar view of the NFA simulations. The spray root line, and thus wetted surface area, agrees well to the experiments. The generation of the spray sheet is correctly predicted; the spray edge matches well to experiments. NFA also models the instabilities in the spray sheet breakup reasonably well. As the model rolls over, the experimental photographs show the flow over the chine initially separating cleanly, but at 20 degrees and higher, the flow wraps around the chine and wets the side of the model.

The NFA simulations exhibit this behavior as well. The bottom of the model was painted with a ruler and the length along the keel and chines from the transom was estimated from the underwater pictures by the experimentalist. The same values were extracted from the NFA results, and the comparison is plotted as Figure 17. The agreement is excellent throughout the range of roll angles; it follows that wetted surface area is being accurately predicted as well.

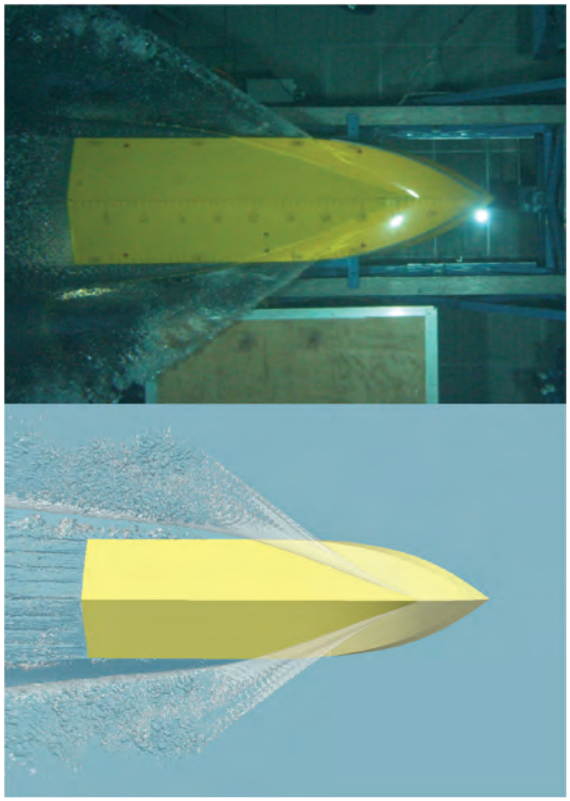


Figure 13. USNA underwater photograph compared to NFA simulation; 0 degrees roll, U=20 ft/sec

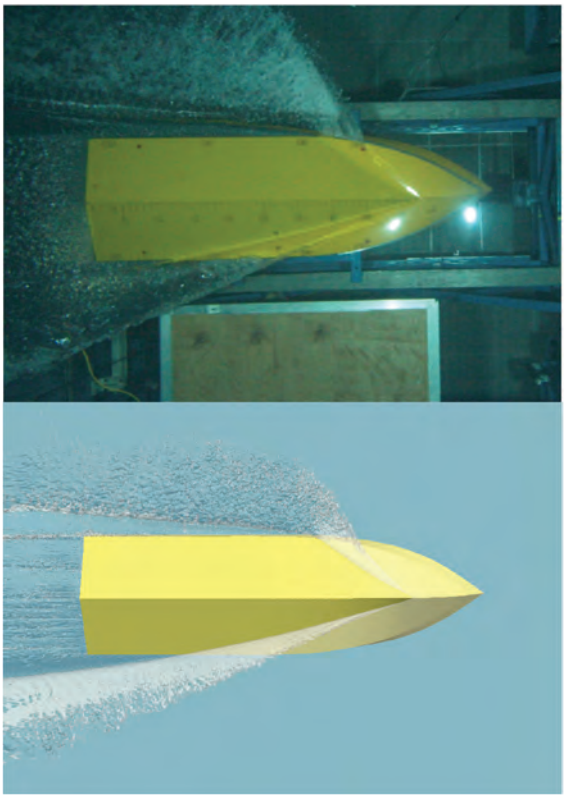


Figure 14. USNA underwater photograph compared to NFA simulation; 10 degrees roll, U=20 ft/sec

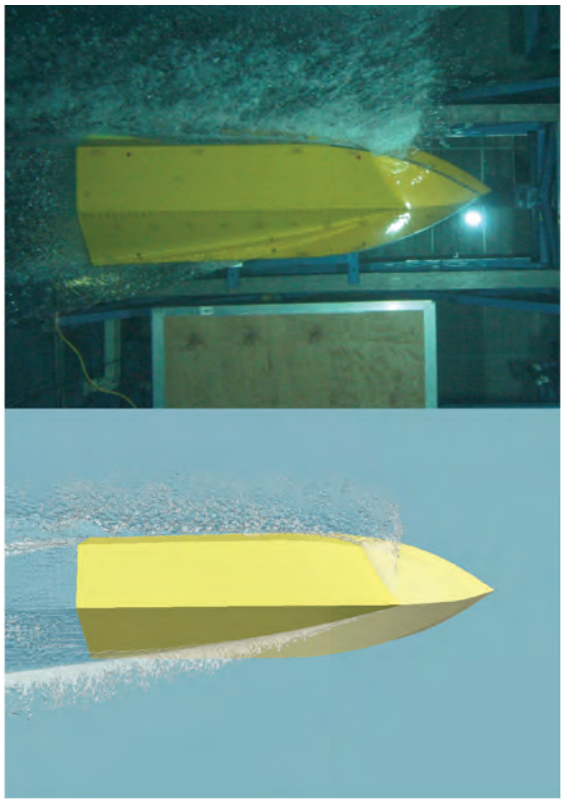


Figure 15. USNA underwater photograph compared to NFA simulation; 20 degrees roll, U=20 ft/sec

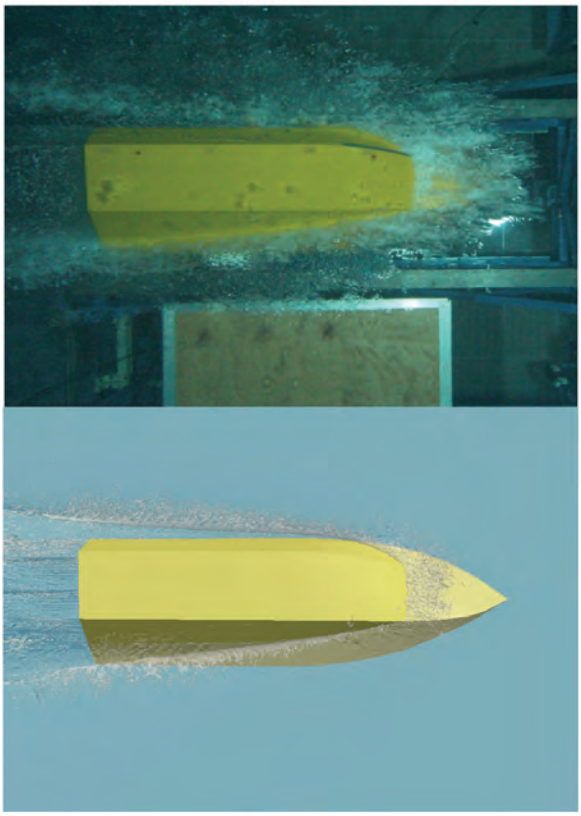


Figure 16. USNA underwater photograph compared to NFA simulation; 30 degree roll, U=20 ft/sec

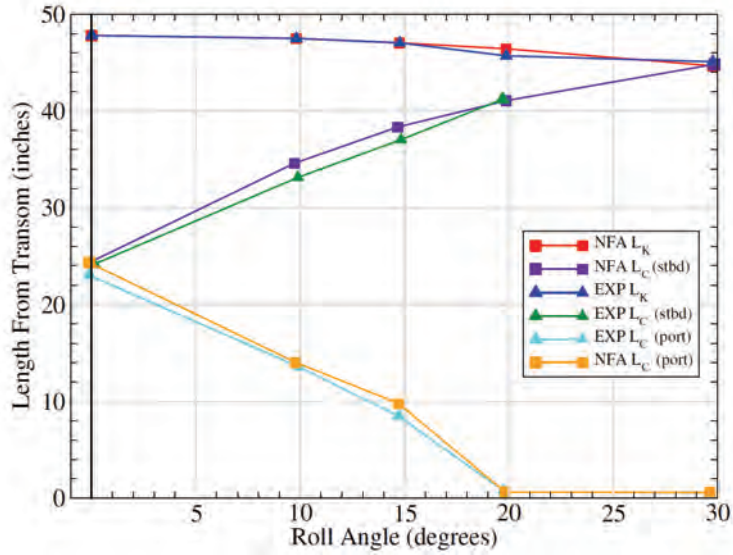


Figure 17. Comparison of NFA results to experiments for length along the keel, length along the starboard chine, and length along the port chine for each roll angle simulated. Lengths are measured in inches from the transom.

7. Conclusion

NFA is poised to be a useful tool for evaluating novel planing hull-forms. As shown, the ability of NFA to accurately predict pressures on a planing hull during a slamming event provides design criteria for structural analyses. Sinkage, trim, and resistance predictions have also been demonstrated to be sufficiently accurate to provide designers with a useful tool. NFA is also capable of predicting complex multiphase flow associated with high-Froude number flows.

In addition to advancing the state of complex hydrodynamic modeling, NFA provides a remarkable ease of use. The ability of NFA to input complicated geometries without a manual gridding requirement allows for the testing of a series of hull-form permutations simultaneously. Leveraging increasingly accessible high performance computing resources, NFA can be used to run a matrix of cases quickly. Due to excellent scaling performance, runs can be spread across greater numbers of processors to drastically reduce calendar time necessary.

8. Significance to DoD

The complex interactions of naval combatants with the ocean environment and the resulting impact on the performance of and threats to naval combatants are some of the main challenges remaining in modern naval ship design. Recent software developments and hardware advances provided by the HPC program have enabled computational ship hydrodynamics, bringing us closer to meeting this challenge.

Acknowledgements

The authors would like to acknowledge Richard for providing the high-quality wedge impact data used in this paper. This work is supported by the Office of Naval Research (ONR), program manager Bob Brizzolara. The numerical simulations have been performed on the Cray XE6 platforms located at the US Army Engineering Research and Development Center (ERDC) and the US Air Force Research Laboratory (AFRL). NFA research and development has been sponsored over the years by ONR and the Naval Surface Warfare Center, Carderock Division (NSWCCD). The program managers are Patrick Purtell, Ph.D., Steven Russell, Ph.D., Paul Hess III, Ph.D., Ronald Joslin, Ph.D., and Thomas Fu, Ph.D. SAIC Internal Research and Development (IR&D) funding also supports the development of NFA. This work is also supported in part by a grant of computer time from the Department of Defense High Performance Computing Modernization Program (HPCMP), (http://www.hpc.mil/). Animations of NFA simulations are available at http://www.youtube.com/waveanimations, and http://www.saic.com/maritime/nfa/.

References

Broglia, R. and A. Iafrati, "Hydrodynamics of Planing Hulls in Asymmetric Conditions." *Proceedings of the 28th Symposium on Naval Hydrodynamics*, Pasadena, CA, September 12–17, 2010.

Brucker, K.A., T.T. O'Shea, and D.G. Dommermuth, "Three-Dimensional Simulations of Deep-Water Breaking Waves." *Proceedings of the 28th Symposium on Naval Hydrodynamics*, Pasadena, CA, USA, 2010.

Bruno, M.S., "Davidson Laboratory and the Experimental Towing Tank: The History of Towing Tank Research at Stevens." *Society of Naval Architects and Marine Engineers*, New York Metropolitan Section, 1993 (book).

Dommermuth, D.G., T.T. O'Shea, D.C. Wyatt, T. Ratcliffe, G.D. Weymouth, K.L. Hendrikson, D.K. Yue, M. Sussman, P. Adams, and M. Valenciano, "An application of cartesian-grid and volume-of-fluid methods to numerical ship hydrodynamics." *Proceedings of the 9th International Conference on Numerical Ship Hydrodynamics*, Ann Arbor, MI, USA, 2007.

Fridsma, G., "A systematic study of the rough-water performance of planing boats." *Davidson Laboratory Report R-1275*, 1969.

Fu, T.C., R. Akers, T.T. O'Shea, K.A. Brucker, D.G. Dommermuth, and E. Lee, "Measurements and Computational Predictions of a Deep-V Monohull Planing Hull." *11th Int. Conf. on Fast Sea Transportation (FAST 2011)*, Honolulu, HI, USA, September 26–19, 2011.

Fu, T.C., T. Ratcliffe, T. O'Shea, K. Brucker, R. Graham, D. Wyatt, and D. Dommermuth, "A Comparison of Experimental Measurements & Computational Predictions of a Deep-V Planing Hull." *Proceedings of the 28th Symposium on Naval Hydrodynamics*, Pasadena, CA, September 12–17, 2010.

Judge, C.Q., "Static and Dynamic Forces and Wetted Lengths for a Planing Hull Model Forced in Roll." *The Third Chesapeake Power Boat Symposium*, Annapolis, MD, June 15–16, 2012.

Lesar, D.E., M. Jiang, B. Lien, A. Engle, and R. Lewis, "Validation of Numerical Analysis Codes for Hydrodynamic Impact Simulation Using Rigid Wedge Drop Test Data." *Draft NSWCCD Technical Report*, 2012.

O'Shea, T.T., K.A. Brucker, D.G. Dommermuth, and D.C. Wyatt, "A numerical formulation for simulating free-surface hydrodynamics." *Proceedings of the 27th Symposium on Naval Hydrodynamics*, Seoul, Korea, 2008.

Rottman, J.W., K.A. Brucker, D.G. Dommermuth, and D. Broutman, "Parameterization of the internal wave field generated by a submarine and its turbulent wake in a uniformly stratified fluid." *Proceedings of the 28th Symposium on Naval Hydrodynamics*, Pasadena, CA, USA, 2010.

Predicting Tides in a High Resolution Ocean General Circulation Model

James G. Richman, E. Joseph Metzger, Jay F. Shriver, and Alan Wallcraft
US Naval Research Laboratory, Ocean Dynamics & Prediction Branch (NRL-SSC), Stennis Space Center, MS
{james.richman, hurlburt, joe.metzger, jay.shriver, alan.wallcraft}@nrlssc.navy.mil

Ole Martin Smedstad
QinetiQ North America, Slidell, LA
ole.smedstad.ctr@nrlssc.navy.mil

Brian K. Arbic
Department of Earth and Environmental Sciences, University of Michigan, Ann Arbor, MI
arbic@umich.edu

Abstract

The ocean tides and the atmospherically-forced oceanic general circulation with its associated mesoscale eddy-field typically are modeled separately. The US Naval Research Laboratory now is modeling the tides and general circulation concurrently in a high-resolution version of the HYbrid Coordinate Ocean Model (HYCOM). The barotropic, or surface tide in the concurrent model compares favorably with other non-assimilative tidal models, but not as well as assimilative tidal models. The presence of horizontally varying density stratification with the addition of atmospheric-forcing allows the generation of internal tides (internal waves of tidal frequency), which are qualitatively different from earlier global internal tide models, where atmospheric-forcing and horizontally variable stratification are absent. The internal tide amplitudes in the new simulations compare well with along-track satellite altimeter data. Errors in the surface tide are nearly evenly split between errors in amplitude and errors in phase. The spatial distribution of the surface tidal amplitude and phase errors are very different. The surface tide generates the internal tide with errors in the surface tide translating into errors in the internal tide. A new technique of a ported boundary condition is used to represent the tides under floating ice shelves in the Arctic and Antarctic, with significant improvement in model skill. HYCOM is the basis for the global ocean forecast system. In the forecast system, data is assimilated to correct the ocean circulation. Initial tests show that the data assimilation technique developed for the mesoscale ocean variability works in the concurrent model with the tides without significant changes to the tides.

1. Introduction

Tides in the ocean are forced by the difference in the gravitational potential of the Moon and Sun over the rotating Earth (Cartwright, 1999). The tidal potential, or equilibrium tide, has a simple shape, but the actual ocean tides have a complex dynamical response to the equilibrium tidal-forcing. Early global tidal models (e.g., Hendershott, 1972) were barotropic shallow water models including the effects of several factors, the Earth's rotation, friction with the seafloor, water depth (which controls the speed of shallow water gravity waves), the presence of continents; as well as seemingly exotic effects such as tides in the solid Earth and perturbations in the gravitational self-attraction of mass in the ocean and Earth brought about by tidal motions. In a shallow water model, the horizontal velocities do not change in the vertical, but do change with latitude and longitude.

In the presence of density stratification tidal flow over topographic features generates internal waves with tidal frequency (e.g., Wunsch, 1975). Internal tides can have large displacements, greater than 50 m, and current speeds greater than 2 m/s. The vertical motions of internal tides can lead to grounding of submarines and damages to oil drilling. Breaking of internal waves is a leading source of mixing, and internal tides may be one of the largest energy sources powering mixing in the deep ocean. Thus, tides may provide an important control on the stratification and circulation of the deep ocean (Munk and Wunsch, 1998).

Internal tides have much smaller horizontal scales than the surface barotropic tides. The models used to describe the surface tides can't be used for internal tides. Global modeling of internal tides is less than a decade old, with only a few papers published (Arbic, et al., 2004; Simmons, et al., 2004; Hibiya, et al., 2006; Simmons, 2008; Arbic, et al., 2010). The earliest global internal tide models were forced by the astronomical tidal potential only with a horizontally-uniform oceanic stratification based upon the subtropical oceanic stratification. For these models, the internal tides in Polar Regions are notably inaccurate (Padman, et al., 2006), with strong tides in regions where the ocean actually is very weakly stratified.

Internal tides are generated by the interaction of the barotropic tides with bottom topography, which, in turn, feeds back onto the barotropic tides. Inferences from data-assimilative tide models indicate that about one-third of the global tidal energy dissipation occurs over regions of rough topography, where internal tides are generated (Egbert and Ray, 2000). Thus, the generation of internal tides acts as a damping mechanism for the barotropic tides.

The conversion of the barotropic surface tide into internal tides depends upon the roughness of the sea floor and the density stratification at the sea floor. Modeling both the surface and internal tides requires accurate bathymetry and a stratified ocean. The stratification in the ocean is determined by the atmospheric-forcing, which can be provided by a general ocean circulation model. The US Naval Research Laboratory (NRL) global ocean circulation model, the HYbrid Coordinate Ocean Model (HYCOM), has been modified to included tidal potential forcing as well as atmospheric-forcing to allow concurrent simulation of the ocean tides and the ocean circulation and stratification as described by Arbic, et al. (2010). The global model, presently, has 32 layers in the vertical, and runs at a horizontal resolution of approximately 9 km at the equator. Thus, only part of the conversion of the barotropic surface tide into internal tides can be directly estimated in the model, and the internal waves with small vertical and horizontal scales must be parameterized as in both the barotropic shallow water models and the global ocean circulation model.

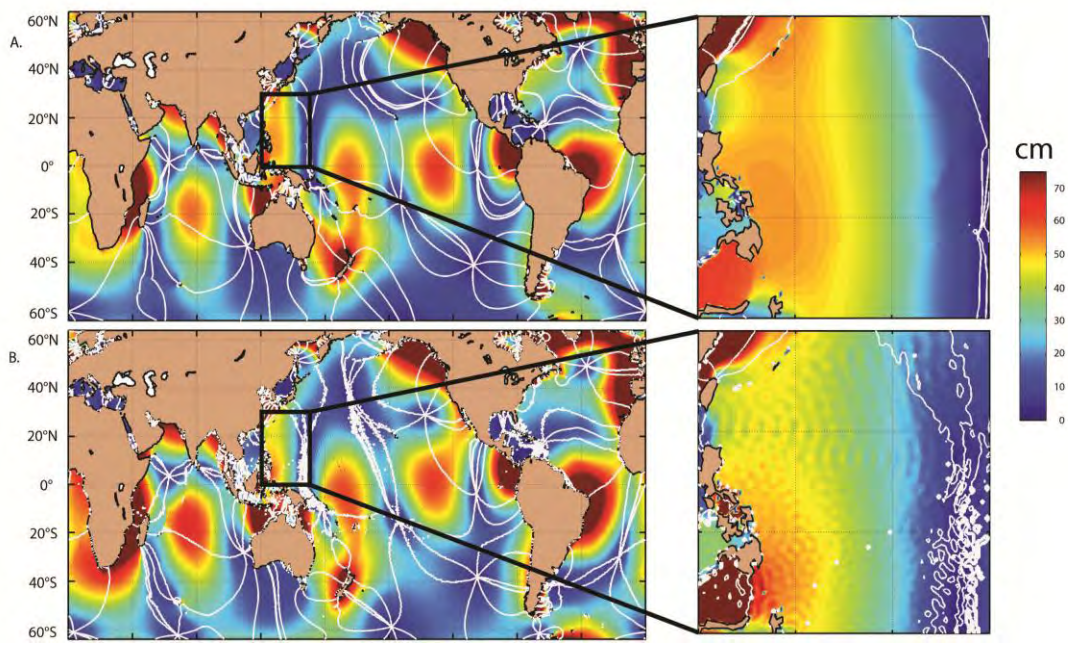


Figure 1. M_2 surface tidal elevation amplitude (cm) from (A) a barotropic tidal model constrained by satellite altimetry [TPXO7.2, adapted from Egbert, et al., 1994] and (B) the HYCOM simulation. Tidal phase is shown with white lines. Inset boxes in the figures expand the amplitude and phase in the Western Pacific, where strong internal waves can be seen in the tidal elevations.

The surface elevation amplitude of the principal lunar semidiurnal tide M_2, the largest tidal constituent in the ocean, is shown in the left-hand side panels of Figure 1. The upper-left panel (Figure 1a) displays the M_2 amplitude and phase from a data-assimilative barotropic tide model (TPXO7.2, Egbert et al., 1994) and the lower-left panel (Figure 1b) displays the M_2 amplitude and phase in a new concurrent HYCOM simulations. The two models are qualitatively similar with an rms difference of 7.5 cm. In regions where internal waves are large, such as the Western North Pacific Ocean indicated by the inset boxes in the figures, the internal tides can be seen in the HYCOM simulation (Left panel of Figure 1b) as small-scale bands in the amplitude and "wiggles" in the lines of constant phase (white lines in Figures 1a and 1b).

2. The Global Ocean General Circulation Model

HYCOM is currently being developed as part of the global ocean forecast system for the United States Navy (Chassignet, et al., 2007; Metzger, et al., 2010; http://hycom.org; http://www7320.nrlssc.navy.mil/GLBhycom1-12/). The Navy needs to know the state of the ocean—quantities such as the strength of currents, height of the water, clarity of the water and sound propagation properties amongst others—on a range of time and space scales at diverse locations around the globe. The model is forced by fluxes obtained from the Navy Operational Global Atmospheric Prediction System (NOGAPS; Rosmond, et al., 2002). The data-assimilative version of the model utilizes satellite altimetric sea surface height, radiometric sea surface temperature, and *in situ* ship, float and profile temperature and salinity data using the Navy Coupled Ocean Data Assimilation system (NCODA; Cummings, 2005) to provide a statistical blending of model and observations. The global ocean model runs on a combination Mercator and tripolar grid with an equatorial grid resolution of 0.08° (8.9 km), and is coupled to a sea ice model at high-latitudes. Simulations of HYCOM at very high-resolution (0.04°, 4.5 km) are being tested. The model reproduces the general circulation of the global ocean, the strength and variability of the western boundary currents, such as the Gulf Stream and Kuroshio, and the mesoscale eddies generated by instabilities of the major currents. The model is validated by comparison to *in situ* observations not used in assimilation (Hurlburt, et al., 2010; Thoppil, et al., 2011). The global model performance is not as good near the coasts where smaller scale features in the bathymetry and new phenomena, such as tides, become important in determining the state of the ocean.

The oceanic general circulation consists of the well-known current systems around the globe, such as the Gulf Stream in the western North Atlantic Ocean, the Kuroshio Current in the western North Pacific Ocean, the Agulhas Current off of Southern Africa, the Antarctic Circumpolar Current in the Southern Ocean, and so on (Schmitz, 1996a,b; Siedler, et al., 2001). On time-scales of about 10–200 days, these currents meander and produce highly-energetic mesoscale eddies; the spinning oceanic dynamical counterparts of atmospheric weather systems. Because currents meander and mesoscale eddies are born and die over relatively long time-scales (scales much longer than, for instance, tidal time-scales), these currents and eddies are sometimes referred to as low-frequency currents.

A substantial percentage of the kinetic energy in the ocean resides in mesoscale eddies and other transient low-frequency features such as current meanders. Since the horizontal length scales of mesoscale eddies are relatively small (about 100 km), numerical models of the ocean must have high-resolution in order for the modeled mesoscale eddy kinetic energies to compare well to those recorded in observations from satellite altimeters and other instruments (Hurlburt and Hogan, 2000; Maltrud and McClean, 2005). The current state-of-the-art in global high-resolution ocean modeling is about 1/10° to 1/12°, which is thought to be sufficient to capture most of the oceanic mesoscale. Indeed, ocean models having resolutions this high or better are known as "eddying" models (Hecht and Hasumi, 2008). Note, however, that Thoppil, et al. (2011) find that increasing the resolution of HYCOM from 0.08° to 0.04° yields further increases in the energy of modeled mesoscale eddies, suggesting that resolutions as high as 1/25° might be necessary to fully-resolve mesoscale eddies.

3. Combining Ocean Tides and the Ocean General Circulation

Our goal is modeling the ocean tides and the atmospherically-forced eddying ocean general circulation simultaneously. We add astronomical tidal potential forcing of the four largest semidiurnal constituents (M_2, S_2, N_2, and K_2) and the four largest diurnal constituents (K_1, O_1, P_1, and Q_1) to the HYCOM general circulation model, which is forced by atmospheric fields as discussed in the previous section. We parameterize the topographic internal wave-drag using the scheme of Garner (2005), modified as described in Arbic, et al., (2010) to limit the maximum damping rates to less than nine (9) hours. The topographic wave-drag arises from the generation of unresolved small-scale internal waves by tidal flow over rough topography. Because the wave-drag acting on tidal motions is known to have a different strength than the wave-drag acting on non-tidal motions (Bell, 1975), we separate the tidal and non-tidal bottom flows in our model using a 25-hour running boxcar filter. The self-attraction and loading term is simplified using the scalar approximation (Ray, 1998) and is applied only to the non-steric component of the sea surface height, which is dominated by the large-scale barotropic tides.

These modifications allow HYCOM to generate both barotropic and internal tides, amidst the eddying general circulation. The barotropic tides in the new HYCOM tidal simulations are compared against the standard set of 102 pelagic tide gauges (Shum, et al., 1997) with about 93% of the sea surface elevation variance of the eight largest tidal constituents in the tide gauge records captured by HYCOM. The non-tidal sea surface height variability is not strongly affected by the presence of tides in the model. In the new HYCOM tidal simulations, in contrast to early internal tide simulations (Arbic, et al., 2004; Simmons, et al., 2004), realistic atmospheric-forcing allows for a realistic horizontally-varying stratification in the ocean model with a corresponding improvement in the internal tides generated at high-latitudes.

When a horizontally-uniform stratification, typical of the subtropics, is used, an artificially strong internal tide in high-latitude regions such as the Labrador Sea and the Drake Passage, is produced, as seen in Figure 2a. With a realistic horizontally-varying stratification from the atmospheric-forcing, the new HYCOM simulations with tides remove the unrealistic internal tides at high-latitudes as seen in Figure 2b.

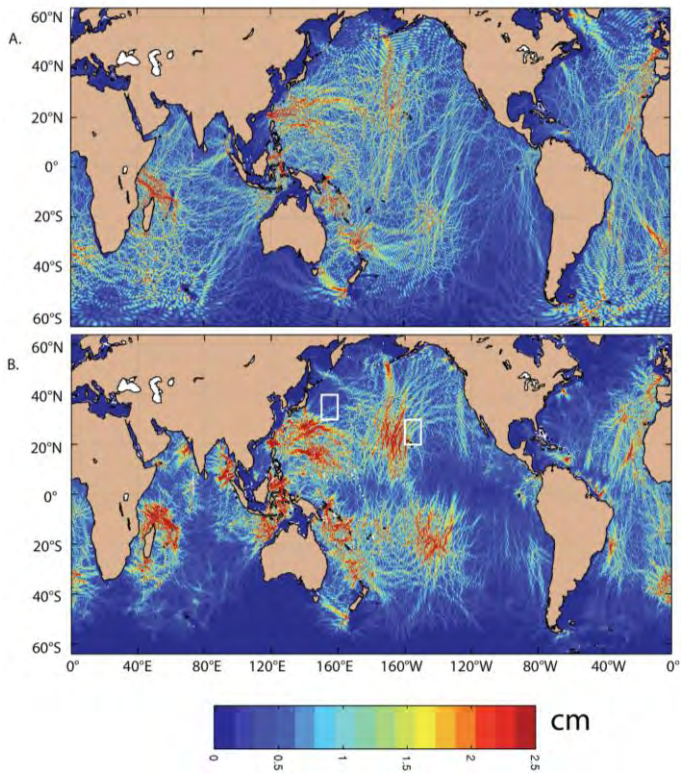


Figure 2. Amplitude (cm) of M_2 internal tide from HYCOM simulations run with horizontally-uniform stratification and tidal potential forcing only (A), and with concurrent atmospheric and tidal potential forcing and realistic horizontally-varying stratification (B). Note the absence of high-latitude internal tides where the stratification is weak and the increase of the amplitude in the mid-latitude generation regions.

4. Comparing the Ocean Internal and Surface Tides to Observations

To assess the accuracy of surface barotropic tide in the concurrent HYCOM simulation, the HYCOM barotropic tide is compared to the tides generated by a data-assimilative shallow water barotropic tide model (TYPO7.2 adapted from Egbert, et al., 1994) and the 102 pelagic tide gauges described by Shum, et al. (1997). When the HYCOM tides are compared to TPXO, the rms error is 7.48 cm for M_2 and 9.52 cm for all 8 constituents. When compared to the 102 pelagic gauges, the rms error increases slightly to 7.80 cm for M_2 and 1,022 cm for all 8 constituents. Data assimilation reduces the errors for the shallow water barotropic tide models relative to the 102 pelagic gauges to ~1.6 cm for M2 and ~3 cm for the 8 constituents (Shum, et al., 1997). The HYCOM tides are comparable to other non-assimilative shallow water barotropic tide models with rms errors for M_2 of ~7 cm for Jayne and St. Laurent (2001) and Arbic, et al. (2004) and ~5 cm for Egbert, et al. (2004). The latter models used a rigorous self-attraction and loading (SAL) correction compared to the scalar SAL correction used in HYCOM. Bathymetry differences between the models can lead to significant differences in the tidal amplitudes. Egbert, et al. (2004) found that 5–10% random differences in the bathymetry could cause ~8 cm differences in the M_2 tidal amplitudes.

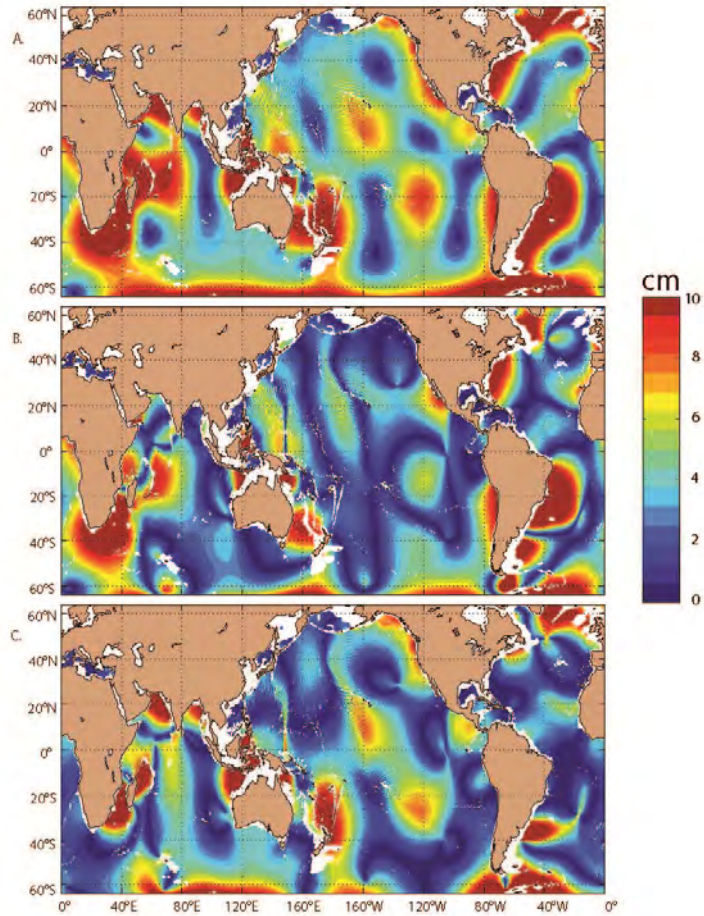


Figure 3. The total (a), amplitude (b) and phase (c) contributions to the rms error (cm) of the HYCOM M_2 surface tidal elevation compared to the TPXO M_2 tidal elevation

Errors in the tidal elevation for the model can arise from errors in the absolute height (amplitude) or timing (phase). We can express the mean square error (MSE) in terms of the tidal amplitude and phase,

$$\mathrm{MSE} = \tfrac{1}{2}\,[(A_{\mathbf{HYCOM}})^2 + (A_{\mathbf{TPXO}})^2] - A_{\mathbf{HYCOM}}\, A_{\mathbf{TPXO}} \cos(\Phi_{\mathbf{HYCOM}} - \Phi_{\mathbf{TPXO}}), \tag{1}$$

where A and Φ are the tidal amplitude and phase, respectively, and the subscripts HYCOM and TPXO refer to the models. The contribution to the errors can be separated into a part arising from amplitude errors only ($\mathrm{MSE}_{amplitude}$) which vanishes if the amplitudes of the models are equal, and a part arising from phase errors (MSE_{phase}) which vanishes if the phases of the models are equal.

$$\mathrm{MSE}_{amplitude} = \tfrac{1}{2}\,[(A_{\mathbf{HYCOM}})^2 - (A_{\mathbf{TPXO}})^2], \tag{2}$$

$$\mathrm{MSE}_{phase} = A_{\mathbf{HYCOM}}\, A_{\mathbf{TPXO}}\,[1 - \cos(\Phi_{\mathbf{HYCOM}} - \Phi_{\mathbf{TPXO}})], \tag{3}$$

Maps of the total, amplitude and phase rms errors for the M2 tide are shown in Figure 3. Globally, the total rms error of 7.48 cm is almost evenly split between amplitude errors of 5.56 cm and phase errors of 5.00 cm. However, the spatial distribution of the amplitude and phase errors is quite different.

Satellite altimetry represents the only source of global tidal elevation data. However, the tidal frequencies are aliased into lower frequencies by the satellite sampling (Parke, et al., 1987). Recognizing that internal tides have much shorter spatial scales than the barotropic tide, we apply a high-pass filter to the M_2 amplitudes at the sea surface in order to remove the large-scale (barotropic) tide, so that only the small-horizontal scale (internal) tide remains. The results of the application of the high-pass filter to the surface M_2 elevations are displayed in Figure 4. Figure 4a displays results from along-track satellite altimeter data (Ray and Mitchum, 1996; 1997), while Figure 4b displays results from the model output interpolated to the same altimeter tracks. The "hotspots" of internal tide generation at prominent topographic features such as Hawai'i and the Aleutians in the North Pacific, the Tuamotu Archipelago in the South Pacific, Madagascar in the Indian Ocean, and so on, are very clearly seen in both panels of Figure 4. The internal waves radiate away from the hot-spots as

focused beams, which propagate for 1000s of kilometers. The M_2 internal tide amplitudes are typically about 1 cm or more in the hot-spot regions. Averaging the amplitudes over the five hot-spot regions, we find that the model agrees with the altimeter data to within about 20% over these regions. This result lends further confidence in the realism of the internal tides in our HYCOM simulations, in similar fashion to the side-by-side plots of the low-frequency sea surface height variance in high-resolution models versus satellite altimeter data (see for instance Figure 12 in Maltrud and McClean, 2005), which lend confidence in the ability of high-resolution models to simulate mesoscale eddies.

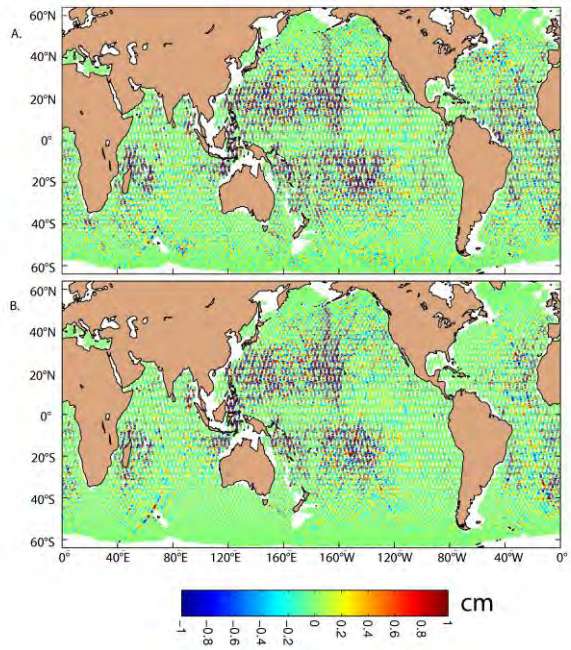


Figure 4. The M2 internal tide amplitude from the altimetric-based (a) and HYCOM (b) tidal analyses

5. Moving the Model Towards a Forecast System

Some of the largest errors in the HYCOM concurrent tides compared to the data-assimilative shallow water barotropic tide occur in the Atlantic Ocean and Southern Ocean as shown in Figure 5, where the rms M_2 tidal elevation difference (Figure 5a) and the model skill (Figure 5c) of HYCOM tides compared to TYPO7.2. In the Southern Ocean, significant differences between the geometry and bathymetry of the two models exist. The data-assimilative model approximates the ocean under the floating ice shelves by the thickness of the ocean beneath the shelves, while HYCOM places a vertical boundary at the edge of the floating ice shelves. The tidal amplitudes and transport at the floating ice shelf edges are not negligible in the data-assimilative model. We exploit a capability in HYCOM to open ports at the edges of the floating ice shelves, where the tidal amplitudes and transport from the data-assimilative model can be used as a boundary condition. The resulting simulation has significantly reduced rms errors (Figure 5b) and increased model skill (Figure 5d) over much of the ocean not just the Southern Ocean where the boundary conditions are applied.

The new concurrent tide model is also expected to be useful for regional and coastal modeling. Regional and coastal models often include tidal forcing, but internal tides are not included in their open boundary conditions. Kelly and Nash (2010) showed that internal tides propagating onto the continental shelf can completely change the tides over the shelf compared to a barotropic tide-only forced model. The Australian North West Shelf is a region where a low vertical mode internal tide is expected to propagate onto the shelf from the open ocean (Kelly, personal communication). Current meter observations show propagation of a mode-1 internal tide onto the shelf. The global model at 4 km resolution underestimates the strength of the internal tide compared to the observations, but shows on-shore propagation from deep water onto the shelf consistent with the observations. However, a regional model forced by barotropic tidal boundary conditions shows onshore propagation of the internal tide only in-shore of the critical slope, and off-shore propagation over the outer continental slope unlike the observations. By including tidal-forcing in a fully three-dimensional (3D) global ocean model, we will provide an internal tide capability everywhere, and allow nested models to include internal tides at their open boundaries.

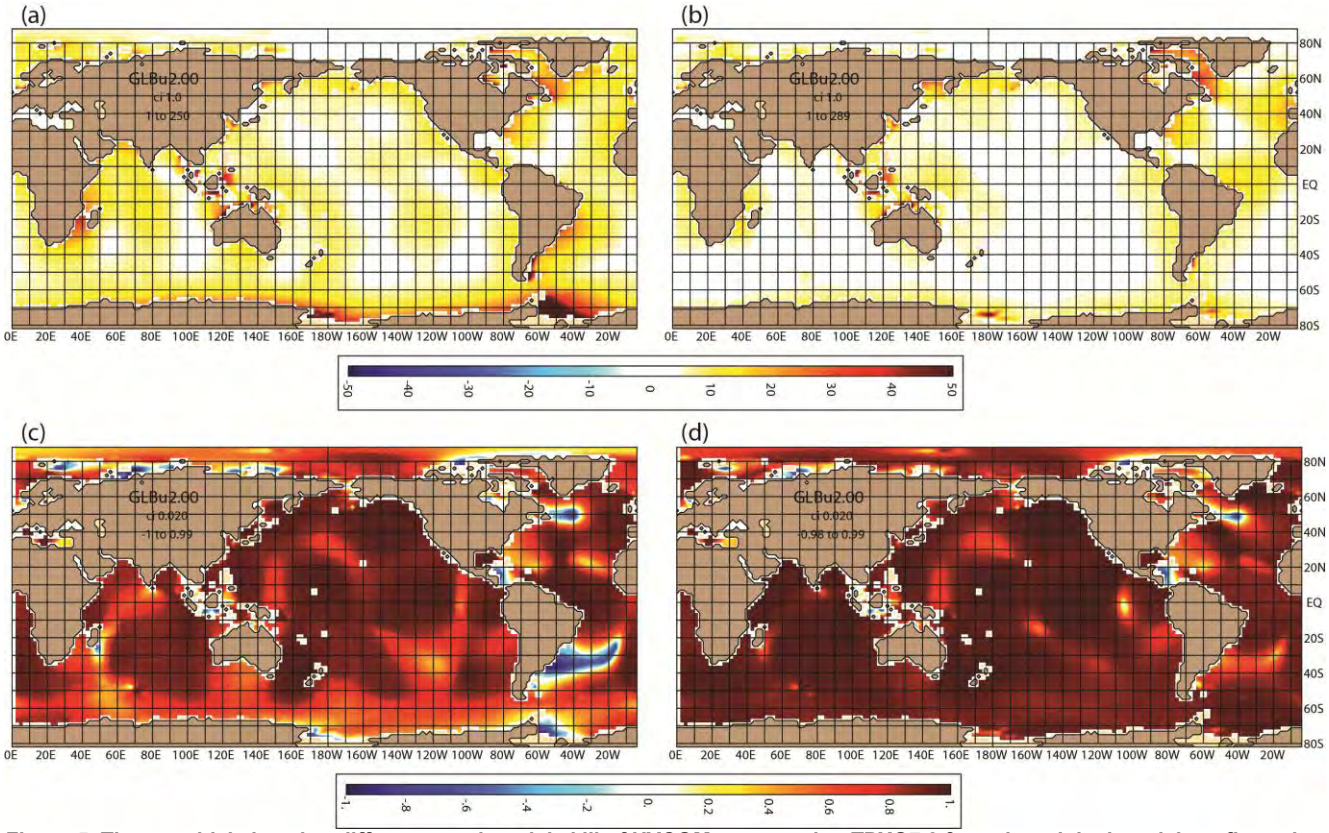


Figure 5. The rms tidal elevation difference and model skill of HYCOM compared to TPXO7.2 from the original model configuration with solid walls at the floating ice shelves (a, c) and with ports mimicking the TPXO tidal elevation and transports at the floating ice shelves (b, d)

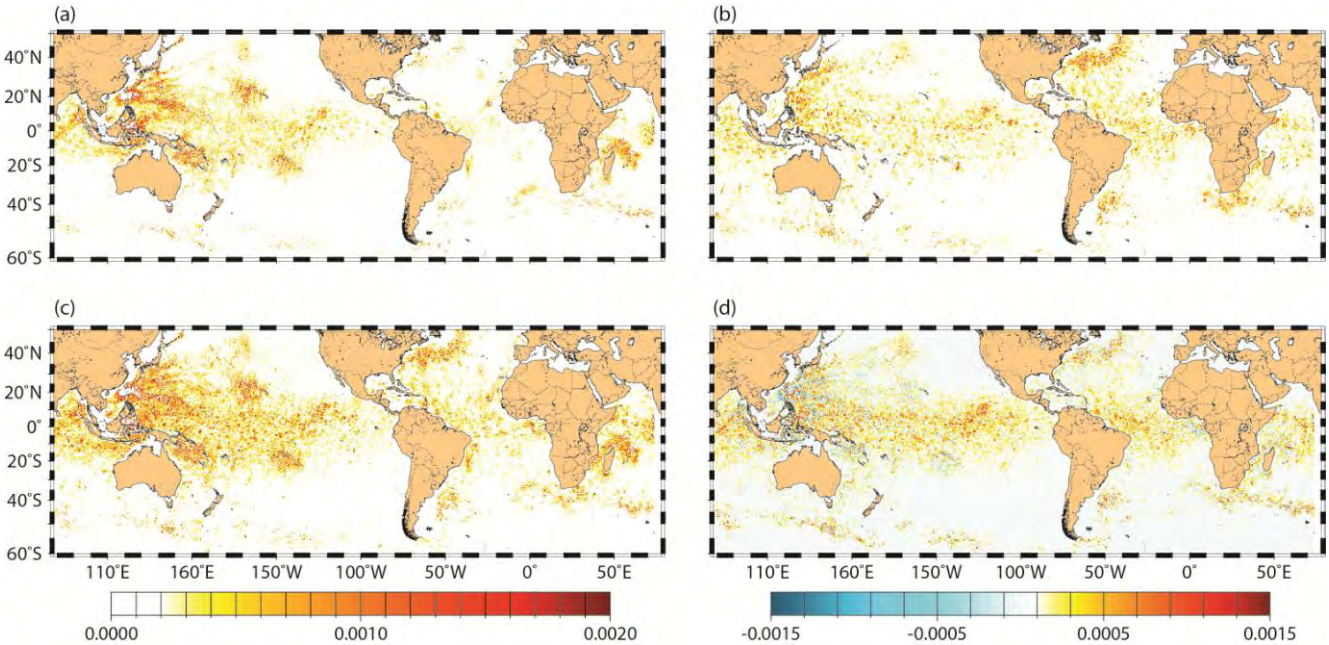


Figure 6. The steric sea surface variance for three model simulations, HYCOM with tides (a), HYCOM with data assimilation of mesoscale eddies using NCODA and HYCOM with tides and data assimilation (c). The lower-left panel (d) shows the difference between HYCOM with tides and assimilation and the sum of the variances of HYCOM with tides only and HYCOM with assimilation only (c – (a+b)).

A number of improvements are needed for the new concurrent tide model. The self-attraction and loading term (Hendershott, 1972) is implemented in a rather crude way, using the so-called scalar approximation (Ray, 1998), and this needs to be improved upon in future versions of the model. The frequency dependence of topographic wave-drag (Bell, 1975) is problematic since the model is run in the time domain, not the frequency domain. Perhaps the greatest technical problem remaining is how to perform data assimilation on the tidal, as well as non-tidal motions within the model. Experiments have begun to use the existing data assimilation scheme (NCODA, Cummings, 2005) with the new concurrent circulation and tide model. The results are encouraging, as seen in Figure 6. In the upper-left panel (Figure 6a), steric sea surface height (SSH) variance for a single day is shown for the model with tides. The hot-spots of internal wave generations are clearly visible. In the upper-right panel (Figure 6b), the steric SSH variance is shown for the model without tides, but assimilating data using the mesoscale covariances in NCODA applied to a daily averaged model background state. In this figure, high-frequency variability is generated by baroclinic adjustment of the mesoscale eddies and meanders of strong ocean currents. The strength of the adjustment internal waves is similar to the strength of the internal tides, but the adjustment waves are strongest near the western boundary currents instead of the topographic hot spot generation sites. In the lower left panel (Figure 6c), the steric SSH variance for the model with tides and assimilation of mesoscale eddies via NCODA is shown. The SSH variance is larger than either the tide-only model or assimilation-only, with large variance at both the topographic hot-spots and western boundary currents. In the lower-right panel (Figure 6d), the difference in the SSH variance for the model with tides and assimilation and the sum of the variances from the tide-only and data assimilation-only models. The difference variance is lower than the variances of the individual models. The hot-spots of tidal generation are not found in the difference plot, suggesting that data assimilation of the mesoscale eddies is not affecting the tidal solution. Successful data assimilation, once performed, will make the model much more useful for operational purposes of the United States Navy and other users.

Acknowledgements

This work was supported in part by a grant of computer time from the DoD High Performance Computing Modernization Program at the NAVY DSRC. It was sponsored by the US Office of Naval Research (ONR) through the following projects: 6.2 NRL: Eddy Resolving Global Ocean Prediction Including Tides and 6.1 NRL: Ageostrophic Vorticity Dynamics.

References

Arbic, B.K., S.T. Garner, R.W. Hallberg, and H.L. Simmons, "The accuracy of surface elevations in forward global barotropic and baroclinic tide models." *Deep-Sea Research II*, 51, pp. 3069–3101, 2004.

Arbic, B.K., A.J. Wallcraft, and E.J. Metzger, "Concurrent simulation of the eddying general circulation and tides in a global ocean model." *Ocean Modelling*, 32, pp. 175–187, 2010; doi:10.1016/j.ocemod.2010.01.007.

Bell, T.H., "Lee waves in stratified flows with simple harmonic time dependence." *Journal of Fluid Dynamics*, 67, pp. 705–722, 1975.

Chassignet, E.P., H.E. Hurlburt, O.M. Smedstad, G.R. Halliwell, P.J. Hogan, A.J. Wallcraft, R. Baraille, and R. Bleck, "The HYCOM (HYbrid Coordinate Ocean Model) data assimilative system." *Journal of Marine Systems*, 65, pp. 60–83, 2007.

Cummings, J.A., "Operational multivariate ocean data assimilation." *Quarterly Journal of the Royal Meteorological Society*, 131, pp. 3583–3604, 2005.

Egbert, G.D., A.F. Bennett, and M.G.G. Foreman, "TOPEX/POSEIDON tides estimated using a global inverse model." *Journal of Geophysical Research,* 99, pp. 24821–24852, 1994.

Egbert, G.D. and R.D. Ray, "Significant dissipation of tidal energy in the deep ocean inferred from satellite altimeter data." *Nature*, 405, pp. 775–778, 2000.

Garner, S.T., "A topographic drag closure built on an analytical base flux." *Journal of the Atmospheric Sciences*, 62, pp. 2302–2315, 2005.

Hendershott, M.C., "The effects of solid earth deformation on global ocean tides." *Geophysical Journal of the Royal Astronomical Society*, 29, pp. 389–402, 1972.

Hibiya, T., M. Nagasawa, and Y. Niwa, "Global mapping of diapycnal diffusivity in the deep ocean based on the results of expendable current profiler (XCP) surveys." *Geophysical Research Letters*, 33, L03611, 2006; doi:10.1029/2005GL025218.

Holloway, P.E., P.G. Chatwin, and P. Craig, "Internal tide observations from the Australian North West Shelf in summer 1995." *Journal of Physical Oceanography*, 31, pp. 1182–1199, 2001.

Hurlburt, H.E. and P.J. Hogan, "Impact of 1/8° to 1/64° resolution on Gulf Stream model-data comparisons in basin-scale subtropical Atlantic Ocean models." *Dynamics of Atmospheres and Oceans*, 32, pp. 283–329, 2000.

Hurlburt, H.E., E.J. Metzger, J.G. Richman, E.P. Chassignet, Y. Drillet, M.W. Hecht, O. Le Galloudec, J.F. Shriver, X. Xu, and L. Zamudio, Dynamical evaluation of ocean models using the Gulf Stream as an example, *Operational Oceanography in the 21st Century*, Brassington, G.B. and Schiller, A. (eds.), Springer-Verlag, New York, pp. 545–609, 2010.

Jayne, S.R. and L.C. St. Laurent, "Parameterizing tidal dissipation over rough topography." *Geophys. Res. Lett.*, 28, pp. 811–814, 2001.

Kelly, S.M. and J.D. Nash, "Internal-tide generation and destruction by shoaling internal tides." *Geophysical Research Letters*, 37, L23611, 2010; doi:10.1029/2010GL045598.

Maltrud, M.E. and J.L. McClean, "An eddy resolving global 1/10° simulation." *Ocean Modelling*, 8, pp. 31–54, 2005.

Metzger, E.J., O.M. Smedstad, P.G. Thoppil, H.E. Hurlburt, D.S. Franklin, G. Peggion, J.F. Shriver, and A.J. Wallcraft, "Validation Test Report for the Global Ocean Forecast System V3.0 – 1/12° HYCOM/NCODA: Phase II." *NRL Memo, Report NRL/MR/7320-10-9236*, 2010 (available at http://www7320.nrlssc.navy.mil/pubs.php).

Munk, W. and C. Wunsch, "Abyssal recipes II: energetics of tidal and wind mixing." *Deep-Sea Research I*, 45, pp. 1977–2010, 1998.

Padman, L., S. Howard, and R. Muench, "Internal tide generation along the South Scotia Ridge." *Deep-Sea Research II*, 53, pp. 157–171, 2006; doi:10.1016/j.dsr2.2005.07.011.

Parke, M.E., R.H. Stewart, D.L. Farless, and D.E. Cartwright, "On the choice of orbits for an altimetric satellite to study ocean circulation and tides." *Journal of Geophysical Research*, 92, pp. 11693–11707, 1987.

Ray, R.D., "Ocean self-attraction and loading in numerical tidal models." *Marine Geodesy*, 21, pp. 181–191, 1998.

Ray, R.D., "A global ocean tide model from TOPEX/POSEIDON altimetry: GOT99.2." *NASA/TM-1999-209478*, National Aeronautics and Space Administration Technical Memorandum, 58 pp, 1999.

Ray, R.D. and G.T. Mitchum, "Surface manifestation of internal tides generated near Hawai'i." *Geophysical Research Letters*, 23, pp. 2101–2104, 1996.

Ray, R.D. and G.T. Mitchum, "Surface manifestation of internal tides in the deep ocean: Observations from altimetry and tide gauges." *Progress in Oceanography*, 40, pp. 135–162, 1997.

Rosmond, T.E., J. Teixeira, M. Peng, T.F. Hogan, and R. Pauley, "Navy Operational Global Atmospheric Prediction System (NOGAPS): Forcing for Ocean Models." *Oceanography*, *15*, pp. 99–108, 2002.

Shum, C.K., P.L. Woodworth, O.B. Andersen, G.D. Egbert, O. Francis, C. King, S.M. Klosko, C. Le Provost, X. Li, J.-M. Molines, M.E. Parke, R.D. Ray, M.G. Schlax, D. Stammer, C.C. Tierney, P. Vincent, and C.I. Wunsch, "Accuracy assessment of recent ocean tide models." *Journal of Geophysical Research*, 102, pp. 25173–25194, 1997.

Shriver, J.F., B.K. Arbic, J.G. Richman, R.D. Ray, E.J. Metzger, A.J. Wallcraft, and P.G. Timko, "An evaluation of the barotropic and internal tides in a high resolution global ocean circulation model." *J. Geophysical Research*, 117, 2012; doi:10.1029/2012JC008170.

Simmons, H.L., R.W. Hallberg, and B.K. Arbic, "Internal wave generation in a global baroclinic tide model." *Deep-Sea Research II*, 51, pp. 3043–3068, 2004.

Simmons, H.L., "Spectral modification and geographic redistribution of the semi-diurnal internal tide." *Ocean Modelling*, 21, pp. 126–138, 2008; doi:10.1016/j.ocemod.2008.01.002.

Stammer, D., "Global characteristics of ocean variability estimated from regional TOPEX/Poseidon altimeter measurements." *Journal of Physical Oceanography*, 27, pp. 1743–1769, 1997.

Thoppil, P.G., J.G. Richman, and P.J. Hogan, "Energetics of a global ocean circulation model compared to observations." *Geophysical Research Letters*, 38, L15607, 2011; doi:10.1029/2011GL048347.

Wunsch, C., "Internal tides in the ocean." *Reviews of Geophysics and Space Physics*, 13, pp. 167–182, 1975.

Prediction of Rogue Waves by Large-Scale Phase-Resolved Nonlinear Wave-Field Simulations

Wenting Xiao, Yuming Liu, and Dick K.P. Yue
Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA
{wtxiao, yuming, yue}@mit.edu

Abstract

*The occurrence of rogue waves has been observed by ship crews and detected in wave measurements. It has been considered as an important cause for catastrophic losses of ships and marine structures at sea. A reliable prediction of the occurrence of rogue waves under various sea conditions is of critical significance to the design of new advanced navy ships, and to increase survivability and operational envelopes of existing ships. Field measurements provide only limited information on the rogue wave statistics because of the rare occurrence of rogue waves. Numerical simulation thus becomes a powerful tool to investigate the generation mechanism and occurrence statistics of rogue waves. In this work, we apply a direct simulation tool, called SNOW, to obtain large-scale phase-resolved nonlinear wave-fields from which we extract rogue wave events, study occurrence statistics, and investigate the rogue wave kinematics and dynamics. Unlike the model equation approach that is based on nonlinear Schrödinger equation, the present simulation includes high-order interactions of broadband wave components with broad spreading directions. With routine large-scale SNOW (**s**imulation of **n**onlinear **o**cean **w**ave-fields) simulations, we obtain a collection of synthetic three-dimensional nonlinear phase-resolved wave-fields associated with a wide-range of wave spectrum parameters (such as effective wave steepness, spectrum bandwidth, and spreading direction angle). Based on these wave-fields, we calculate the detailed nonlinear statistics of ocean waves and obtain dependences of rogue wave properties on wave spectrum parameters. We confirm that the probability of rogue wave occurrence can be significantly underestimated by the classical linear and second-order theories, depending on the wave spectrum parameters. We find that the rogue wave statistics are strongly correlated with the kurtosis of the wave-field. Moreover, we characterize the kinematics of rogue waves under various sea conditions. In addition, the validity and limitations of existing model equation approaches are assessed by direct comparison with the present simulations.*

1. Introduction

Rogue waves are water surface waves whose wave heights are unexpectedly large for a given sea state. A common criterion used to define a *rogue* wave is when its crest-to-trough height is at least twice the significant wave height of the wave-field. The occurrence of rogue waves has been observed by ship crews and identified from point and wide-area wave measurements (Lawton, 2001; Kharif, 2003; Forristall, 2005). It has been considered as an important cause for catastrophic losses of ships and marine structures at sea.

A reliable prediction of the occurrence probability of rogue waves under various sea conditions is of critical significance to the design of advanced ships, and to increase survivability and operational envelopes of existing ships. In the linear Gaussian random wave theory, the wave-field is considered as a superposition of a large number of sinusoidal waves with different frequencies and directions of propagation. Under the assumption that the spectrum is narrow-banded, Longuet-Higgins (1952) has shown that the wave height follows the Rayleigh distribution. The probability of rogue wave occurrence is, therefore, described by the tail of the distribution. The linear theory can be easily extended to include the second-order bound wave effects (Tayfun, 1980; Forristall, 2000). The inclusion of the second-order nonlinearity leads to sharper crests and flatter troughs, but the wave height remains the same. The second-order theory is generally used to estimate the crest (or trough) height distribution, with good agreements with field measurements for mild seas. Recent wave records show the presence of extremely large waves with wave heights larger than twice the significant wave height of the

wave-field. The occurrence of these large waves cannot be properly described by the classical linear and second-order theories (Skourup, et al., 1997; Stansell, 2005). Several possible physical mechanisms have been proposed to explain the enhanced occurrence probability of large waves. In the context of linear mechanism, large wave events are a result of superposition of a large number of waves with appropriate frequency and phase combinations. In this mechanism, first, rogue waves could be formed from a dispersive focusing of a linear wave group. Second, rogue waves could be resulted from wave focusing associated with wave refraction by variable current or bottom topography (Lavrenov, 1998). Third, wind forcing may also lead to the development of rogue waves (Kharif, et al., 2007). The interested reader is referred to Kharif & Pelinovsky (2003) for a complete review in this context.

Nonlinear focusing is also an important generation mechanism of rogue waves, which is fundamentally different from the linear mechanisms. A periodic Stokes wave-train is unstable to side-band modulations, well-known as Benjamin-Feir (BF) instability (Benjamin & Feir, 1967). The instability causes the wave-train to disintegrate into wave groups within which large waves could form. Alber (1978) has shown the BF-type instability could exist in random wave-fields with narrow-band wave spectra. The relation between BF-type modulational instability and rogue wave formation has been studied theoretically, experimentally and numerically. Janssen (2003) has shown theoretically that in severe seas with narrow-band wave spectra, the effect of modulational instability can cause an exponential growth of the central wave mode. He pointed out that the probability of rogue wave occurrence depends on the ratio between the wave-field steepness and spectral bandwidth, named as Benjamin-Feir index (BFI). This agrees with wave flume experiments (Onorato, et al., 2004).

In addition to theoretical analyses, numerical simulations are also used to investigate nonlinear generation mechanisms and understand statistical characteristics of rogue waves. In the numerical study of modulational instability effect on rogue wave statistics, simulations of nonlinear wave-field evolution in a large domain over a long period of time are needed in order to obtain statistically reliable information. This requires the use of very large numbers of wave modes and time-steps in direct phase-resolved simulations that were limited by available computing resources in the past. Most of the studies are thus based on the wave envelope models developed with nonlinear Schrödinger equation (NLS) and modified nonlinear Schrödinger equation (MNLS) (Dysthe, 1979; Trulsen & Dysthe, 1996). Typically, these models assume that the wave spectrum is narrow-banded and the modulation of the wave profile varies slowly with time and space. Although a considerable amount of progresses has been made in the past, the reliability and accuracy of these models has not been carefully justified. In particular, the application of NLS-type model to realistic ocean wave-fields is questionable because the inherent assumption of narrow bandwidth wave spectrum is often invalid.

With recent development of computational capabilities and fast numerical algorithms for phase-resolved simulations of nonlinear wave dynamics, the time and opportunity have arrived for the application of direct phase-resolved simulations for nonlinear ocean wave predictions. Based on an efficient high-order spectral method (HOS), we have established a powerful computational capability for phase-resolved prediction of large-scale nonlinear ocean wave-field evolutions (e.g., Tsai & Yue, 1996). We call this direct-simulation-based wave prediction capability as SNOW (**s**imulation of **n**onlinear **o**cean **w**ave-fields). Unlike the phase-averaged model, SNOW obtains the detailed phase information of the wave-field during its nonlinear evolution, which enables the modeling of complex physics such as wave-breaking and wind forcing in a more physics-based manner. In this study, large-scale direct phase-resolved SNOW simulations are used to study the effect of modulational instability on nonlinear ocean wave statistics, to assess the accuracy and reliability of the existing wave prediction models, and, more importantly, to study the dependence of occurrence probability of rogue waves on sea states.

2. Numerical Methodology

The problem of nonlinear ocean surface wave-field evolution is considered in the context of potential flow formulation. SNOW solves the primitive field equations with nonlinear kinematic and dynamic free-surface boundary conditions in the Zakharov form (Zakharov, 1968). SNOW solves these equations based on the use of a high-order pseudo-spectral method (Dommermuth & Yue, 1987) that is capable of following nonlinear evolution of a large number of wave modes in the wave-field with a minimum computation requirement. The method includes nonlinear interactions of all wave components up to an arbitrary order (M) in wave steepness. The nonlinear wave interactions with variable currents and bottom bathymetry are also considered in a direct way while the wave-breaking dissipation and wind input are accounted for using physics-based modeling.

With SNOW, the computational effort is almost linearly proportional to M and the large number of wave modes (N). Exponential convergence of the solution with M and N is also obtained. Such high-efficiency and accuracy makes SNOW an ideal approach for the computation of large-scale nonlinear phase-resolved wave-field evolution.

At the initial stage of SNOW simulations of nonlinear wave-field evolution, the free-surface boundary conditions are smoothly transitioned in time from linear to nonlinear conditions for minimizing any standing wave effect that results from

using linear initial conditions (Dommermuth, 2000). In a typical SNOW simulation, this is done over five periods of the dominant wave mode.

Parallel implementation of SNOW is achieved by decomposing the spectral domain across processors using Message Passing Interface (MPI). SNOW utilizes transposes and processor fast-Fourier transforms to move between the physical and spectral domains. SNOW achieves almost linear scalability for a large number of processors (*O* (2000)), making it highly-scalable. Figure 1 shows the typical performance of SNOW as a function of processor number and problem size on the Cray XT3, Cray XT4, Cray XT5, and Cray XE6 at the US Army Engineer Research and Development Center (ERDC). The performance is measured as the number of time-steps of simulations achieved in five wall-clock minutes. In general, the near-linear scalability is achieved on all these systems.

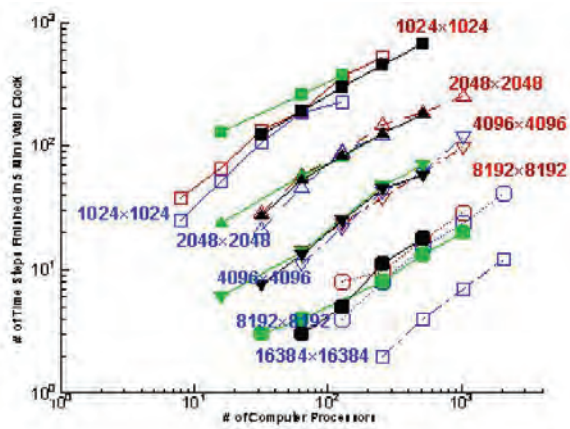


Figure1. Scalability of typical SNOW simulations on Cray XT3 (blue), Cray XT4 (red), Cray XT5 (green) and Cray XE6 (black). Plotted is the number of time-steps of computations completed within 5 minutes of wall-clock time versus the number of processors used in the simulation for different problem scales ($N_x \times N_y$).

3. Results and Discussion

3.1 SNOW Simulated Wave-field Database

Our main purpose is to study the dependence of the occurrence probability of rogue waves on various sea states. To do this, we have simulated a large number of nonlinear directional ocean wave-fields described by different values of the spectral parameters in the initial (JONSWAP) wave spectra. The effective steepness, $\varepsilon=H_s k_p/2$, ranges from 0.06 to 0.16 with an increment of $\Delta\varepsilon=0.2$. The spectral bandwidth in the dominant direction is controlled by the enhancement parameter γ that takes values of 1.0, 3.3, 5, 6, and 10. The crest length is controlled by the spreading angle Θ of the wave-field. Θ varies from 20° to 120° with an increment of $\Delta\Theta=10°$ for $\Theta\leq 40°$ and $\Delta\Theta=20°$ for $\Theta>40°$. In total, we have considered 6×5×7=210 wave spectra specified with different combinations of ε, γ, and Θ values. For these wave-fields, the BFI ranges from 0.2 to 1.2. Our study on rogue wave statistics will be based on the collection of rogue wave events detected in this wave-field dataset and for convenience we refer this dataset as MITWAVE. To include the nonlinear wave-wave interactions at least to third-order (for including quartet resonant wave interactions), the typical domain size of the simulated wave-field is L_x, $L_y=O(\lambda_p\varepsilon^{-2})$ and the typical evolution time $t=O(T_p\,\varepsilon^{-2})$. We use the order of nonlinearity M=4 in SNOW simulations. Different realizations have been obtained for a few cases, which show similar results. Thus, for most of the cases in MITWAVE, only one realization is achieved for one spectrum. Each SNOW simulation in this study takes ~10 CPU-hours on Cray XT4 (Jade) using 512 processors.

3.2 Occurrence Probability of Rogue Waves

The probability of rogue wave occurrence at a given time instance, P_N, is defined as the ratio between the number of rogue waves, N_R, to the total number of waves in the wave-field N at that time, $P_N=N_R/N$.

Figure 2 shows the variation of rogue wave probability (scaled by the Rayleigh solution) during nonlinear evolution of the wave-fields with different spectral parameters. All the wave-fields have the same wave steepness $\varepsilon=H_s k_p/2=0.16$, but different bandwidths and spreading angels. For comparisons, the results obtained by both SNOW and MNLS are shown.

In general, the probability of rogue waves, P_N, decreases with increasing spreading angle and spectral bandwidth. The rogue wave probability is greatly enhanced in the wave-fields with narrow-band spectrum and small spreading-angle (e.g., γ=6, Θ=15° as shown in Figure 2a). The probability of rogue waves can be underestimated by an order if Rayleigh theory is used. In initial evolution time of the wave-fields, $t<\sim 50T_p$, the rogue wave statistics from SNOW and MNLS simulations increase in a similar manner, although MNLS gives a larger occurrence probability. As evolution continues, $t>\sim 50T_p$, SNOW and MNLS simulations give very different descriptions on the rogue wave statistics. In MNLS results, the rogue wave probability decreases to the level of Rayleigh solution. This is due to the effect of over-spreading of energy into broad direction in MNLS simulations. The rogue wave probability from SNOW simulations approaches to a quasi-stationary value.

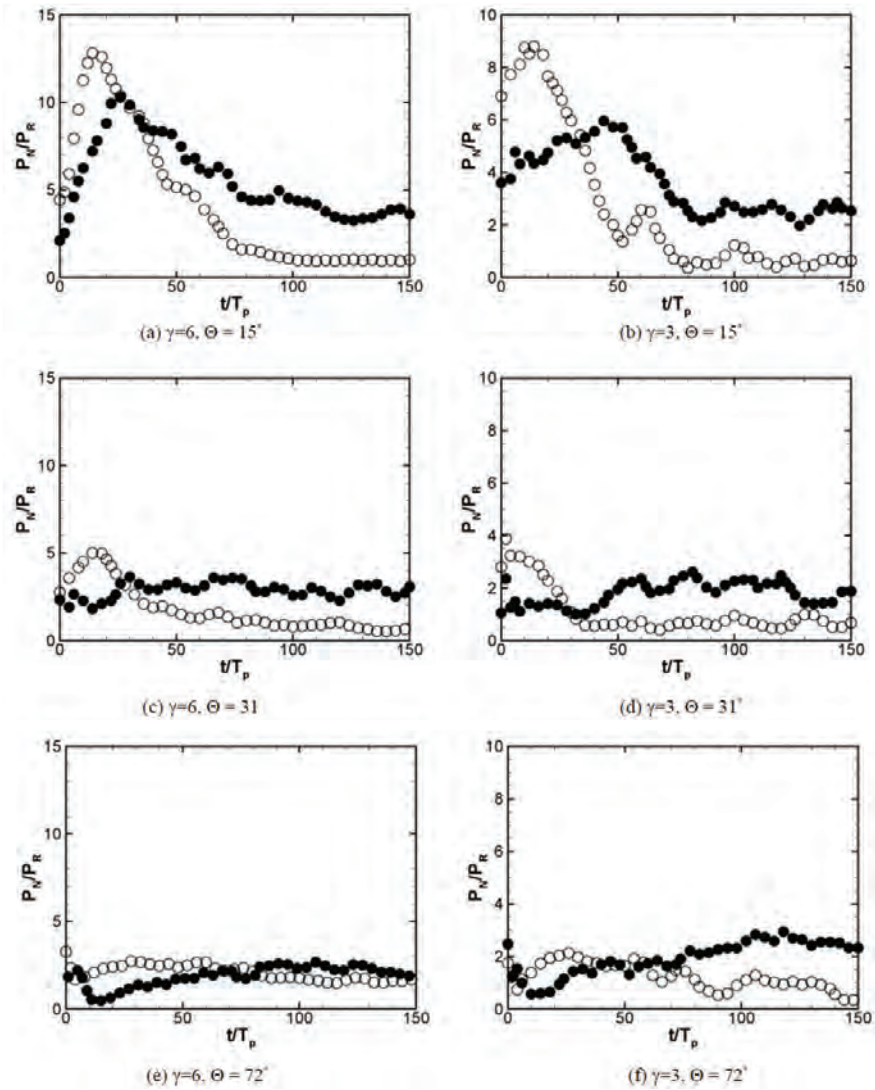


Figure 2. The variation of rogue wave probability P_N scaled by Rayleigh statistics, P_R, over time. The wave-fields are specified by same steepness $\varepsilon=H_s k_p/2=0.16$. The plotted results are from SNOW (●) and MNLS (○). Results are averaged over five (5) realizations.

3.3 Prediction of Rogue Waves from Spectral Parameters

The enhanced occurrence of rogue waves is resulted from nonlinear wave effects, especially that of the modulational instability. Once the effect of third-order nonlinearity is included, the distribution of wave heights in a uni-directional, narrow-banded wave-field is known to follow the modified Edgeworth-Rayleigh (MER) distribution (Mori & Janssen, 2006)

$$P_H\left(H/\sigma\right)=e^{-1/8\left(H/\sigma\right)^2}\left[1+\left(Kur-3\right)B_H\left(H/\sigma\right)\right] \tag{1}$$

where σ is the standard deviation of the surface elevation and *Kur* is the kurtosis of the wave elevation, and $B_H(H/\sigma)=(1/384)(H/\sigma)^2[(H/\sigma)^2-16]$. Figure 3 shows the dependence of rogue wave probability, P_N/P_R, on the maximum kurtosis, based on the rogue wave data collected from ~ 200 wave-fields in MITWAVE. The estimation based on Equation 1 using maximum kurtosis is also plotted as a reference. A strong positive correlation between the rogue wave probability and maximum kurtosis is observed. The MER theory is seen to underestimate the occurrence probability of rogue waves in comparison with SNOW simulations.

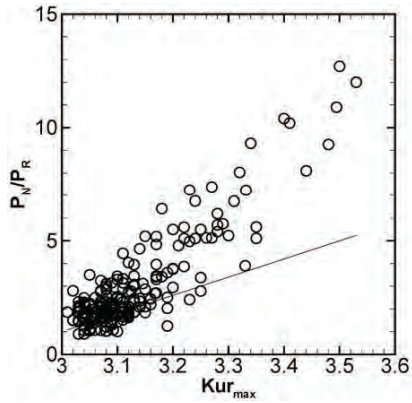


Figure 3. Rogue wave statistics P_N/P_R as a function of maximum kurtosis based on ~200 SNOW simulated directional wave-fields. The plotted results are from SNOW simulations (○) and MER distribution (—).

Despite the fact that there is a close correlation between the rogue wave probability and the maximum kurtosis of the wave-field, the kurtosis itself is not an ideal parameter used for rogue wave prediction because it cannot be calculated as *a priori*. A parameterization of kurtosis, therefore, is essential for the prediction of rogue wave statistics. For uni-directional wave-fields with Gaussian-like, narrow-banded spectra, it is shown that the kurtosis is a function of the BFI (Janssen, 2003): $Kur = 3+\pi\sqrt{3}BFI^2$. Here $BFI=\varepsilon/(\Delta k/k_0)$ and $\Delta k/k_0$ is a measurement of the spectral bandwidth. To assess the validity of parameterization of kurtosis using BFI, Figure 4 plots the relation between the maximum kurtosis, denoted as Kur_{max}, vs. the *BFI* for wave-fields with same steepness ε=0.14 but different (initial) spreading-angles. We observe that, for wave-fields with small spreading-angles, the value of maximum kurtosis strongly depends on the BFI; however, the dependence decreases rapidly as the initial spreading-angle increases. This is because the modulational instability that is responsible for enhanced occurrence probability of rogue waves diminishes in short-crested wave-fields. The dependence of maximum kurtosis on the spreading-angle for various values of BFI can be seen in Figure 5. The kurtosis decreases rapidly with spreading angle for given BFI. This suggests that the value of kurtosis cannot be parameterized by the BFI in short-crested wave-fields; therefore, BFI by itself is not efficient in predicting the occurrence probability of rogue waves in general directional seas. This is further confirmed in Figure 6, in which the rogue wave probability is plotted against the BFI for various spreading-angles. The rogue wave statistics increase with the BFI for wave-fields with small spreading-angles; while for wave-fields with relatively large spreading-angles, the rogue wave statistics is nearly independent of the value of BFI.

To see these more clearly, the dependence of maximum kurtosis on the value of BFI for all ~200 wave-fields in MITWAVE is shown in Figure 7 while that of rogue wave probability on BFI for all wave-fields in MITWAVE is shown in Figure 8. For rogue wave prediction, it is of practical importance to combine BFI and spreading-angle into one new simple parameter.

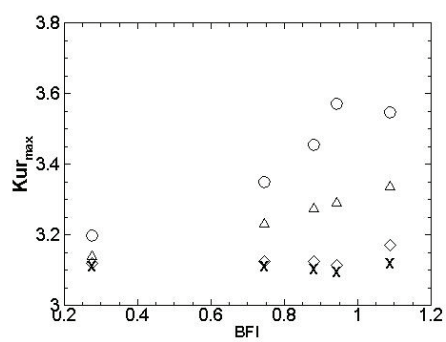


Figure 4. Dependence of the maximum kurtosis on the values of BFI for wave-fields with same steepness ε=0.14 but different spreading-angles: different spreading-angles: Θ=20° (○); Θ=30° (Δ); Θ=60° (◊); Θ=80° (×)

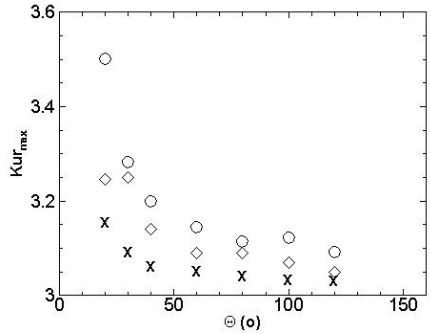


Figure 5. Dependence of the maximum kurtosis on the spreading-angle of the initial wave spectrum for wave-fields with different values of BFI: BFI=0.54 (×); BFI=0.8 (◊); BFI=1.08 (○)

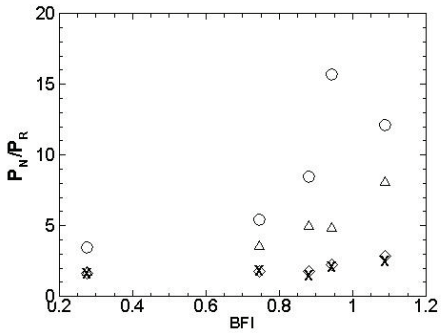


Figure 6. Dependence of the rogue wave statistics, P_N/P_R, on the values of BFI for wave-fields with different spreading-angles: Θ=20° (○); Θ=30° (Δ); Θ=60° (◊); Θ=80° (×)

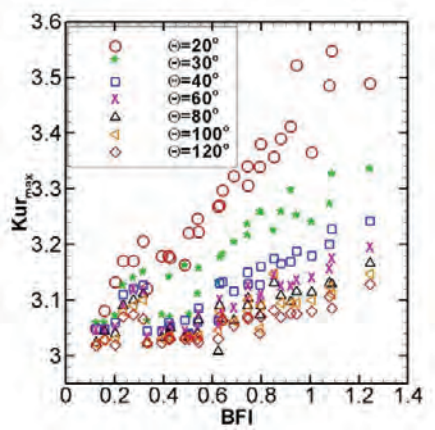


Figure 7. Dependence of the maximum kurtosis on the value of BFI for all ~200 wave-fields in MITWAVE

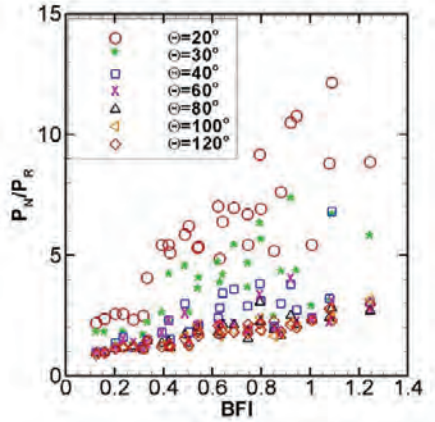


Figure 8. Dependence of the rogue wave statistics on the value of BFI for all ~200 wave-fields in MITWAVE

3.4 Geometric Characteristics of Rogue Waves

The average area of rogue waves scaled by the square of the dominant wavelength, $\bar{A}_R / \lambda_p^2$ for wave-fields with different spreading-angles is shown in Figure 9(a). Not surprisingly, the average area of rogue waves decreases with the spreading-angle. The large variation of rogue wave area observed in the wave-fields with small spreading angle Θ=15° is caused by the significant broadening of the spreading-angle during the wave-field evolution. The ratio between the average area of rogue waves and average area of waves $\bar{A}_R/\bar{A}$ is shown in Figure 9(b). The ratio increases with the spreading angle of the wave-fields.

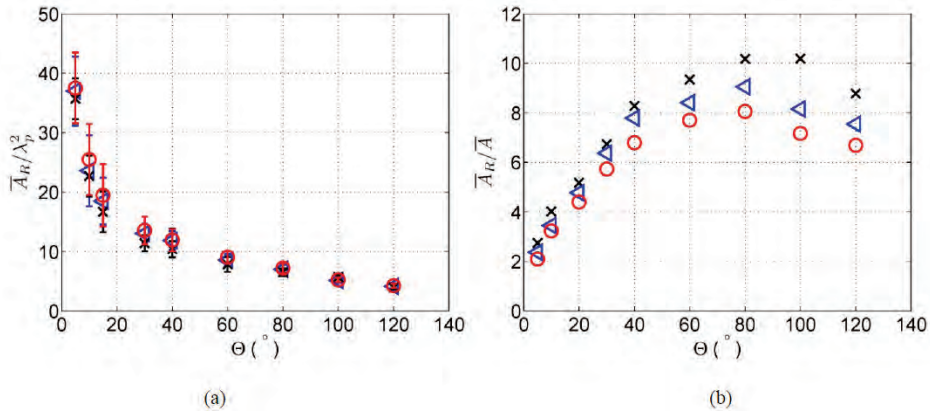


Figure 9. (a)The average area of rogue waves. The error bar indicates one standard deviation. (b) The ratio between the average size of single rogue wave and the average size of one wave. The wave-fields have the same effective steepness ε=0.14 but different bandwidths: γ=3(×); γ=6(<); γ=10(○).

The average wave length and average crest length of the rogue waves for wave-fields with different spectral parameters are shown in Figures 10(a) and 10(b). It is observed that $\lambda_r \approx \lambda_p$ independent of spreading angle Θ and spectral bandwidth γ; while crest length C_r decreases as the spreading angle increases. For given spreading angle, the dependence of the crest length on the spectral bandwidth is not significant. The standard deviation of crest lengths increases as the spreading angle decreases. The large variation of crest lengths in wave-fields with small spreading angles is a result of the significant spectral angle broadening during nonlinear wave-field evolution.

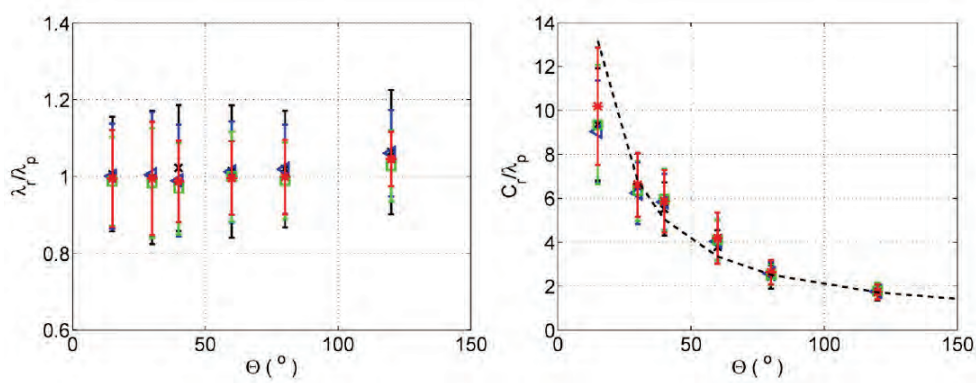


Figure 10. (a) Average wave length of rogue waves; (b) Average crest length of rogue waves as a function of spreading angles. γ=1 (×); γ=3(□); γ=6(□); 10 (*). The error bar is one standard deviation.

4. Conclusion

A direct phase-resolved wave prediction tool, SNOW, is applied to simulate the nonlinear evolution of large-scale three-dimensional ocean surface wave-fields under a variety of sea conditions. Based on the SNOW-simulated nonlinear wave-fields, the dependence of rogue wave statistics on the sea states is investigated. We confirm that the occurrence probability of rogue wave events increases with classic BFI for wave-fields with small spreading angle; while for wave-fields with broad spreading angle, the probability of rogue waves is nearly independent of BFI and BFI alone is not sufficient for

rogue wave prediction. It also suggests that for a wave-field with large spreading angle, as modulational instability diminishes, the probability of rogue wave occurrence can be described by Rayleigh distribution.

5. Significance to DoD

A reliable prediction of rogue wave occurrence in realistic sea environments using phase-resolved SNOW simulations significantly increases the operational safety of naval ships, and is of importance to the design of new Navy advanced ships. Phase-resolved wave-field simulations by SNOW also provide a powerful framework for the assessment and improvement of phase-averaged wave-prediction models, and greatly enhance the capability of ocean wave sensing deployment and data interpretation.

Acknowledgements

We gratefully acknowledge the support of the DoD High Performance Computing Modernization Program and the US Office of Naval Research (N000140810610 managed by Dr. Charles Linwood Vincent) for this study.

References

Alber, I.E., "The Effects of Randomness on the Stability of Two-Dimensional Surface Wavetrains." *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 363(1715), pp. 525–546, 1978.

Benjamin, T.B. and J. Feir, "The disintegration of wave trains on deep water." *J. Fluid Mech*, 27(3), pp. 417–430, 1967.

Dommermuth, D.G. and D.K.P. Yue, "A high-order spectral method for the study of nonlinear gravity waves." *J. Fluid Mech.*, 184, pp. 267–288, 1987.

Dommermuth, D.G., "The initialization of nonlinear waves using an adjustment scheme." *Wave Motion*, 32, pp. 307–317, 2000.

Dysthe, K., "Note on a modification to the nonlinear Schrödinger equation for application to deep water waves." *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 369(1736), pp. 105–114, 1979.

Forristall, G.Z., "Wave Crest Distributions: Observations and Second-Order Theory." *J. Phys. Oceanogr*, 30(8), pp. 1931–1943, 2000.

Janssen, P.A.E.M., "Nonlinear Four-Wave Interactions and Freak Waves." *J. Phys. Oceanogr*, 33(4), pp. 863–884, 2003.

Kharif, C. and E. Pelinovsky, "Physical mechanisms of the rogue wave phenomenon." 22(6), p. 603, 2003.

Kharif, C., J.P. Giovanangeli, et al., "Influence of wind on extreme wave events: experimental and numerical approaches." *J. Fluid Mech.*, 594, pp. 209–247, 2007.

Lavrenov, I.V., "The Wave Energy Concentration at the Agulhas Current off South Africa." *Nat Hazards*, 17(2), pp. 117–127, 1998.

Longuet-Higgins, M.S., "On the statistical distribution of the height of sea waves." *J. Mar. Res*, 11(3), pp. 234–266, 1952.

Mori, N. and P. Janssen, "On kurtosis and occurrence probability of freak waves." *J. Phys. Oceanogr*, 36(7), pp. 1471–1483, 2006.

Onorato, M., A. Osborne, et al., "Observation of strongly non-Gaussian statistics for random sea surface gravity waves in wave flume experiments." *Physical Review E*, 70(6), p. 67302, 2004.

Skourup, J. and N. Hansen, "Non-Gaussian extreme waves in the central North Sea." *J. Offshore Mech. Arct. Eng.*, 119, p. 146, 1997.

Stansell, P., "Distributions of extreme wave, crest and trough heights measured in the North Sea." *Ocean Eng.*, 32(8–9), pp. 1015–1036, 2005.

Tayfun, M.A., "Narrow-band nonlinear sea waves." *J. Geophys. Res.*, 85(C3), pp. 1548–1552, 1980.

Trulsen, K. and K.B. Dysthe, "A modified nonlinear Schrödinger equation for broader bandwidth gravity waves on deep water." *Wave motion*, 24(3), pp. 281–289, 1996.

Tsai, W. and D.K.P. Yue, "Computations of nonlinear free-surface flows." *Ann. Review of Fluid Mech.*, 28, pp. 249–78, 1996.

Toward Efficient Calculations of High Speed Jet Noise

Chris C. Nelson and Alan B. Cain
Innovative Technology Applications Company, LLC, Chesterfield, MO
{ccnelson, abcain}@ITACLLC.com

Philip J. Morris and Yongle Du
Pennsylvania State University, State College, PA
pjmaer@engr.psu.edu, yxd116@psu.edu

Abstract

The noise from the latest generation of military fighter aircraft engines poses a significant health threat to the crews which must work with these systems on the tight confines of an aircraft carrier deck. The Navy has undertaken a long-term comprehensive program to develop technology that can predict the acoustical fields from these systems, and also evaluate proposed mitigation techniques. As part of this effort, ITAC and Penn State have been working together under Navy sponsorship, to extend the CHOPA aero-acoustics solver to enable efficient and accurate predictions of the noise from high-speed jets. To this end, a number of technologies have been added to the solver, including an Immersed Boundary Method, wall-function boundary conditions, and the ability to use computational meshes with non-matching block boundaries. In addition, work is ongoing to implement and validate an implicit scheme, with sub-iterations, which offers the promise of significant speed-up over the original algorithms in CHOPA.

An important aspect of this work is ensuring that larger cases involving many hundreds or thousands of processors on the Department of Defense high performance computing systems can be run successfully and efficiently. To this end, representative military-style jet nozzle simulations have been performed using computational grids which vary from less than six million points to over 200 million points. While more work is required to fully-optimize the code for very-large-scale parallel runs, the code's ability to run large cases has been demonstrated while benefiting from the use of many hundreds of processors.

1. Introduction

In modern low-bypass ratio power plants, the noise from the turbulent, hot, supersonic jet exhaust dominates that generated by other components of the engine, such as the fan and combustor. This is true throughout the flight envelope—at take-off and landing, as well as in high-Mach cruise at altitude. The intense acoustic waves have significant safety implications for ground operations personnel, as well as an environmental impact in the form of noise pollution around military installations.

1.1 Noise Generation in Supersonic Jets

Though the noise generation mechanisms of hot supersonic jets contain some features in common with subsonic jets, the dominant noise mechanisms are quite distinct. Principally, the supersonic convection of the turbulence in the jet shear layer, relative to the ambient speed of sound, results in an efficient direct coupling between the turbulence and the radiated noise. This is the Mach wave radiation. Very intense and highly-directional, it is the dominant noise generation mechanism in the jet downstream arc. For jets that are perfectly expanded, it can be more than 20 dB more intense than noise radiated normal to the jet or in the upstream arc.

When the jet is operating off-design; however, which is often the case for military aircraft engines, two or three additional noise mechanisms may be present. These mechanisms are associated with the interaction between the turbulence in the jet shear layer and the jet's shock-cell structure. The first is broadband shock–associated noise. As the name suggests, this mechanism generates noise at all frequencies, though its peak frequency is higher than that of jet turbulence mixing noise. The large-scale turbulent structures in the jet convect through the jet's shock-cells, where they intersect the jet

shear layer. As this happens, the turbulent structures generate a sequence of coherent sources at these locations. The resulting constructive interference between the noises radiated by these sources yields the broadband shock-associated noise that radiates at large angles to the jet downstream axis and dominates the noise at these large angles.

The broadband shock-associated noise that radiates towards the jet nozzle can trigger turbulent structures that then propagate downstream, interact with the jet's shock-cells, and radiate shock-associated noise. This sequence can form a phase-locked loop and can generate an intense tonal noise source known as screech. Screech is rarely observed in single, full-scale hot jets, but it can occur in twin jet configurations. When screech occurs, because of its high-intensity, it can cause structural damage to the jet nozzle and nearby structural elements.

Finally, the levels of noise generated by hot supersonic jets can be so high that nonlinear acoustic mechanisms occur. In particular, a phenomenon called "crackle," which is identified by shock-like features in the noise time history is observed. In addition, the propagation of the sound is nonlinear, and simple linear extrapolation of the sound in the near-field results in an under-prediction of high-frequency noise.

Each of these features of hot supersonic jet noise presents challenges for numerical simulations.

1.2 Present Approach

The primary goal of the work which has led to this paper is the development of an advanced computational aero-acoustics (CAA) tool, named Compressible High-Order Parallel Acoustics (CHOPA), for accurate prediction of the acoustics of realistic supersonic jets as applied to Navy aircraft. The notion of a "practical" tool has been key to our approach throughout this effort. In performing numerical simulations of jet flows and their radiated noise, our contention is that a balance is needed between the computational cost required for the noise predictions and the accuracy sought in the simulations. Without sufficient accuracy, obviously, the solution is pointless; but likewise, if timely solutions cannot be obtained, the utility of the resulting tool is marginal at best.

We recognize that, for high-speed jets, the dominant noise sources are the large-scale structures in the jet shear layer. Since these turbulent structures convect supersonically with respect to the ambient speed of sound, they are very efficient noise radiators. This scale of turbulence is captured well with moderate grid sizes (on the order of 4–6 million grid-points) at a moderate computational cost using resources that are available to a wide group of users in government and industry.

In summary, our approach to the simulation of supersonic jets with CHOPA seeks an accurate resolution of the larger-scale structures that make up the dominant noise sources. In normal use (on smaller systems), we sacrifice a certain level of accuracy in the fine scales of motion so that we can obtain solutions in a reasonable amount of time on moderate computational resources available. The algorithms used are still valid for higher frequencies; however, if a sufficiently fine-grid is used and one can afford the computational expense. In the discussion below, we explore the use of the code with computational meshes as large as 200 million points, which we solve using more than 1,500 CPUs.

2. Compressible High-Order Parallel Acoustics (CHOPA) Solver

Development on the CHOPA solver began at Pennsylvania State University (PSU) in 1997. This code was originally developed for aero-acoustic studies of airfoil gust response (Lockard and Morris[1]) and cavity acoustics (Shieh and Morris[2]). It has been adapted more recently for jet noise simulations (Paliath and Morris[3,4]).

2.1 Baseline Implementation

CHOPA solves the compressible Navier-Stokes equations in a finite-difference formulation on multi-block structured computational grids. A fourth-order Dispersion-Relation Preserving scheme is used for spatial differencing. A sixth-order filter is used to prevent the unphysical build-up of energy in the highest resolvable scales. Second-order artificial dissipation is used to keep CHOPA stable near discontinuities, such as shocks. There are three options for controlling the activation of the second-order dissipation: a pressure-based switch, a density-based switch, and a combination of both.

In the past, a four-stage Runge-Kutta scheme was used for steady-state cases. A different four-stage fourth-order Runge-Kutta scheme was available for time-accurate simulations. In practice, this was only used for smaller cases; however, because of the limitation on the time-step imposed by the explicit scheme. Most larger cases, such as the jet noise simulations related to the work discussed here, have been solved using the second-order dual-time-stepping method.

By design, CHOPA uses parallel processing to obtain flow solutions. Standard Message Passing Interface (MPI) library calls are used, and the code had been successfully run with a variety of MPI implementations. Each block of a multi-block mesh is run on a minimum of one MPI process. At the user's discretion, individual blocks can be further

decomposed to run on multiple processors. In this way, the parallel load can be reasonably balanced across all the available processors for a run.

A broad selection of boundary conditions is available to enable a variety of flow configurations to be modeled. These include a total condition inflow boundary for nozzle inflow, a radiation boundary condition for the far-field, viscous wall conditions, and characteristic-based boundary conditions.

Turbulence modeling in CHOPA, when required, is available through a number of variations of the Spalart-Allmaras turbulence model. For steady-state flows, the original Reynolds-averaged Navier-Stokes (RANS) version of the Spalart-Allmaras model is used. For unsteady simulations, the Detached-Eddy Simulation (DES) version of the Spalart model[5] has been implemented. In previous jet simulations, it has been found that the DES model is too dissipative, and instead a version which uses the ideas of the "Implicit Large-Eddy Simulation (ILES)" approach is recommended[6,7]. The ILES method implemented in CHOPA uses a standard Spalart model in near-wall regions. Away from walls; however, the model is turned off, and the dissipation inherent in the numerical discretization is relied upon in place of a formal turbulence model.

2.2 Code Enhancements

Innovative Technology Applications Company (ITAC) and PSU have been working together under a Navy STTR (Phase I and Phase II) to modify CHOPA to better enable it to model hot supersonic jets. As a result of that work, along with other concurrently running projects, a number of important features have been added to the code.

For example, an implicit residual smoother algorithm has been added to increase code stability (allowing larger time-steps to be taken). Another option which has been added, that can greatly speed-up simulations, is a multi-grid algorithm. CHOPA now has Full-Approximation Storage (FAS) multi-grid which can be used with either single-block or multi-block computational meshes (with point-matched boundaries).

In addition, a wall-function boundary condition was successfully demonstrated in CHOPA recently. When fully-generalized and integrated into the code, this will allow larger time-steps and reduce the number of points required for viscous regions where the fine details of boundary-layers are not important. Ultimately, this will enable users to obtain solutions faster.

2.2.1 Immersed Boundary Method and Non-Matching Boundaries

To improve the general applicability of the solver, several new algorithms have been implemented. One of the most significant of these is the ability to used immersed boundaries. In CHOPA, this approach is typically used when one has relatively small geometric features on a larger geometric configuration (e.g., chevrons on a jet nozzle). In such a case, rather than requiring the creation of a detailed mesh around each chevron, an immersed boundary approach can be used. First, a mesh is created for the overall geometry without the detailed feature. Then, this mesh is compared with the surface of the detailed feature, and points which would be inside the feature are tagged as "immersed". This process is illustrated conceptually in Figure 1. This knowledge is then used to approximate the feature in the simulation.

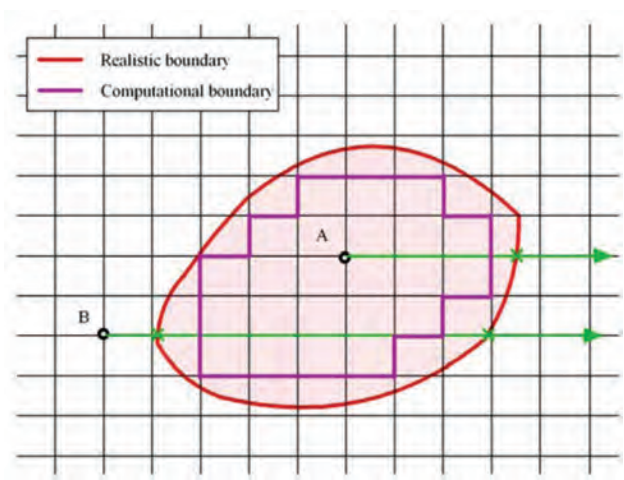


Figure 1. Conceptual drawing of the Immersed Boundary Method

Using immersed boundaries in this way allows us to greatly simplify the grid-generation process, which would otherwise require significant effort in order to create a high-quality body-conformal mesh that includes all the small features.

In many cases, when using immersed boundaries in this way, it is desirable to increase the grid-density in the immediate vicinity of the immersed features. Rather than propagate the increased point-density throughout the entire domain, CHOPA now has the capability to solve multi-block meshes with non-point-matched boundaries. This can be used more generally to add resolution in specific areas that need it. An example of this is shown in Figure 2, where the inner region of the Gaussian pulse mesh has increased resolution compared to the outer blocks.

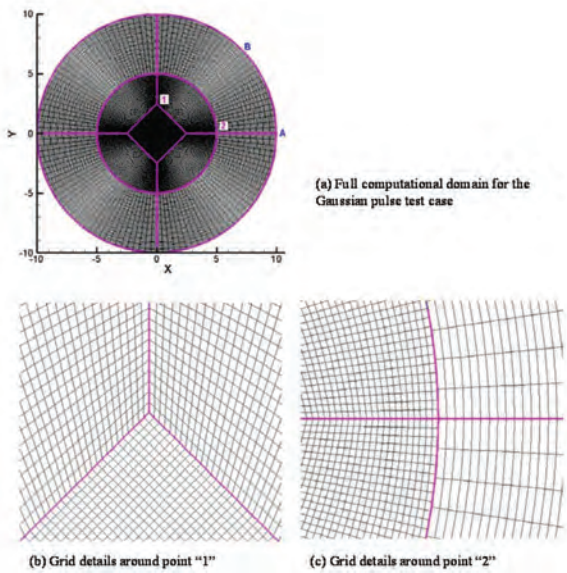


Figure 2. Computational mesh for a Gaussian pulse test case illustrating use of non-matching block boundaries

2.2.2 Implicit Solver

The above modifications have largely been carried out at PSU, with ITAC assisting with testing and debugging. ITAC has also been working to implement an implicit solver into CHOPA. This technique offers the promise of being able to run simulations with larger time-steps (in the case of steady-state runs) and/or fewer sub-iterations per time-step for unsteady runs, compared to the current explicit dual-time marching method. In addition, though a matrix solve is required, only a single right-hand side evaluation is necessary per time-step (sub-iteration), which has been observed to be significantly faster than the multi-stage Runge-Kutta process, which requires RHS evaluations at each stage. Other researchers have successfully modeled a wide-variety of unsteady flow physics, including acoustics applications, using similar implicit schemes with second-order-accuracy in time[8,9]. There are three essential components to this approach:

1. A basic steady-state implicit flow solution algorithm provides a foundation
2. A second-order temporal scheme is added to the basic scheme
3. A Newton sub-iteration algorithm is wrapped around the first two components to ensure that the interior and boundaries are kept in sync.

Properly implemented, such an implicit scheme can operate in steady-state or time-accurate mode, with either first- or second-order time-marching. It can also be run with or without Newton iterations. This allows the user maximum flexibility, and developers benefit also, since they can focus on validating one piece at a time.

The implicit scheme in CHOPA has been successfully run on a number of test cases which include both laminar and turbulent flow on single-block and multi-block meshes. Solutions have also been successfully run in parallel with additional domain-decomposition of individual blocks.

One such test of the implicit scheme, which demonstrates its potential for use in impinging jet flows, is a laminar simulation of an impinging jet. This run was performed on a coarsened mesh which had approximately an order-of -magnitude less points than a full production mesh.

Figure 3 shows instantaneous Mach number contours on a centerline cross-section of the demonstration impinging jet case. A plot of the magnitude of the pressure gradient is shown in Figure 4. While the coarseness of the computational mesh and the lack of a turbulence model definitely affect the results, CHOPA is clearly able to model the unsteady nature of the flow, and the basic plume structure.

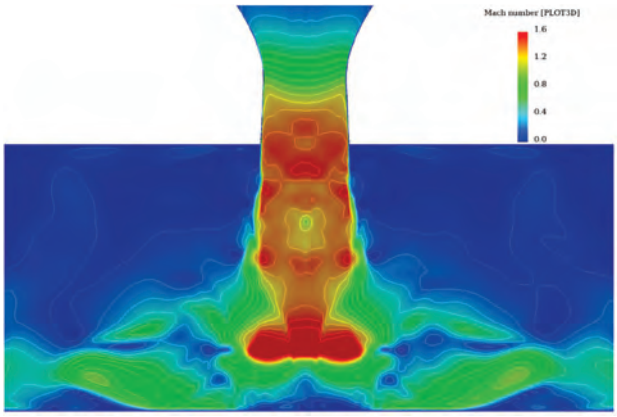


Figure 3. Mach number contours in sample impinging jet case computed with CHOPA using an implicit solver

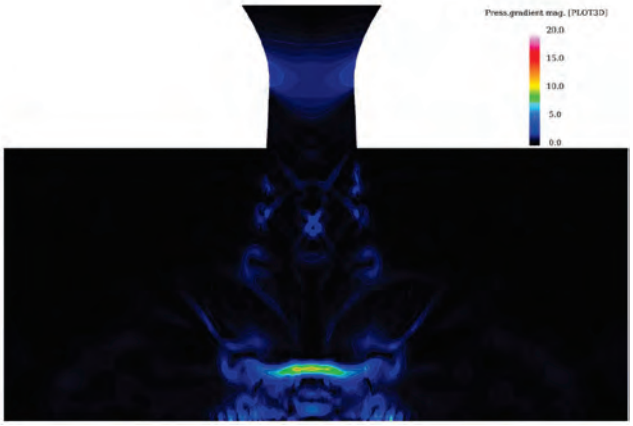


Figure 4. Contours of the magnitude of pressure gradient in a coarse-grid simulation using the implicit scheme

Timings were made to compare the implicit scheme with the dual time-marching algorithm for this case. Comparing just the time for a single sub-iteration, the second-order-implicit scheme showed a speed-up of 1.8. In addition, the Newton sub-iteration method generally requires fewer iterations to converge than the dual time-marching scheme. In this case, six Newton sub-iterations were used with the implicit scheme, compared to fifteen for the dual time-marching scheme. The overall speed-up for the implicit scheme, therefore, was on the order of 4.5 or 5.0 (some variation was noted due to the time per sub-iteration fluctuating—particularly for the dual time-marching run). The promise offered by this level of performance gain is substantial, and work to enable more general use of the implicit scheme is ongoing.

3. Military-Style Nozzle Test Case

The parallel timings reported here were performed on a military-style nozzle which can be run in a baseline configuration, or fitted with chevrons for noise reduction. The current simulations were run on the baseline nozzle without chevrons.

Figure 5 shows the geometry of this faceted nozzle. The design Mach number for this configuration is 1.5. Supersonic hot jet noise simulations have been performed to predict the noise emanating from the nozzle at three different operating conditions, which correspond to exit-Mach numbers of 1.36, 1.47, and 1.56. The total temperature ratio for all these simulations is 3.0. The nozzle throat has a diameter of 4.45 inches.

The computational mesh created for this configuration contained 5.89M grid-points. To faithfully simulate the jet noise from this configuration, all the geometric details, including the faceted inner contours, and the finite nozzle thickness, are captured in the computational mesh. Several views of the mesh are shown in Figure 6. Figure 6(a) shows the center-line cross-section of the entire mesh. Figure 6(b) shows a cut through the mesh at the trailing-edge of the nozzle. As is usual with CHOPA, the mesh has been designed to avoid having a singular axis at the center-line. Figure 6(c) shows a close-up of the nozzle region in a center-line cross-section. The finite-thickness of the nozzle lip is resolved on the grid, as illustrated by Figure 6(d).

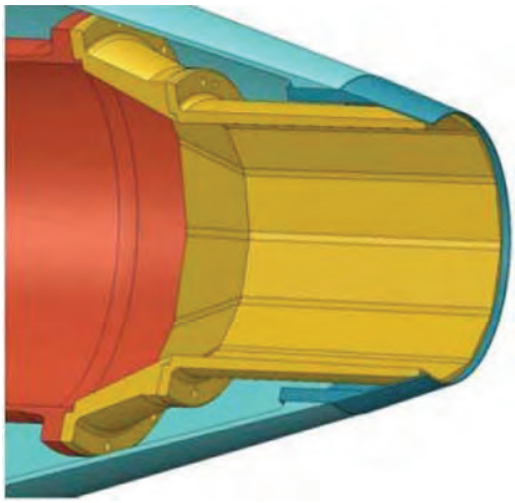


Figure 5. Military-style faceted nozzle baseline configuration

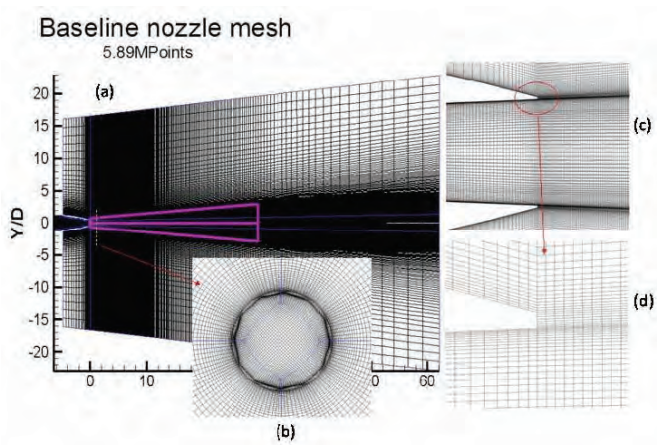


Figure 6. Baseline computational mesh for the military-style nozzle simulations

4. Lessons Learned from Porting to HPC Systems

A significant level of effort has been expended to enable the team to take advantage of the high performance computing (HPC) systems at the Department of Defense (DoD) Supercomputing Resource Centers (DSRCs). To that end, we obtained accounts on seven systems at four separate DSRCs. Work was undertaken to port CHOPA to each of these different systems and verify that the solver works properly.

4.1 System Quirks

The process of working with CHOPA on the various HPC systems where we have obtained access has revealed a number of issues, not only with CHOPA itself, but this process has revealed several quirks in these HPC systems as well. For example, it was discovered that, on the 'jade' system, the MPI variables MPICH_UNEX_BUFFER_SIZE, and MPICH_MAX_SHORT_MSG_SIZE must be tweaked to find a combination which allows CHOPA to run for very large cases, such as the 200+ million point jet simulation. This effort has also resulted in the discovery (and fixing) of a number of programming issues.

To date, we have run small test cases successfully on almost all of the systems. Only two have been used for full-scale production runs, however: 'einstein' at the NAVY DSRC and 'jade' at the Army's Engineer Research and Development Center (ERDC) DSRC. Difficulties have been encountered on several systems as we have worked to port the code. On the 'diamond' system at ERDC, for example, we were unable to run anything more than the simplest of test cases. We were never able to understand the cause of this, or find a solution.

Similarly, the 'garnet' system, also at ERDC, has had problems running the production jet flow simulations. Support personnel have been contacted, and this issue continues to be investigated, but at the time of this writing, no explanation for this behavior has been found. In general, we work to understand any such issues and rectify any problems found to ensure that the code is as widely-portable as possible.

4.2 Compiler Bugs

One of the more unusual findings that arose from our work to port CHOPA to the various HPC systems highlighted the importance of maintaining a diversity of development tools. The problem arose early on in our porting efforts. While we had some initial success, it became clear fairly quickly that something was wrong, though it was not immediately clear whether the problem was with the coding of CHOPA, the compilers used on the various systems, the MPI libraries (or other system libraries) installed on the HPC system, or a combination of all three.

Working with the system support technicians at ERDC, it was determined that, for reasons then unknown, some inter-processor messages were being sent (by MPI functions) that were of a different size than the receiving processors were expecting. Even on other HPC systems where CHOPA appeared to run correctly for some test cases, the code did not exhibit the same stability observed on the Penn State systems, where it was originally developed and where it has long been used to produce excellent results.

In light of the above findings, it was decided to step back and perform more extensive testing of the code on ITAC's local systems. Both the then-latest version of CHOPA used at PSU (obtained from Dr. Morris) and an ITAC-modified version were installed on an ITAC development system.

It was quickly discovered that the same problem seen on the ERDC HPC system was also present on the ITAC system, as the MPI libraries emitted errors complaining of sent message lengths being different from expected. An extended round of debugging was undertaken to discover the cause of the problem.

The cause of at least some of the problems encountered on the ERDC systems (and ITAC's) was eventually traced back to a compiler defect in the Intel FORTRAN compilers that were used. Specifically, when performing the initial processing for coupled boundary conditions, there are several lines of code similar to:

```
bcinfo%itrgt1 = abs( bcinfo%itrgt1 )
```

For reasons yet unknown, when compiling with any optimization activated, the Intel Fortran compiler (both v11.1 and v12) would ignore the absolute-value function on these lines and return negative numbers. This resulted in corrupted data which eventually crashed the solver. By trial-and-error, a workaround was found which uses (rather bizarre) constructs similar to:

```
if ( bcinfo%itrgt1 < 0 ) then
  itmp = -bcinfo%itrgt1
  if ( itmp < 0 ) itmp = -itmp
  bcinfo%itrgt1 = itmp
end if
```

This has been implemented in all subsequent versions of CHOPA.

4.3 Effect of Compiler Optimization on Runtime

One question which was raised in discussions of the above work regards the need for using compiler optimization. What is the impact of optimizing the code on the execution speed? Why not simply use an unoptimized, or minimally-optimized, version of the code?

To answer these questions, a simple test case was run using both an optimized version of the code, and a version which was compiled with code optimization disabled. The case chosen for the timing tests was a Gaussian pulse in stationary flow. This case used the same mesh referred to above in the discussion of the non-matching boundary conditions (Figure 2). The total grid contained 67,300 points.

For the purposes of timing, only ten physical time-steps were run (with a non-dimensional time-step of 0.1). The 4th-order Runge-Kutta time-marching algorithm was used to advance the solution in time. When run with this algorithm, CHOPA uses a sub-iteration strategy where the RK scheme advances in time according to a given CFL number (to maintain stability) until the solution nears the user-specified physical time-step. At this point, the sub-iteration time-step is adjusted to ensure that the solution is computed for the time that the user expects to see. Thus, for the current test case, there are actually six full RK-steps for each user-specified time-step.

The same case was run on several different systems as part of the overall effort to ensure portability and reliability of the solver. These systems included both 32 bit and 64 bit Linux operating systems (multiple versions of the latter). Both Intel 11.1 and 12.0 FORTRAN compilers were used to compile the code.

The effect of optimization on execution speed was found to be quite striking. On a representative 4 CPU system, the fully optimized code ran the test case in less than 34 seconds. The version with optimization disabled, on the other hand, took just over 595 seconds. As desired, the results from both optimized and non-optimized versions were identical (allowing for numerical precision).

In addition, the effect of optimization level on run-time was investigated. On the same test case, we found that using Intel's "-O3" level of optimization rather than the default "-O2" results in approximately a 20% speed-up of the code. While this is not terribly significant for a small test case such as the Gaussian pulse, it saves days of compute-time for the much larger jet cases when this sort of speed-up can be obtained without compromising the results. The same preliminary tests indicate that the results were not adversely affected by increasing the optimization level.

As a side note, it was found that, when using MPI's wrapper scripts for compiling (e.g., 'mpif90' and 'mpicc') it was not always straightforward to avoid optimization. These scripts are often silently activating various optimization-related compiler switches which act at cross-purposes to what the user may be seeking to accomplish.

4.4 Testing CHOPA's Portability and Reliability

After addressing the issues discussed above to work around the compiler problems, as well as modifying the code to eliminate all compiler warnings (and errors) and incorporating an MPI wrapper library from Dr. Hixon (which provides a modern FORTRAN interface to MPI calls), a renewed effort was undertaken to port CHOPA to the various HPC resources to which we had access. Accordingly, CHOPA was uploaded, compiled, and tested on five of the seven systems available to us.

In contrast with the previous attempts to do this, the code compiled and ran a small circular cylinder test case successfully on all five of the systems. In addition, the same code and test case were compiled and executed on both Penn State's systems and ITAC's in-house cluster in order to ensure that the modifications had not broken compatibility with them. This verified the ability of the CHOPA to solve at least a simple test case using a wide variety of hardware, compilers, and MPI libraries.

When attempting to use CHOPA for larger production aero-acoustic simulations of jet flows; however, additional difficulties were encountered on many of the systems (such as those discussed above regarding "diamond" and "garnet" at ERDC). As a result, until very recently, the only HPC systems where CHOPA could reliably be run for large cases were the Cray XT4 ("jade") at ERDC and the Cray XT5 ("einstein") at NAVY. These systems have similar architectures and both use versions of the Portland Group compilers. On other systems, such as "hawk" at the US Air Force Research Laboratory (AFRL) and "harold" at the US Army Research Laboratory (ARL), we encountered persistent problems. At the time of this writing, it appears that the SGI Altix ICE 8200 at ARL ("harold") is now working for larger cases. That being the case, it is likely (though not yet demonstrated) that the similar SGI Altix 4700 ("hawk") may now be usable as well.

4.5 Parallel Performance

As mentioned above, CHOPA has long used parallel processing to reduce the time required to obtain solutions. Most of these runs; however, have been performed on small and medium-sized clusters at PSU and ITAC. In such cases, the number of cores employed for the run was relatively small—between 50 and 120 for most cases. The Navy customer (and DoD in general); however, has vastly larger systems at their disposal, and thus the question arose as to how CHOPA would perform when using more parallel resources.

As a consequence, the performance of CHOPA on parallel systems has been investigated using two different methods. In the first, the same case was used, with the same computational mesh, and was run using different numbers of processors on the same system. This tests the ability of CHOPA to scale when the amount of work remains the same, but the available processing power increases. In the second approach, the same case was run on the same machine(s), but the size of the computational mesh was increased to match the number of cores being used. This tested the ability of the code to scale the parallel performance when the number of processors increases, but the work per processor remains roughly the same.

4.6 Parallel Scaling with Increasing CPUs for the Same Size Case

For the first approach to parallel scalability testing, the ITAC version of CHOPA was used to run the military-style nozzle test case (with a jet-Mach number of 1.47) in steady-state mode on the 'jade' system at the Army's ERDC supercomputing resource center. A short timing-run was run three times using different numbers of processors each time. A nominal baseline-run used 102 cores. The other two runs employed 204 and 408 cores.

The baseline-run with 102 cores required approximately 8.05 seconds per iteration. Doubling the number of cores dropped the time to about 5.75 seconds per iteration. Further doubling the number of CPUs to 408 reduced the time to 4.94 seconds per iteration. This represents a relative parallel efficiency of 70% for the 204 processor run, but only 41% for the 408 core run.

Obviously, this represents a significant drop-off in the efficiency of the code for this case. Note; however, that when running this case, which has 5.89 million grid-points in the original mesh, on 408 processors, there are only 14K points (on average) being computed on a given core. Distributing such a small mesh over hundreds of cores was not; however, an optimal test. This is because the number of points on block-to-block boundaries, relative to the number of interior points in each block, inevitably increases as the number of processors goes up. This means that the amount of less-efficient inter-processor communication required between the block boundaries increases relative to the more-efficient solution of the flow on the interior grid-points.

We estimate that use of significantly more than about 120 CPU-cores on such a mesh probably incurs some degree of reduction in parallel efficiency. It is also important to note that, because of the way domain decomposition and parallel processing is currently done in CHOPA, it is likely that parallel efficiency would drop off if significantly fewer than 102 cores were used. The reason for this is that while the original mesh has only fourteen blocks, the block sizes vary dramatically (from 65K to 1.2M points). Since CHOPA currently only allows a single-mesh to be solved by a given MPI process, using too few processes can result in a significant load imbalance.

If; however, a much larger mesh were used, then many more cores might be usable, while maintaining a reasonable level of parallel efficiency. Following this line of thinking, we estimated that, for example, if the military-style nozzle was refined to contain a total of 250 million points, we might be able to achieve good parallel performance using as many as 5,000 cores, or even more. This led us to the second study of parallel performance.

4.7 Parallel Scalability When Adding CPUs and Maintaining Workload

Here, we created three refined versions of the initial military-style nozzle mesh with 32.9M, 106.3M, and 200.5M grid-points, respectively. Each grid, like the original baseline mesh, consisted of 14 blocks. These cases were then run on the Navy's 'einstein' system using 284, 884, and 1,676 CPUs, respectively.

The test case was run using the second-order-implicit solver with Newton sub-iterations. The ILES variant of the Spalart turbulence model was used, and both point-by-point and FWH surface-data were saved in the course of the run. A total of two hundred time-steps with six sub-iterations per step were run for all the meshes.

For comparison, the original 5.89 million grid-point mesh was run with the same options on 175 CPUs using the same options. This case acted as a 'control' and represents something of a "normal" jet acoustics application of CHOPA.

The same set of runs was also performed on the 'jade' system at ERDC. It should be noted that runs were also attempted on the 'garnet' system (also at ERDC), but that (as discussed above) there are some still-unresolved issues when running CHOPA on that system, and we were unable to complete the tests there.

The results, summarized in Tables 1 and 2, indicate that the code performed quite similarly on both the systems tested. The newer and somewhat more powerful 'einstein' system was slightly faster, and showed better parallel performance, but 'jade' was not far behind. The initial scaling from the original mesh to the 32.9 million point grid was quite successful; with the CPU-time per point indicating more than 90% (estimated) efficiency relative to the control-run on the smaller mesh. The estimated efficiency drops-off considerably for the larger meshes; however, as the 100 million grid-point grid solution was only about 60% as efficient in CPU-time per grid-point (relative to the performance on the smallest mesh). The largest mesh showed less than 50% efficiency relative to the small-grid run.

There are several factors which may be at work in these cases. First, these simulations did include file input/output (I/O), and that has in the past been found to deteriorate parallel performance. Also, the domain decomposition was imperfect in all cases, in that the number of points per processor was not constant. It might, therefore, have been better to perform the analysis using the largest number of points per processor to determine the system load (instead of the average). There may also be programming practices which, while perfectly acceptable when CHOPA is used on smaller systems, degrade parallel performance on very large systems.

Table 1. Parallel performance of CHOPA on a Cray XT5 system

Einstein

Case	Points in Grid	No. CPUs	Points per CPU	CPU-Time/Pt	Inner loop (sec/iter)
Grid 1	5.89 million	175	33,630	0.0572	1.3
Refine 1	32.9 million	284	115,690	0.0597	4.7
Refine 2	106.3 million	884	120,239	0.0917	7.2
Refine 3	200.5 million	1676	119,647	0.1200	10.3

Table 2. Parallel performance of CHOPA on a Cray XT4 system

Jade

Case	Points in Grid	No. CPUs	Points per CPU	CPU-Time/Pt	Inner loop (sec/iter)
Grid 1	5.89 million	175	33,630	0.0604	1.4
Refine 1	32.9 million	284	115,690	0.0647	5.1
Refine 2	106.3 million	884	120,239	0.1022	8
Refine 3	200.5 million	1676	119,647	0.1360	11.6

As a final note on parallel performance, as this was being written, an initial-run of the baseline grid was completed on the SGI Altix ICE 8200 at ARL ("harold"). On this system the CPU-time per grid-point was observed to be 0.023 seconds. This represents an almost three-fold speed-up relative to the Cray systems discussed above. Since the nominal capability of this system is only 5 percent better on a per core basis, there is clearly some other factor at work. One possible factor is the choice of compilers. The SGI system uses Intel compilers, instead of the Portland Group software used on the Crays. As our group has more experience with Intel, we were able to use specific optimization flags which are known to work with CHOPA. With the Portland Group compilers, we use only the default options.

5. Summary and Conclusion

In summary, the CHOPA solver has been ported to a number the DoD HPC systems. While more difficulties were encountered than originally expected for such a generic FORTRAN plus MPI programming model, the code has been successfully used on a number of these systems.

Among the challenges encountered have been compiler bugs and systems where CHOPA does not run properly, but no useful error messages are obtained. In addition, it has been observed in several cases that systems will run smaller test cases, but then fail on larger production runs.

Many of these obstacles have now been overcome, and we have demonstrated an ability to run large (200 million point) cases on more than 1,500 processors. While parallel efficiency has not been as great as one would desire, the time-to-solution is still improving as we add processors. Therefore, users of CHOPA can have confidence that good use can be made of any HPC resources they may be given access to.

Also, the newest results from the SGI Altix system at ARL indicate that perhaps CHOPA has potential for significant performance gains compared to what was observed in the other tests presented here. The reasons for the improved performance on the Altix may be related to compiler options, the MPI library used, or the particular system architecture. More research is required to fully understand the results.

Acknowledgments

This work was supported by NAVAIR Propulsion and Power SBIR/STTR contracts N68335-09-C-0370, N68335-10-C-0407, and the authors also acknowledge the support provided by the F-18 program office. Any conclusions are those of the authors and not necessarily those of NAVAIR or the DoD.

Computer resources for this work were provided by the DoD High Performance Computing Modernization Program; the support of the NAVY DoD Supercomputing Resource Center and the US Army Engineer Research and Development Center is noted in particular.

References

1. Lockard, D.P. and Morris, P.J., "The Radiated Noise From Airfoils in Realistic Mean Flows." *AIAA Journal*, 36(6), pp. 907–914, 1998.

2. Shieh, C.M. and P.J. Morris, "Comparison of Two-and Three-Dimensional Turbulent Cavity Flows." *AIAA Paper 2001-0511*.

3. Paliath, U. and P.J. Morris, "Prediction of Jet Noise from Circular Beveled Nozzles." *AIAA Paper 2005-3096*, 2005.

4. Paliath, U. and P.J. Morris, "Prediction of Jet Noise from Rectangular Nozzles." *AIAA Paper 2006-0618*, 2006.

5. Spalart, P.R., Jou, W.-H., Strelets, M., and Allmaras, S.R., "Comments on the Feasibility of LES for Wings and on a Hybrid RANS/LES Approach." *Advances in DNS/LES—Proceedings of the First AFOSR International Conference on DNS/LES*, 1997.

6. Shur, M.L., Spalart, P.R., and Strelets, M.Kh., "Noise Prediction for Increasingly Complex Jets. Part I: Methods and Tests." *International Journal of Aeroacoustics*, 4(3-4), pp. 213–246, 2005.

7. Shur, M.L., Spalart, P.R., and Strelets, M.Kh., "Noise Prediction for Increasingly Complex Jets. Part II: Applications." *International Journal of Aeroacoustics*, 4(3–4), pp. 247–266, 2005b.

8. Visbal, M.R., and Gaitonde, D.V., "High-Order-Accurate Methods for Complex Unsteady Subsonic Flows." *AIAA Journal*, Vol. 37, No. 10, pp. 1231–1239, 1999.

9. Rizzetta, D.P., Visbal, M.R., and Morgan, P.E., "A High-Order Compact Finite-Difference Scheme for Large-Eddy Simulation of Active Flow Control (Invited)." *AIAA Paper 2008-526*, 2008.

6.3 Advanced Technology Development

Air Force

Application of the Graphics Processing Unit to Computational Electromagnetics Using the Finite Element Method

Huan-Ting Meng, Baolin Nie, and Jian-Ming Jin
University of Illinois at Urbana-Champaign, Urbana, IL
{meng2, baolinn, j-jin1}@illinois.edu

Steven Wong
Dynamics Research Corporation, Wright-Patterson AFB, OH
swong@drc.com

Charles Macon
US Air Force Research Laboratory (AFRL), Wright-Patterson AFB, OH
charles.macon@wpafb.af.mil

Abstract

General-purpose graphics processing unit (GPU) computing environments, such as the Compute Unified Device Architecture (CUDA) by NVIDIA, offer the capability to accelerate the solution process of computational electromagnetics. However, due to the communication-intensive nature of the finite-element algorithm, both the assembly and the solution phases cannot be implemented via the fine-grained, many-core GPU processors in a straightforward manner. In this paper, we identify the bottlenecks of the GPU parallelization of the finite-element method for three-dimensional antenna analysis, and we propose potential solutions to alleviate the bottlenecks.

1. Introduction

General-purpose graphics processing unit (GPU) computing environments, such as the Compute Unified Device Architecture (CUDA) by NVIDIA, offer the capability to accelerate the solution process of computational electromagnetics. However, due to the communication-intensive nature of the finite-element algorithm, both the assembly and the solution phases cannot be implemented via the fine-grained, many-core GPU processors in a straightforward manner. This research focuses on three practical goals. First, modification of the existing CPU code has to be kept minimal. Second, memory usage on the GPU needs to be conserved. And last, good parallelization utilizing the various hardware advantages of the GPU needs to be achieved. This paper is organized as follows: Section 2 formulates the finite-element antenna analysis. Section 3 introduces the GPU software and hardware architectures, as well as some remarks on modifying existing codes. Section 4 discusses the assembly of the finite-element algorithm with a traditional CPU computation and compares it to a novel GPU computation. Section 5 discusses the solution of the finite-element matrix with a CPU computation and compares it to a novel GPU computation. The goal is to accelerate an existing finite-element code using GPU, but without major modifications on the existing code or the need of re-writing the entire code.

2. Antenna Analysis Formulation

This section lays out the finite-element formulation for the antenna analysis[1]. For antenna analysis, the field satisfies the vector Helmholtz equation

$$\nabla \times \left(\frac{1}{\mu_r} \nabla \times \bar{E} \right) - k_0^{\,2} \varepsilon_r \bar{E} = -jk_0 Z_0 \bar{J}_{\text{imp}} , \tag{2.1}$$

with the Dirichlet boundary condition on the surface of the perfect electric conductor (PEC) where $\hat{n} \times \bar{E} = 0$ and the mixed boundary condition on the absorbing boundary and the waveguide port, both of which take the form

$$\hat{n} \times \left(\frac{1}{\mu_r} \nabla \times \vec{E} \right) + \vec{P}(\vec{E}) = \vec{U}^{\text{inc}} , \tag{2.2}$$

where

$$\vec{U}^{\text{inc}} = \hat{n} \times \left(\frac{1}{\mu_r} \nabla \times \vec{E}^{\text{inc}} \right) + \vec{P}(\vec{E}^{\text{inc}}) , \tag{2.3}$$

$$\vec{P}(\vec{E}) = jk_0 \hat{n} \times (\hat{n} \times \vec{E}) . \tag{2.4}$$

Equation 2.2 corresponds to the first-order absorbing boundary condition (ABC) on the absorbing boundary and the transverse electric and magnetic field (TEM) waveguide-port boundary condition (WPBC) on the waveguide-port. Specifically, since the only source of the excitation comes from the waveguide-port, there will be no incident plane-wave contribution to the ABC as in the scattering case. The $\vec{U}^{\text{inc}}$ term for the ABC is thus zero, leaving only the $\vec{U}^{\text{inc}}$ contributed from the waveguide-port.

To seek a finite-element solution, the computational domain is discretized into tetrahedral elements. To formulate the problem via the Ritz method, the functional of the problem is written as

$$\begin{aligned} F(\vec{E}) = \frac{1}{2} \iiint_{\Omega} \left[\frac{1}{\mu_r} (\nabla \times \vec{E}) \cdot (\nabla \times \vec{E}) - k_0^{\,2} \varepsilon_r \vec{E} \cdot \vec{E} \right] dV + \oiint_{S_{\text{ABC,port}}} \frac{jk_0}{2} (\hat{n} \times \vec{E}) \cdot (\hat{n} \times \vec{E}) \, dS_{\text{ABC,port}} \\ + \iint_{S_{\text{port}}} \vec{E} \cdot \vec{U}^{\text{inc}} dS_{\text{port}} \ . \end{aligned} \tag{2.5}$$

To approximate the electric field in each element, the vector basis functions $\vec{N}_j(x, y)$ are used to expand the field as

$$\vec{E}(x, y) = \sum_{j=1}^{N} \vec{N}_j(x, y) E_j \ , \tag{2.6}$$

where N is the total number of unknowns in the computational domain. Taking the derivative of the functional, the variational problem in the computational domain can be converted into the system of equations of the form

$$\left\{ \frac{\partial F}{\partial \vec{E}} \right\} = [K]\{E\} - \{b\} = \{0\} , \tag{2.7}$$

where the stiffness matrix $[K]$ and the right-hand side $\{b\}$ are expressed as

$$\begin{aligned} K_{ij} = \iiint_{\Omega} \left[\frac{1}{\mu_r} (\nabla \times \vec{N}_i) \cdot (\nabla \times \vec{N}_j) - k_0^{\,2} \varepsilon_r \vec{N}_i \cdot \vec{N}_j \right] dV + \oiint_{S_{\text{ABC,port}}} jk_0 (\hat{n} \times \vec{N}_i) \cdot (\hat{n} \times \vec{N}_j) dS_{\text{ABC,port}} \\ i, j = 1, \ 2, \ ..., \ N, \end{aligned} \tag{2.8}$$

$$b_i = \iint_{S_{\text{port}}} \vec{N}_i \cdot \vec{U}^{\text{inc}} dS_{\text{port}} , \quad i = 1, \ 2, \ \ldots, N \ . \tag{2.9}$$

To excite the waveguide-port, the TEM modal function $\vec{e}_0$ for a coaxial waveguide is used in the formulation, rendering

$$b_i = j2k_0 \iint_{S_{\text{port}}} \vec{N}_i \cdot \vec{e}_0 dS_{\text{port}} \tag{2.10}$$

where

$$\vec{e}_0 = \hat{\rho} \frac{1}{\sqrt{2\pi \ln(b/a)}} \frac{1}{\rho} e^{-jk_0 z} \tag{2.11}$$

in which a and b are the inner and the outer radii of the coaxial port.

The finite-element algorithm implementation begins with the assembly of the stiffness and mass matrices of the discretized elements in the computational domain, along with the incorporation of the boundary conditions. These will be

discussed in Section 4. The solution to the system of equations can then be obtained using either direct or iterative solvers, which will be discussed in Section 5.

3. GPU Architecture

This section introduces NVIDIA's GPU hardware architecture as well as its CUDA software architecture. For the hardware, this research uses NVIDIA's GeForce GTX480 of Compute Capability 2.0, with 1.5GB of memory. For the CUDA software, this research uses CUDA v.4.1. The hardware architecture and the memory distribution of CUDA are shown in Figure 3.1. A GPU, denoted as the device, contains hundreds of processors which execute the same command simultaneously. Each GPU contains a number of streaming Multiprocessors (MP), each of which contains several Scalar Processors (SP) executing in groups. The SPs are the fine-grained processing cores of the GPU, enabling the GPU to carry out simultaneous computations on its many-cores, in contrast to the multi-core architecture of the CPU, in which only a few processing cores are present. Each SP has its own local registers. The SPs in each MP are capable of accessing a common shared-memory residing in the MP hardware architecture. Above that, all SPs in all MPs can have access to the global memory residing off the GPU chip. A separate, cached constant memory is also accessible to every SP. From the hardware aspect, the acceleration capability of the GPU comes from the apparent parallel architecture of its numerous processors.

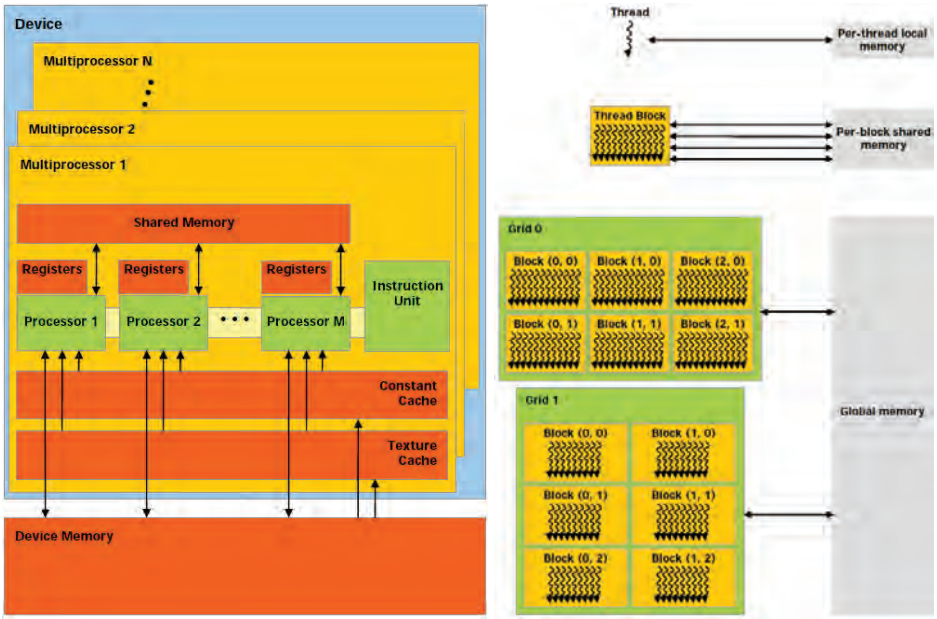


Figure 3.1. NVIDIA GPU a) hardware and b) software architecture[2]

CUDA[2] is the software programming interface of NVIDIA's GPU. The flow of CUDA is as follows: once the CPU, denoted as the host, reaches the parallelizable portion of the program flow, the data on the host memory is sent to the different memory locations in the device. Next, CUDA will invoke a 'kernel,' which calls the interface function to initialize and allocate the physical MPs and SPs. The physical MPs and SPs of the device hardware are then mapped by CUDA into controllable software elements denoted as the 'blocks' and the 'threads.' The blocks and threads in a kernel can be controlled by their unique indices for parallel computation. At the end of each kernel, the computed result can then be transferred back from the device memory to the host memory, thus completing the parallel portion of the program.

Corresponding to the hardware architecture, a thread carries its own local registers and the threads in the same block share the same shared memory. These two memories have very-low-latency, ranging from one to a few clock-cycles. On the other hand, the device global memory that is accessible to all blocks and threads has very high-latency, around several hundred clock-cycles. This high-latency can be hidden with coalesced global memory access by using the concept of warps. A warp contains 32 threads which physically execute on the same multiprocessor together at a given time. If the threads in the same warp access a consecutive segment from the global memory, then the 32 memory accesses will be translated into a single fetch instruction. This effectively reduces the memory access delay by 32 times. This is yet another important acceleration capability of the GPU, and speedup is achieved by using coalesced memory-access to reduce memory latency.

To modify an existing finite-element code for acceleration on a GPU, many existing storage structures need to be broken down into smaller substructures. In general, the common array-of-structures type of class format is not suitable for a GPU computation. As stated above, besides the apparent many-core parallelization, one of the important advantages of a GPU is the coalesced-memory-access pattern. If data are accessed from a single-array with large structures, then the data accessed will not be consecutive and thus will not be coalesced. On the other hand, if each thread in the same warp can access a small amount of data that are consecutive in the global memory, then memory access is coalesced. This is the preferable structure-of-arrays type of memory storage and accessibility for GPU computation, in which arrays of different but relevant data are transferred to the device memory separately. Each warp can then access the memory from this agglomeration of arrays in a coalesced manner.

Along this line, CUDA is, by nature, a C language, but it supports C++ to a certain extent. Specifically, structures and classes can be written to be host/device transferrable. The same set of data and functions within a class can be used both on a CPU and a GPU, if written correctly. However, the CUDA compiler currently does not support linking for device functions. That is, all of the CUDA-related portion of the code has to be compiled into one single-object file. As a result, existing libraries have to be modified to suit the GPU. This is another reason why splitting existing data structures are absolutely necessary.

4. GPU Implementation - Assembly

The computation of the finite-element assembly process is discussed in detail in this section, as well as all of the data structures involved in the assembly process. It will be shown that the existing assembly process is not suitable for GPU computation, and that modification of the data structures is needed. This section will lay out the finite-element structures, point out the difficulty in their direct application on GPU, and propose the modification needed in a minimally-invasive way.

The finite-element program begins by reading in the mesh file into some structure type, e.g., a mesh class in the C++ language. This will be the class of geometric information from which other classes will access to carry out the assembly process. Usually, other relevant geometric information will be deduced at this time for the generalization and extensibility of the program. The deduced geometric information will also be stored in the mesh class, but much of it is not needed for the antenna analysis. Since memory transfer needs to be kept at a minimum, and coalescing is hard to achieve with bulky class structures, ultimately the mesh information will need to be split for GPU adaptation. However, the mesh reading process is inherently serial, and thus is best kept on the CPU.

The next step of the finite-element assembly process is the computation and assembly of the stiffness and mass matrices of each three-dimensional (3D) element. Here tetrahedrons are used. Traditionally, the CPU program loops over each tetrahedral element, computes an elemental tetrahedral matrix, then assembles the element matrices into the final FEM matrix. This process uses the tetrahedron class, which allocates and stores all relevant information on the tetrahedral elements, ranging from the given mesh data to the computed intermediate values. One apparent parallelization scheme is to parallelize the computation of the element matrices, with one GPU thread calculating one element.

However, as the GPU global memory is usually much smaller than the CPU memory, it is impractical to duplicate the entire CPU tetrahedron class structure onto the GPU. Moreover, the transfer rate between the CPU and the GPU memories is slow, and this poses another bottleneck for any meaningful acceleration if the entire class is to be transferred. The methodology for writing the GPU code is then to first identify which data in the CPU tetrahedron class is absolutely necessary for the computation of the elemental matrices. Then, relevant data from the CPU tetrahedron class can be transferred into the GPU memory, while the other intermediate information can be deduced in the GPU kernels. The transferred data takes about only 10% of the original memory usage, therefore greatly reducing the memory transfer-time.

With this in mind, notice that the governing volume integral for the tetrahedral matrices in Equation (2.8) only calls for the information of the nodes, namely, the corresponding global nodal numbers, their coordinates, and the corresponding global edge numbers in a tetrahedron. Therefore, we can isolate the global node and edge indices of each tetrahedral element into a host/device transferable class, as shown in Figure 4.1. The global indices can then be used to fetch information from the mesh data arrays. For the calculated intermediate data, such as the volume or the basis of the tetrahedron, they can be packaged into other transferable classes. These intermediate data are stored in the shared memory or local registers for quick access in the same kernel. Since only some of them will be reused for later kernels, it is more efficient to re-compute them than to port all of them onto the global memory.

Notice that small global constants such as the frequencies and the quadrature integration table can be stored in the global constant memory for cached access. The generated stiffness and mass matrices are stored in the global memory, waiting to be assembled in a later step.

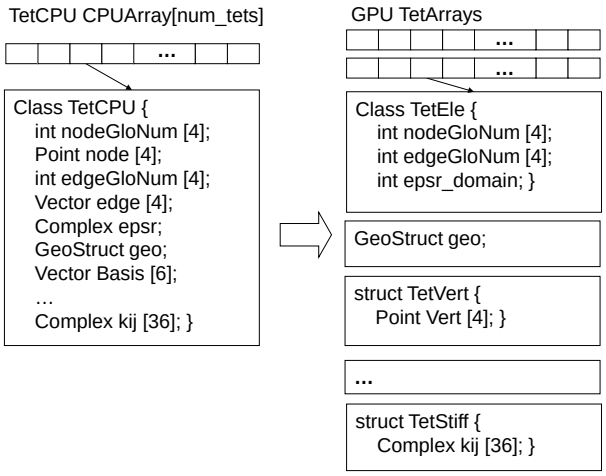


Figure 4.1. Splitting of the CPU tetrahedron class and the creation of the GPU tetrahedron classes and structures

The next step of the finite-element assembly process is the computation and the assembly of the two-dimensional (2D) boundary conditions. Here, the 2D elements are the triangular faces of the outermost boundary tetrahedral elements of the computational domain. Integrations on the surfaces are numerically turned into elemental triangular matrices and the right-hand-side of the final FEM matrix. Similar to the tetrahedral elements, the original CPU program loops over each boundary triangular element, computes a triangular element matrix, then assembles the element matrices into the system of equations. For antenna analysis, both the absorbing boundary condition and the waveguide-port condition are needed. Although a different equation governs each of the two boundary conditions, the assembly steps of the element matrices remain the same as the assembly of the tetrahedral matrices. Thus, while the computation of the different boundary conditions has to remain serial, the computation and the assembly of each element matrix can still be parallelized.

Similar to the tetrahedron class, the triangle class also contains all kinds of unused data and thus needs to be split, and new host/device transferable classes need to be created. Notice that the governing equation for the triangular element matrices only requires the information of the residing tetrahedron, along with the local boundary face number. Therefore, for each boundary face, the only additional information needed is the relevant global tetrahedral element number and the corresponding local boundary face number. This information is then grouped into a new transferable class and copied into the device memory, as shown in Figure 4.2. Each kernel then computes one boundary condition and the threads in each kernel compute in parallel the elemental matrix and the right-hand-side, which are stored in separate structure arrays in the global memory, waiting to be assembled in the next step.

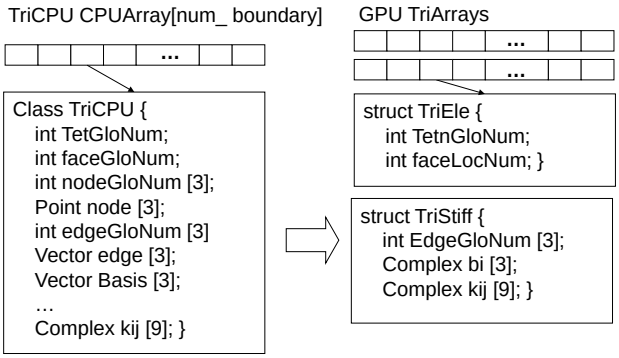


Figure 4.2. Splitting of the CPU triangle class and the creation of the GPU triangle classes

The final step of the finite-element assembly process is the actual assembly of the 2D and 3D element matrices and the right-hand-side. The traditional assembly process involves first looping over all elements, then identifies and pre-allocates all the non-zero locations of the FEM matrix to obtain the sparsity pattern in the compressed-sparse-row (CSR) format. The program then again loops through all the elements, but this time fills in the actual non-zero elements in their proper location. This is done by looping through the non-zero indices of the target row and puts the values of each of the element matrices in the corresponding CSR location.

At this point, the GPU has parallelized the computation of the element matrices and stored them in the device (GPU) global memory. However, assembly using the same simple parallelization scheme over all finite-elements involves potential race conditions, since most of the non-zeros in the final system of equations are contributed by more than one element, and the contributions cannot be simultaneously assembled to the same memory location. This scheme can thus dramatically reduce GPU acceleration. Also, since the FEM matrix is stored in a sparse matrix format (CSR), the target location of each non-zero matrix element has to be predetermined serially, which is not suitable for a GPU. To overcome this obstacle, a novel assembly method needs to be used.

Each row of the FEM matrix consists of interactions between an edge with all of its adjacent edges. One apparent parallelization scheme from this point of view is then to parallelize the assembly-by-edge interactions (instead of by elements). If the number of non-zeros in each row can be pre-determined, then we can use one thread to compute each row. Specifically, each thread handles one edge-to-edges correlation in the adjacent elements of the target edge, which consists of all the interactions between a single edge and all of its neighboring edges.

To proceed with assembly by edge, we first construct an edge-to-element table in the edge-class during the edge construction phase. After transferring the table to the device memory, an assembly kernel is invoked, and each thread then loops over an edge's adjacent elements to obtain its adjacent edge numbers and their corresponding non-zero values. The rows are first assembled in the shared-memory for fast memory access, and then are stored in the global memory to form the final CSR FEM matrix according to the pre-determined starting location of each row.

Similarly for the assembly of the surface boundary conditions, notice that each boundary edge is adjacent to precisely two triangular boundary elements. The edge-to-face table can also be pre-computed on the CPU and be incorporated into the edge-class, with an extra indicator in the edge-class indicating the type of boundary condition of the edge. The layout of the edge-class is shown in Figure 4.3. The parallelized edges then assemble the boundary matrices and the right-hand side array to the FEM system of equations in the same assembly kernel.

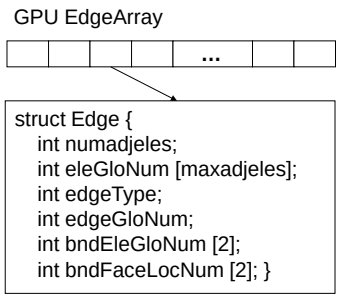


Figure 4.3. Creation of the GPU edge-classes

The flow of the finite-element program on a GPU is then as follows: The mesh information is first stored in arrays of coordinates, connectivity, and boundary conditions, etc. The CPU will still be used to form edge structures and element classes. After all the relevant data are transferred onto the device global memory, kernels are invoked in sequence to compute the sparsity pattern, tetrahedral element matrices, and triangular element matrices, and store them in the global memory. The assembly kernel is then invoked to parallelize the assembly by rows. Each assembled row is temporarily stored in the shared-memory before exporting onto the global memory. At this point the assembly of the FEM matrix and the right-hand-side is completed. Figure 4.4 illustrates the program flow.

```
Void FEMAssembly {
   // Preprocessing on CPU
   CPU.readmesh(h_nodeArr, h_tetArr, h_boundArr, …);
   CPU.makeEdges(h_nodeArr, h_tetArr, h_edgeArr, …);
   // Initialize GPU
   GPU.initializeCUDA(d_nodeArr, d_tetArr, d_edgeArr, …);
   GPU.memoryTransfer(h_nodeArr->d_nodeArr, …);
   // Element matrices computation on GPU
   GPU.KernelTetMat(d_nodeArr, d_tetArr, …, d_tetMarArr);
   GPU.KernelTriMat(d_nodeArr, d_triArr, …, d_triMatArr);
   // Element matrices assembly on GPU
   GPU.KernelAssem(d_edgeArr, d_tetMarArr, d_triMatArr, …);
}
```

Figure 4.4. Pseudocodes for finite-element assembly on GPU

5. GPU Implementation – Solution

This section discusses the solution process of the finite-element antenna analysis. The BiCGStab iterative method is used for both the host and the device codes. The SpMV and the BLAS libraries in the Intel® Math Kernel Library with OpenMP capability are used in the host code to perform sparse matrix-vector products (spmv) and vector-vector calculations such as scalar multiplications and dot products. As for the device code, CuSparse and CuBLAS library functions are used. For the sparse matrix-vector product on the device, it was discovered that the Jagged Diagonal (JAD) format provides a more coalesced memory access by the device threads, and thus reduces the spmv time in some matrices. The speedup of the JAD spmv is compared to that of the CSR format using the CuSparse library. Table 5.1 below shows the spmv speedup in a scattering by a sphere problem:

Table 5.1. Comparison of CSR and JAD SpMV timing

Frequency	Unknowns	CuSparse (s)	Iterations	JAD (s)	Iterations	Speedup
300 MHz	172,067	6.35	1,450	5.10	1,523	1.25
500 MHz	720,051	29.12	1,793	25.63	1,664	1.14
800 MHz	1,605,614	81.11	2,422	78.79	2,291	1.03

The JAD format indeed outperforms the CSR format, although the speedup is more apparent for smaller matrices. This is because the CSR format has intrinsic load-balancing disadvantages due to the highly-uneven number of non-zeros in each matrix row. The JAD format rearranges the rows and columns to balance the load. It is unsure whether the speedup can be further improved, since the JAD format intrinsically loses its formatting efficiency as the number of non-zero elements in the matrix increases. In addition, it is not easy to directly interpret the JAD matrix to construct pre-conditioners for an iterative solver. Also, the speedup is very limited. Due to the above reasons, the CSR format is still preferable over the JAD format for practical purposes.

On the pre-conditioner side, searching for a good pre-conditioner for the problem is essential to decreasing the number of iteration steps or even to achieving convergence. Two criteria in the selection of a suitable pre-conditioner on a GPU are proposed. First, construction of the pre-conditioner needs to be parallelizable in order to be performed on a GPU. Second, the application of the pre-conditioner in the iterative solver needs to be parallelized in order to achieve speedup. The most obvious choice is then the Jacobi (diagonal scaling) pre-conditioner, in which the construction and application processes are both inherently parallel. However, the Jacobi pre-conditioner does not provide sufficiently good pre-conditioning. Sometimes it does not help with achieving convergence at all, which is especially true for antenna analysis. On the other hand, the construction process of good pre-conditioners such as the ILU pre-conditioner is inherently serial, so this is not desirable either. The SSOR pre-conditioner is parallelizable and cheap, but its application in the iterative solver involves backward and forward substitutions, which contain very few parallelizable processes, despite their availability in the CuSparse library. As shown in Table 5.2 for a sphere scattering problem, although the SSOR pre-conditioner successfully reduced the number of iteration steps by a factor of 4 compared to the Jacobi pre-conditioner, the actual speedup of the SSOR pre-conditioner on the GPU was actually less than the simple Jacobi pre-conditioner.

Table 5.2. Comparison of Jacobi and SSOR pre-conditioners

Pre-conditioner	CPU (s)	Iterations	GPU (s)	Iterations	Speedup
Jacobi	59.31	1,369	6.35	1,545	9.34
SSOR	33.70	346	9.22	400	3.66

Through investigation, it was found that the family of approximate inverse (AI) pre-conditioners matches our criteria. These pre-conditioners use mathematical approximations to invert the pre-conditioning matrix and, therefore, their incorporation in the iterative solver involves only additional sparse matrix-vector products instead of the time-consuming backward and forward substitutions. These pre-conditioners are potential candidates to further speedup the solution time. The SSOR-AI pre-conditioner is thus chosen due to the fact that its construction process is also inherently parallel[3]. Furthermore, with the incorporation of the Jacobi diagonal scaling before applying the SSOR-AI pre-conditioner, the number of iterations can be further reduced. The number of iteration steps can also be reduced through a careful selection of the relaxation parameter ω. The relaxation parameter ω was swept for $0<\omega<2$, and the valley points were picked as the optimum value for ω. This process is done on the scattering of a 1m PEC cube, and the result is shown in Figure 5.1.

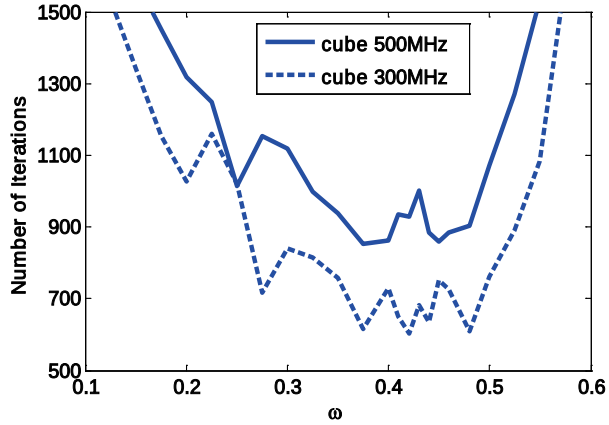


Figure 5.1. Number of iterations with respect to the relaxation parameter ω

The figure shows ω in the range $0.1 < \omega < 0.6$. Other values are either way off from the valley or do not produce a converged solution. It can be seen that the valley is somewhere around ω=0.4, and that it is invariant to the size of the problem. Similar sweeping was done with a monopole and a Vivaldi antenna for the antenna analysis, and the valley is again found to be around ω=0.4. Since it is difficult to numerically estimate ω, the experimental result obtained here will be used for other cases. All of the following SSOR-AI-related results use ω=0.42.

6. Examples and Speedups

The first example is a monopole on top of an infinite PEC plate, where h=1m, a=0.1m, and b=0.23m. For simplicity, the relative permittivity of the waveguide-port is set to ε_r=1. A first-order spherical ABC boundary is set at an appropriate distance away from the monopole. Table 6.1 shows the effect of the pre-conditioners and the GPU speedup of the monopole problem with 436,920 unknowns at a frequency of 75MHz. The timing comparison to an OpenMP-enabled Xeon W3520 quad-core CPU is also shown in Table 6.1. The geometry and the resulting radiation pattern of the monopole antenna are shown in Figure 6.1.

Table 6.1. 75MHz radiation of a monopole antenna with 436,920 unknowns

Pre-conditioner	CPU 4c. (s)	Iterations	GPU (s)	Iterations	Speedup
Jacobi	655.79	11,878	94.07	10,552	6.97
SSOR-AI	405.95	3,853	57.77	3,439	7.03
SSOR-AI + Jacobi	402.84	3,823	55.74	3,314	7.23

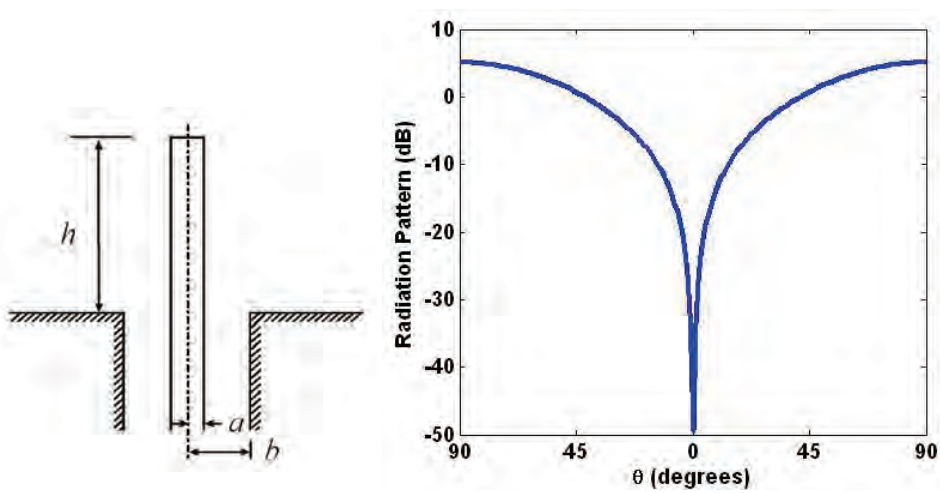


Figure 6.1. Geometry and radiation pattern of the monopole antenna at 75MHz

The second example is a Vivaldi antenna with h=33.3mm, w=34mm, and d=1.27mm. The radius of the hollow circle is R=2.5mm. The taper is an exponential function with $w(z)=0.25e^{0.123z}$mm. The coaxial feed has an inner- and outer-radii of 0.375mm and 0.875mm with a relative permittivity of $\varepsilon_{r,\text{coax}}$=1.0. The dielectric substrate has a relative permittivity of $\varepsilon_{r,\text{dielectric}}$=6.0. The timing comparison to an OpenMP-enabled quad-core CPU is shown in Table 6.2, while the geometry and resulting radiation pattern of the Vivaldi antenna operating at 3GHz is shown in Figure 6.2.

Table 6.2 3GHz radiation of a Vivaldi antenna with 1,015,550 unknowns

Pre-conditioner	CPU 4c. (s)	Iterations	GPU (s)	Iterations	Speedup
Jacobi	N/A	Failed	N/A	Failed	N/A
SSOR-AI	7,901.83	23,915	999.43	22,199	7.91
SSOR-AI +Jacobi	2,659.64	7,764	370.74	8,216	7.17

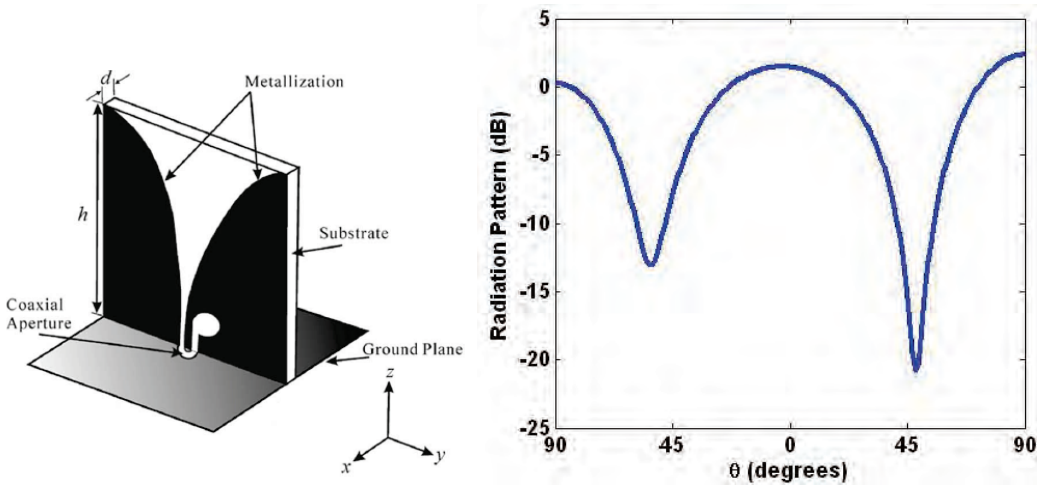


Figure 6.2. Geometry and radiation pattern of the Vivaldi antenna at 3GHz

While the application of the simple Jacobi pre-conditioner failed to obtain a converged solution in this case, the application of the SSOR-AI pre-conditioners managed to converge. Furthermore, by pre-applying the Jacobi diagonal scaling, the SSOR-AI pre-conditioner achieved reduction in the number of iterations. The GPU further accelerated the total solution time by approximately 7x speedup when compared to the quad-core CPU parallelization.

7. Conclusion

Bottlenecks in the GPU parallelization of the finite-element algorithm for 3D antenna analysis are identified in this paper. A possible parallelization scheme on a GPU is proposed, which involves minimal modifications to the existing finite-element program structures. Existing CUDA libraries are used for the iterative solver. It is found that although the JAD matrix storage format outperforms the standard CSR storage format, it is not very practical to implement due to the difficulty of interpreting the JAD matrix to obtain pre-conditioners, which are essential to computation time reduction, or even convergence for the antenna analysis. It is shown that the SSOR-AI pre-conditioner is a very suitable candidate for GPU acceleration, although its relaxation parameter needs to be carefully chosen. Our study finds that for a standard finite-element algorithm, the solution time of the problems greatly exceeds that of the assembly time. But through a proper implementation, the overall computation can be accelerated, while the memory usage can be kept to a minimum.

Acknowledgment

The authors gratefully acknowledge the funding support for this project from the DoD High Performance Computing Modernization Program's User Productivity, Enhancement, Technology Transfer and Training (PETTT) initiative.

References

1. Jin, J.-M. and Riley, D.J., *Finite Element Analysis of Antennas and Arrays*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2008.

2. NVIDIA. *CUDA C Programming Guide*, retrieved from http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf.

3. Helfenstein, R. and Koko, J. (in press), "Parallel preconditioned conjugate gradient algorithm on GPU." *Journal of Computational and Applied Mathematics*.

PETTT Signal and Image Processing CTA

Juan Carlos Chaves, Jose Unpingco, Bracy Elton, and Peter Raeth
High Performance Technologies, Inc. (HPTi), a DRC company
{jchaves, junpingco, belton, praeth}@drc.mil

Abstract

The Signal and Image Processing (SIP) computational technology area (CTA) covers the extraction of useful information from a variety of sensor outputs often, in real-time. Nowadays, signals and images are flowing from numerous types of intelligent devices, sensors, and objects. In addition, Department of Defense (DoD) organizations are experiencing explosive growth of unstructured content. All kinds of unstructured information from audio, images, and text records that demand urgent analysis are being authored by DoD scientists and engineers. Indeed, the SIP CTA involves ever-increasing data-intensive applications. Making sense of the enormity of data requires innovative HPC solutions. Developing those solutions is a major SIP CTA challenge. Since the data sources and types of data are ever-changing, each situation may require its own unique solution. Therefore, rapid prototyping is an important aspect of the SIP CTA and the User Productivity, Enhancement, Technology Transfer and Training (PETTT) SIP team has extensive knowledge and experience in high-productivity languages (e.g., MATLAB and Python) and its parallel extensions. DoD applications include: surveillance, reconnaissance, intelligence, communications, avionics, smart munitions, and electronic warfare. Sensor types include sonar, radar, visible and infrared imagery, and signal intelligence (SIGINT) and navigation assets. Typical signal processing functions include detecting, tracking, classifying, and recognizing targets in the midst of noise and jamming. Image processing functions include the generation of high-resolution low-noise imagery and the compression of imagery for communications and storage. In addition, data mining and text analytics techniques are methods used to find patterns in, or otherwise interpret, huge repositories of unstructured information. In this short paper, we highlight important activities of the PETTT SIP team that are having a significant impact on DoD applications for the benefit of the Warfighter.

1. Introduction

In this short paper, we highlight some of the important activities of the User Productivity, Enhancement, Technology Transfer and Training (PETTT) Signal and Image Processing (SIP) team that are having a significant impact on Department of Defense (DoD) applications for the benefit of the Warfighter. The main areas in which SIP computational technology area (CTA) has a major DoD impact are: actionable intelligence, pervasive, all-weather, all-hour surveillance, real-time battle space awareness and reduced casualties. In addition, the SIP key value-add of PETTT/HPC is refining codes, reducing time-to-solution, adapting to continually changing threats and generating exceptional creative solutions to DoD problems. For example, the new emphasis of high-end computing systems is rapidly evolving towards productivity and value, rather than traditional HPC standards such as raw theoretical peak computing performance. Researchers' idea-to-solution or time-to-solution is becoming a more important metric than old fashioned raw computing capacity. Moreover, total end-user computing life-cycle costs and mission responsiveness are becoming increasingly critical to operational scenarios of modern DoD and homeland defense systems. Many SIP users instead prefer high-productivity languages within integrated development environments, such as MATLAB and Python. Therefore, an important activity of the SIP team is high-productivity languages. Another important activity of the SIP team is collaborating and contributing to the advance of Intelligence, Surveillance and Reconnaissance (ISR) applications.

In Section 2, we present a list of the ongoing SIP team activities. In Section 3, we list the past and current PETTT SIP pre-planned and special projects. In Section 4, we present a short overview of the SIP trends and challenges. Finally, we conclude by briefly mentioning our emerging activities and opportunities.

2. Ongoing Activities

- High-level/High-productivity languages and its parallel extensions: MATLAB, Python, Octave, MatlabMPI, bcMPI, PCT/DCS, IPython, etc. (See Figure 1)
- OpenUTF: common, scalable, open-source, open-development Modeling & Simulation framework.
- Intelligence, Surveillance and Reconnaissance (ISR) applications: SAR, ATR, GMTI, STAP, etc. (See Figure 2)
- Distributed matrices applications: MATLAB PCT/DCS, PETSc, and Global Arrays Toolkit.
- GPGPU Computing: C/CUDA, MATLAB, and Python. (See Figure 3)
- Computer Forensics and Text Analytics. (See Figure 4)

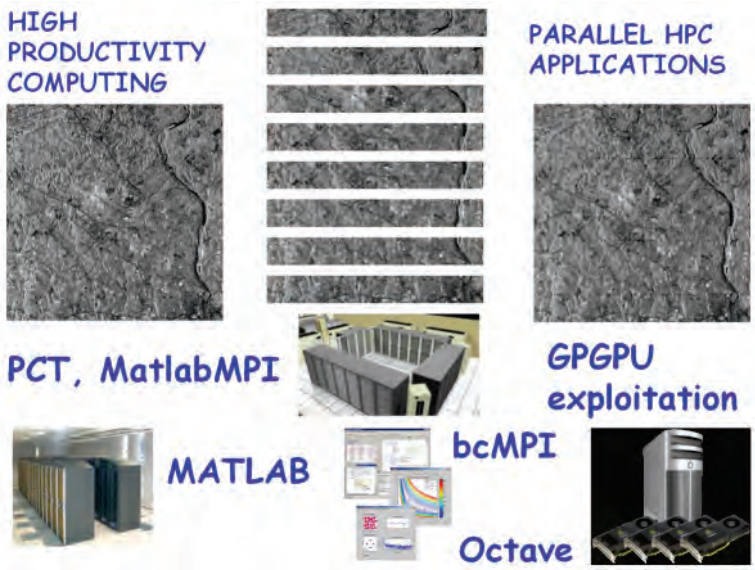


Figure 1. High-productivity Languages

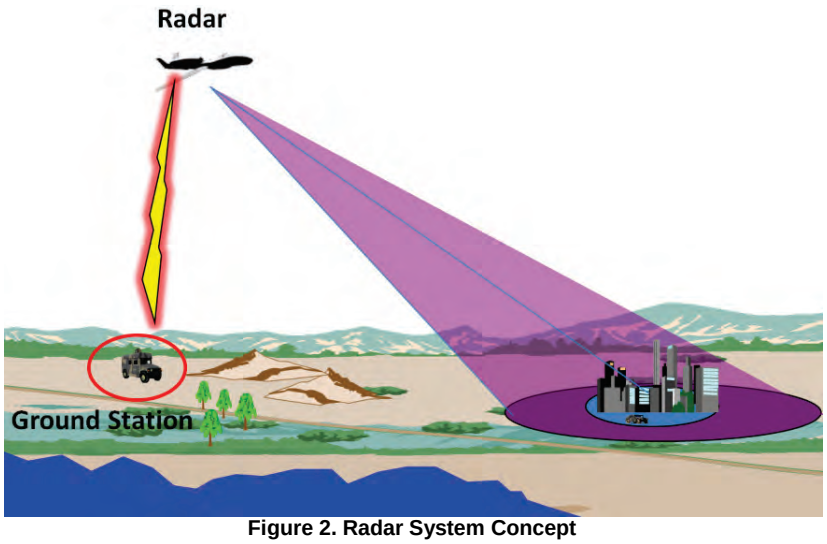


Figure 2. Radar System Concept

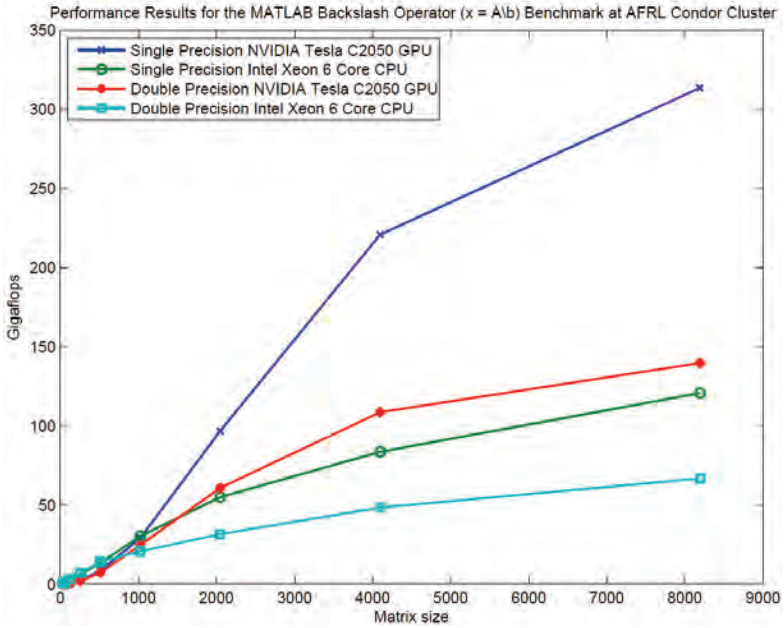


Figure 3. GPU Computing with MATLAB

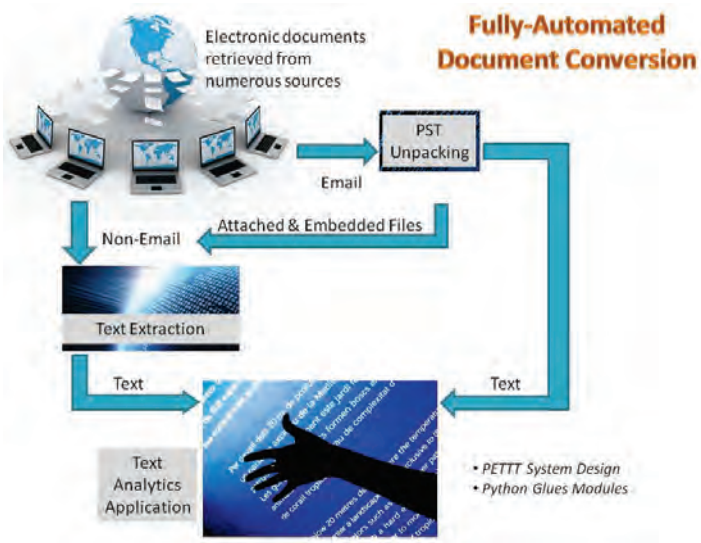


Figure 4. Computer Forensics and Text Analytics

3. Pre-planned and Special Projects

- Modeling and Simulation of Large Superconducting Quantum Interference Devices (SQUID arrays) (KY01 and KY02). (See Figure 5)
- Interactive and Web-Enabled HPC with IPython (KY03). (See Figure 6)
- License-Free MATLAB codes via Python (KY04).
- Modeling and Simulation of Large Superconducting Quantum Interference Devices (SQUIDs) on Graphics Processing Units (GPUs) (KY04).
- Assessment of Double-Precision GPUs as Computational Platforms for Gotcha Radar Exploitation Systems (KY02).

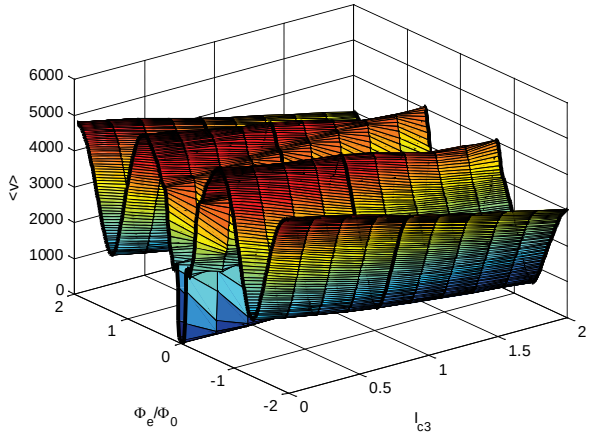


Figure 5: Average voltage response for a non-uniform series bi-SQUID array with 6,000 bi-SQUIDs

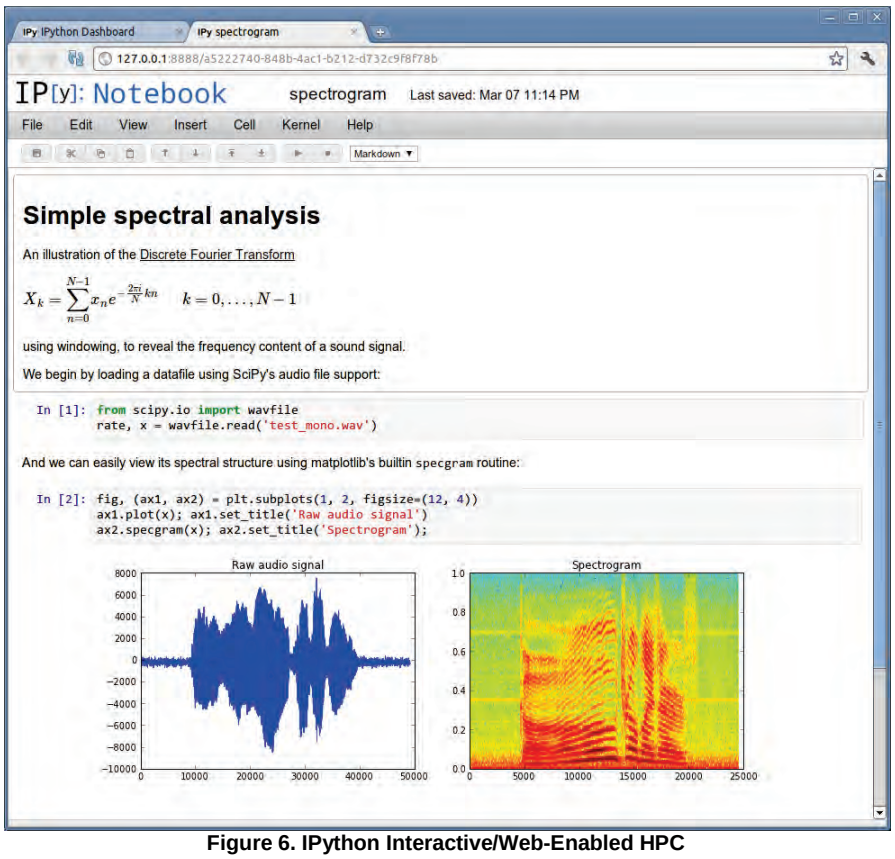


Figure 6. IPython Interactive/Web-Enabled HPC

4. SIP CTA Trends and Challenges

- The inherent complexity in utilizing and programming HPC systems is the main obstacle to widespread exploitation of HPC resources and technologies in SIP. The challenge is to fulfill the persistent need to simplify the programming interface for the generic SIP user.
- Deployable, persistent, real-time, all-hour, all-weather surveillance systems with forensic and target tracking capabilities. The challenge is twofold: 1) substantially scale the algorithms, and 2) effectively exploit the latest HPC technologies to reach the required resolutions that will fulfill the future DoD surveillance requirements.

- A data explosion is coming upon the DoD. As sensors become cheaper, smaller, lighter, consume less power, become more capable and are connected to global defense networks (including the public Internet), a data tsunami is apparent. Making sense of this data is a really tough problem, and it is one of the main SIP challenges.

5. Conclusion

In this short paper, we gave a brief overview of the PETTT SIP main activities. In addition, we are planning to expand our efforts into important emerging activities such as Compressive Sensing and Natural Language Processing that promise to significantly enhance the DoD capabilities in actionable intelligence, pervasive, all-weather, all-hour surveillance, real-time battle space awareness and reduced casualties.

Portal Development for High Performance Computing (HPC) at Maui High Performance Computing Center (MHPCC)

David Morton
Maui High Performance Computing Center (MHPCC), Maui, HI
david.morton@maui.afmc.af.mil

Abstract

The goal of this program is to transform Department of Defense (DoD) high performance computing (HPC) application delivery, for selected applications, from stand-alone to cloud-based. The benefits of this program to the end application user are manifold: a) eliminates traditional HPC application stove-pipes making application and data sharable world-wide on-demand; b) no installed software required eliminating user configuration and maintenance; c) simplified and enhanced security model d) does not require HPC knowledge or traditional hurdles to fully exploit HPC; and e) provides immediate access to HPC resources using fieldable, low-power, portable devices such as the iPad. Two stand-alone software product areas were specified for initial deployment:

- *Computational Research & Engineering Acquisition Tools and Environments (CREATE) Kestrel*
- *Distributed MATLAB*

Figure 1shows the current application suite start-up panel which includes Kestrel, CREATE Job Manager, Distributed MATLAB, and Virtual Applications. The application delivery technology is described in a subsequent section. For example, virtual applications provide a quick-start capability for existing or commercial-off-the-shelf (COTS) programs, and uses a VNC-based technology referenced as Level 0.

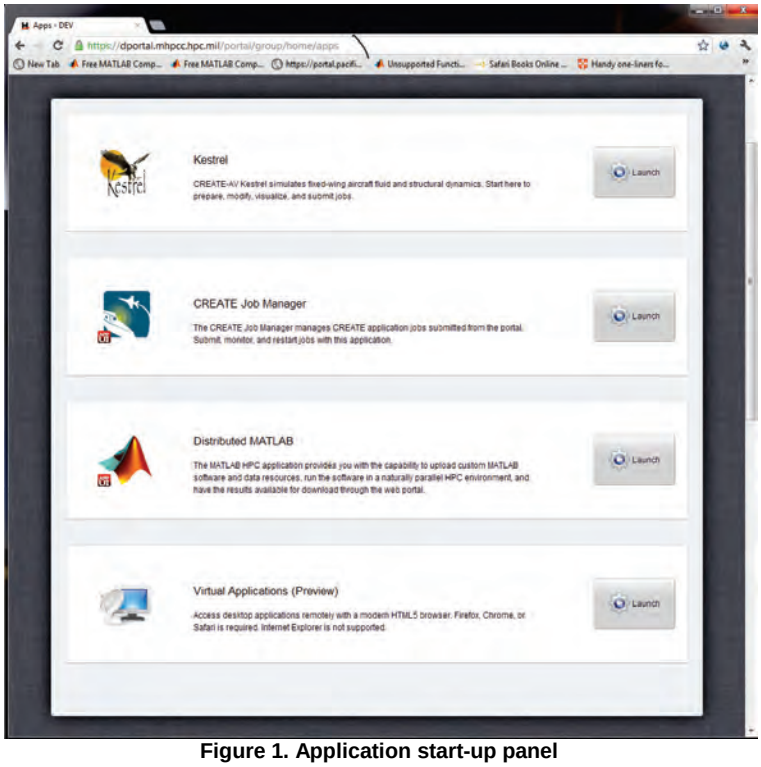


Figure 1. Application start-up panel

1. Introduction

In November 2011, these products were successfully demonstrated live at the 2011 NDIA Physics-based Modeling in Design & Development for US Defense Conference in Denver, CO, with the web-server and compute resources running at Maui High Performance Computing Center (MHPCC) in Maui, HI.

An MHPCC team of consisting of University of Hawaii (UH) and Pacific Defense Solutions (PDS) developers was assembled in Q1 2011. This team has fashioned key alliances with the US Army Research Laboratory (ARL), the USACE Engineer Research and Development Center (ERDC), and the Computational Research and Engineering Acquisition Tools and Environments (CREATE). Initially, the team is focusing on the development of the necessary infrastructure. In Q1 2012, MHPCC began offering on-demand Portal-based services for Pioneer users for distributed MATLAB and CREATE Kestrel. Development will continue on other CREATE applications including DaVinci and Helios beginning in Q2 2012.

A Portal Software Development Kit (SDK) will be described that will be used as a force-multiplier to speed application transformation by both internal and external development teams. MHPCC is planning to implement a support infrastructure leveraging existing High Performance Computing Modernization Program (HPCMP) resources to train customers on the use of the Portal environment, provide customer assistance, and grow the user base for the Portal environment.

2. Long-Term Vision

While working to demonstrate specific application delivery of CREATE and MATLAB, the MHPCC team has adopted a longer-term vision to guide the near-term development making sure the portal design is applicable and scalable to other domains in the future. As networks and browsers continue to improve cloud-based application, delivery appears well-suited to the Department of Defense (DoD) and may become the standard delivery mechanism in the future. Immediate benefits of application delivery with portal include: a) no client software required on user's machine, b) eliminates all user configuration and maintenance, c) seamless collaboration within groups, d) high performance computing (HPC) scalability, e) delivery is OS and device independent, and f) simplified security model. Newer technologies are being incorporated into the portal today, such as openID and oAuth, which will facilitate secure access to the full power of HPC resources using low-power, non-traditional portable devices such as the iPad. These same technologies will allow applications to scale across DoD Supercomputing Resource Centers (DSRCs) without requiring user interaction.

This year we plan to demonstrate seamless access to compute resources between MHPCC and ERDC. We are working to provide a portal software development kit (SDK) that will get legacy applications up and running quickly but provide a common workflow and common application user interface (appUI) for the longer term or for new application development.

3. Application Delivery

Application Delivery Levels for the portal classify how existing or new applications will be delivered within the portal environment. A higher numerical level delivery generally implies more integrated web-based client-side architecture, while a lower-level implies a legacy (thick) client application configured for quick portal deployment. A key objective in portal development is to provide all application access, regardless of delivery level, through a standard web-browser. Applications delivered in this manner are considered to provide a "zero-footprint," in that there are no additional user-side applications or browser plug-ins required. Since most modern browsers support JavaScript and significant portions of the HTML-5 spec, only these intrinsic browser facilities are assumed. Examples of browsers with these capabilities include Internet Explorer 9 (IE9) and all recent versions of Chrome, Firefox, and Safari. We will also test applications with IE8 under the Windows XP operating system to support some existing legacy DoD computer systems.

Level 0 Application Delivery. Level 0 application delivery allows legacy applications which are not web-based to be quickly implemented within the portal environment. Such legacy applications may currently require considerable client-side software and associated configuration. In Level 0, the server side software is assumed unchanged. The client-side software is also unchanged, but is installed at the server site (e.g., at the HPC facility) using various options including:

- Installation on a Windows-based node (terminal server or single-user)
- Virtualized machine (Windows or Linux) running on a Linux server

For example, if the legacy application interface only operates in the windows environment, previously assumed to be running on the user's windows box, the first step in the Level 0 solution is to install the client code on an HPC node using the first option or the second option with the virtual machine instantiation script invoking the required windows OS. The second step to delivering the application in Level 0 is to provide a client viewer to the, now remote, client using zero-footprint HTML5 + JavaScript (AJAX) technologies (e.g., Guacamole). The goal of this effort is to provide an Information Assurance (IA)-approved mechanism to reliably and efficiently provide this display remotely to the user.

Portal currently provides a Level 0 solution using an enhanced version of Guacamole, which should allow nearly any application to be accessed from the portal with a very minimal development effort. In some cases, the actual user experience may suffer, depending on the complexity of the display output bandwidth, and/or the technology used in the initial client-side development. Even so, there will be immediate benefit to implementing a Level 0 solution including:

- No client software required on user's machine
- Eliminates all user configuration and maintenance
- Legacy CREATE applications up and running quickly

Figure 2 shows the virtual application menu, and the result of selecting FieldView, a non-web-based, commonly used commercial application, delivered via Portal to user's browser using the portal zero-footprint technology.

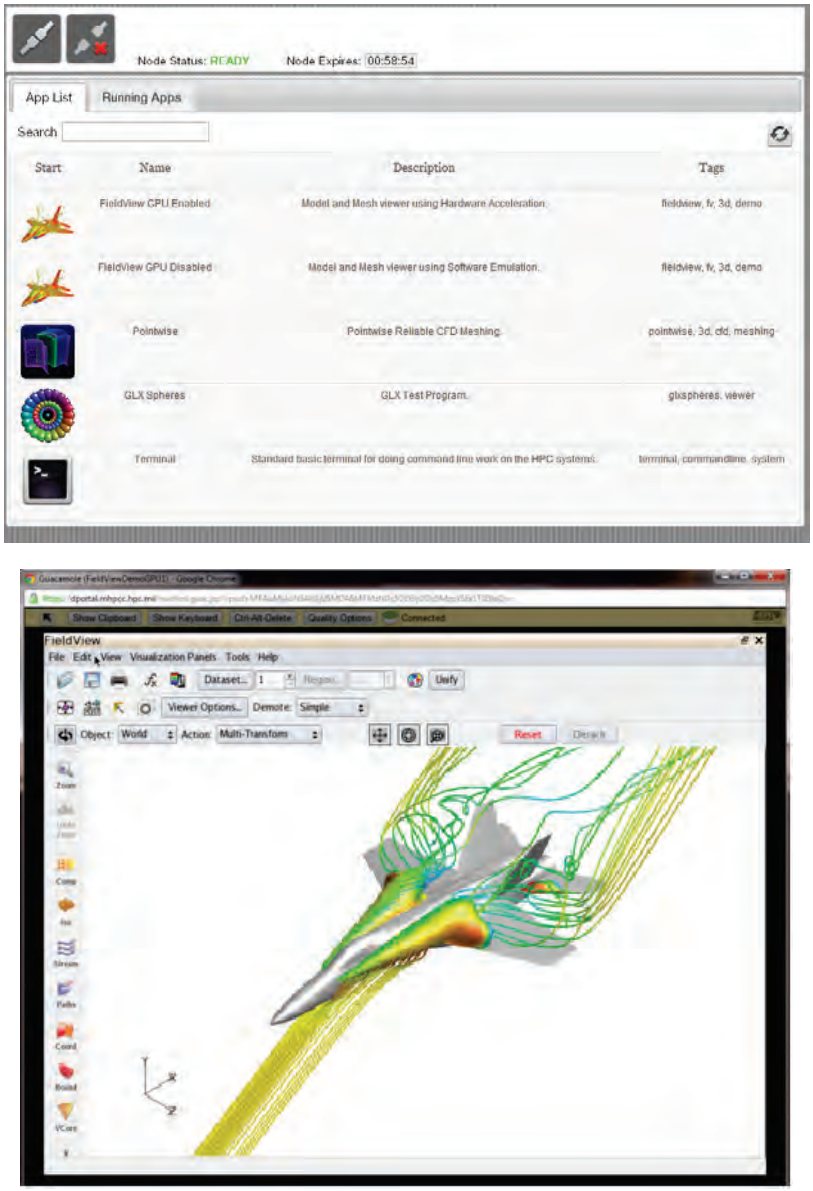


Figure 2. Virtual App Menu and FieldView delivered using zero-footprint Guacamole (Level 0)

Level 1 Application Delivery. Level 1 application delivery is considered a hybrid between a clean-slate, web-based application design and the Level 0 solution. For example, many legacy CREATE applications are based on sound design concepts, such as the model view controller (MVC) architecture, which separate the view logic from control logic. Instead of moving the entire client to the server-side using, for example, Level 0 virtualization, only the control logic needs to migrate to the server side. Serializable message-passing such as eXtensible Markup Language (XML) is typically used to communicate between the browser and the server application. Once this separation is complete, the Level 0 virtualization is unnecessary and its viewer is no longer needed, since the user interface components are implemented directly using AJAX technologies within the browser.

Level 2 Application Delivery. This represents a native HTML5/JavaScript web-based application. For future CREATE (other) applications intended to be integrated with the portal, the goal is to provide a bandwidth-aware software development kit (SDK) to assist with the development process. Since viewing large datasets is often integral to tasks that demand HPC, efficient technologies to support changing level of detail (e.g., mesh viewing for CREATE Air Vehicles (CREATE-AV) applications) are required. Examples of commercial products supporting efficient HTML5/JavaScript implementations are Google Maps, and Google Body. We are exploring existing open-source toolkits such as Visual Toolkit (VTK), and ParaViewWeb as a basis for developing a bandwidth-aware SDK. For legacy applications, the SDK will help facilitate a migration path from Level 0 or 1 to the preferred Level 2 delivery.

4. Portal Software Development Kit (SDK)

The portal SDK itself is being developed for the DoD. Developers share a common repository and the MHPCC portal team operates as source code change moderators. In this way, new versions of the SDK are compatible with prior versions, but may include enhancements and bug fixes supplied by outside teams using the SDK to get their applications incorporated into the portal. The SDK is still in development, but the first version is slated for release in Q3 of 2012. As shown in Figure 3, the SDK supplies a Portal FrameWork (back-end) for HPC resource access and user interface appUI (front-end) to provide a common look and feel interface to the portal application, while AppTop supports legacy and COTS applications. When developers use the SDK, end-users benefit from the commonality that the SDK provides in the user interface and resource management including files, job setup/status, quick-look capabilities, etc.

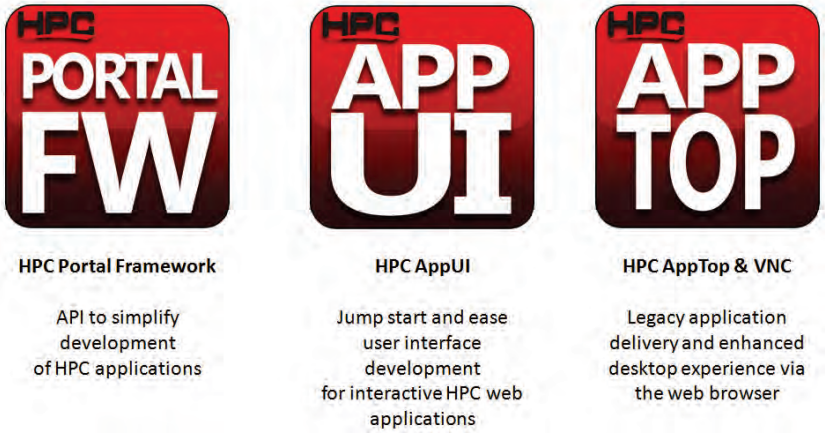


Figure 3. Portal Developer Tools

5. Open-Source Frameworks

To expedite implementation and facilitate future enhancements, a number of open-source Portal and content management frameworks were traded including Liferay, WSO2, Drupal, and Joomla. Example proof-of-concept test cases were implemented in each as part of the evaluation process to gauge effectiveness of leveraging such frameworks for the HPC portal implementation. Although the proof-of-concept test cases were successfully completed for all frameworks, the selection was narrowed to Liferay because the others had considerably less functionality. Liferay includes built-in content management, enterprise collaboration, and identity management (including OpenID), all of which made it a clear choice relative to the HPCMP Portal goals.

Independent developers using different frameworks successfully implemented a proof-of-concept HPC MATLAB objective to "drag-and-drop" a MATLAB script into a portal window for execution on a remote server, and then displaying the output back into the portal window. Figure 4 details the proof-of-concept test case as implemented in Liferay.

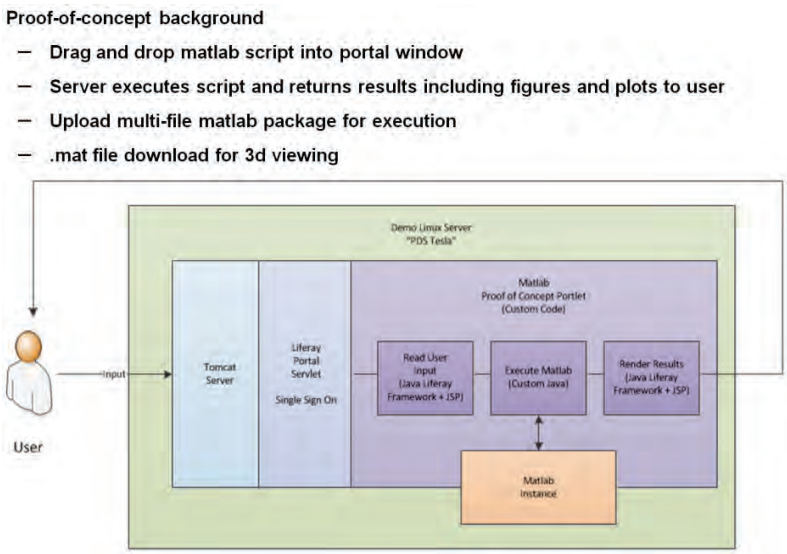


Figure 4. Liferay Proof-of-Concept Test Case

After selection of Liferay, the developers were challenged to extend this capability to include the upload and execution of an entire directory of user code and data. This proof-of-concept demonstration represents the use case in which the user-developed package could include an arbitrarily large number of functions (MATLAB .m files) and data files. The developers successfully implemented both the initial and extended proof-of-concept test cases in Liferay. An additional critical objective was demonstration of the implementation in a "zero-footprint," browser-only environment. Achieving this objective provides immediate flexibility to potentially deploy HPC capability to a wide-range of devices, independent of the underlying operating system (see Figure 5).

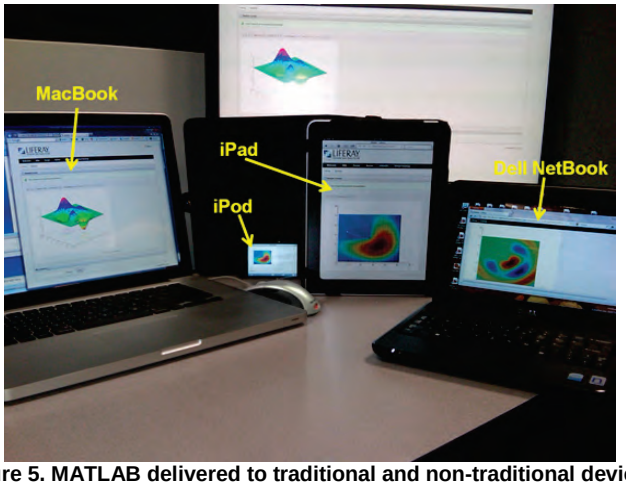


Figure 5. MATLAB delivered to traditional and non-traditional devices

6. Single Sign-on

Single sign-on (SSO) is a combination of identity management and authentication. As documented in the "*Enhancements to High Performance Computing Modernization Program (HPCMP) Authentication and Authorization Services*," memorandum of 5 April 2011, the HPCMP is updating existing access mechanisms, such as PKI (CAC & hToken) to be based on Open-Identity technologies (OpenID) and Open-Authentication (OAuth). In support of this new

initiative, the MHPCC portal development team has successfully integrated Portal with the DoD OpenID concept demonstration server. As an initial risk reduction, the MHPCC portal development team has also implemented a traditional PKI-CAC authentication solution also using Central Authentication Service (CAS). Figure 6 shows the common CAC login page using either OpenID or Legacy PKI-CAC.

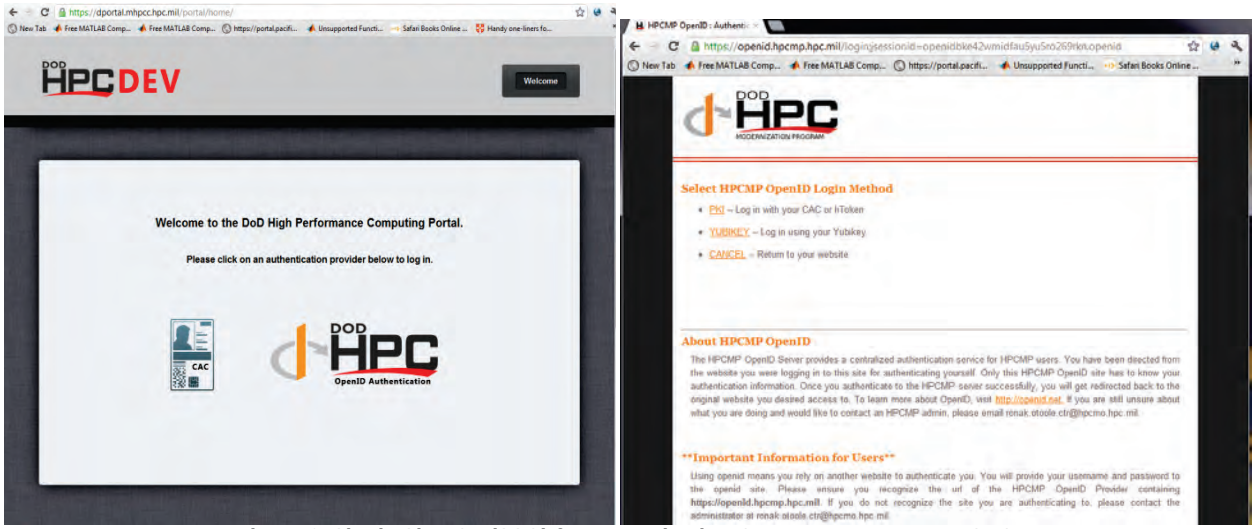


Figure 6. Single Sign On (SSO) from Portal using OpenID or Legacy PKI-CAC

The portal implementation of SSO also requires a security token in support of DoD high-security standards. As of this writing, only the DoD Common Access Card (CAC) token is supported but soon the Yubikey token (shown in Figure 7) will allow access to a wider community.

Figure 7. Yubikey

The Yubikey was selected by the HPCMP as a lower-cost, higher-reliability alternative to existing solutions such as hToken and secureID.

7. User Interface Toolkit

The MHPCC team is leveraging key functionality from ERDC's User Interface Toolkit (UIT) for Portal development including providing authentication for remote HPC services as shown in Figure 8. In direct support of portal objectives and for future applications, a new version of UIT is being developed by ERDC that incorporates OpenID and Oauth in place of Kerberos. Incorporation of UIT into the portal framework should allow easy access to resources across DSRCs. As planned, the portal will demonstrate seamless resource management between MHPCC and ERDC as a proof-of-concept for future DSRC-wide resource sharing.

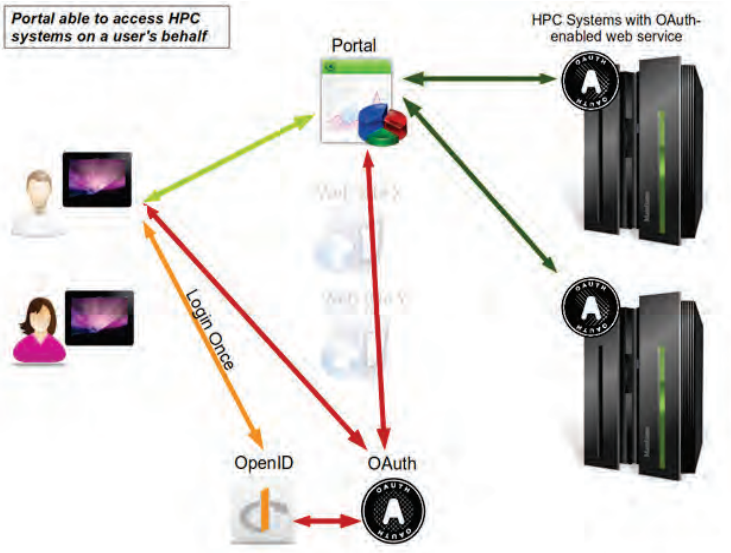


Figure 8. Portal Can Access (Remote) HPC resources on user's behalf

8. CREATE-Kestrel Integration

Overview

Kestrel, part of the CREATE-AV product line, was selected as the first application to be integrated into the portal. Version 2.1.2 of Kestrel has been installed on the portal development platform. An experienced aerodynamicist has been identified to help support testing and compare, contrast, and evaluate the user experience both within and outside the portal environment. Early feedback from the user is considered a critical part of the CREATE portal integration process. Within the next year the entire suite of CREATE applications will be available through the Portal.

Implementation

Kestrel was well-suited for level 1 application delivery, since it is based on the MVC architecture and uses xml messages to communicate. The level 1 integration effort of Kestrel was shared by the CREATE and MHPCC portal teams as shown in Figure 9. The CREATE team will be primarily responsible for separating the control logic from the existing Kestrel client (i.e., Kestrel User Interface or KUI) so that it can reside on the server-side. They provide a JavaScript/HTML5 interface that the MHPCC portal development team will integrate and test within the framework.

As shown, the portal provides a scheduler interface via the UIT framework, and a visualization capability to support a quick-look user assessment of the Kestrel run. Since the Kestrel architecture uses xml for client/server communication, the development teams can easily exploit this existing design feature for interface testing. The teams compare results using a common set of xml files to ensure consistent behavior between the existing code and the incremental portal development builds. The xml file provides a clean, documented interface between teams facilitating development.

A Liferay portlet that provides a user with a web browser-based (zero-footprint) user interface to upload required mesh files and resources to the HPC is provided. The Portal Framework is used to upload files to begin the workflow process. The key advantage of Portal Framework for Kestrel is that it provides quick, controlled access to prior runs and data without requiring continual uploads large mesh files. Under user control, the full state of the system, including code and data, can be saved for future recall.

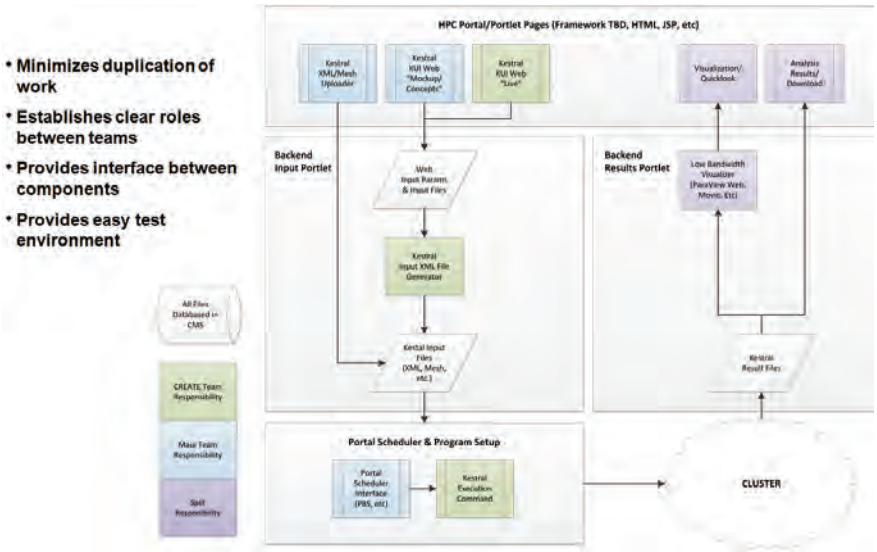


Figure 9. Kestrel integration into Portal

As describe, UIT is a service oriented framework which supports remote job submission and monitoring across all DSRCs. Integration with the UIT framework facilitates support for DSRC job activities in later phases of the project. The Portal leverages UIT to submit jobs to the HPC cluster via the cluster's scheduler (e.g., PBS).

A key feature of the portal implementation is dynamic status displays and quick-look views that help the user determine run-quality at a glance. Shown in Figure 10 is a dynamic convergence plot. As the run progresses, the user is provided a dynamically changing display to provide insight into the run-quality before run completion.

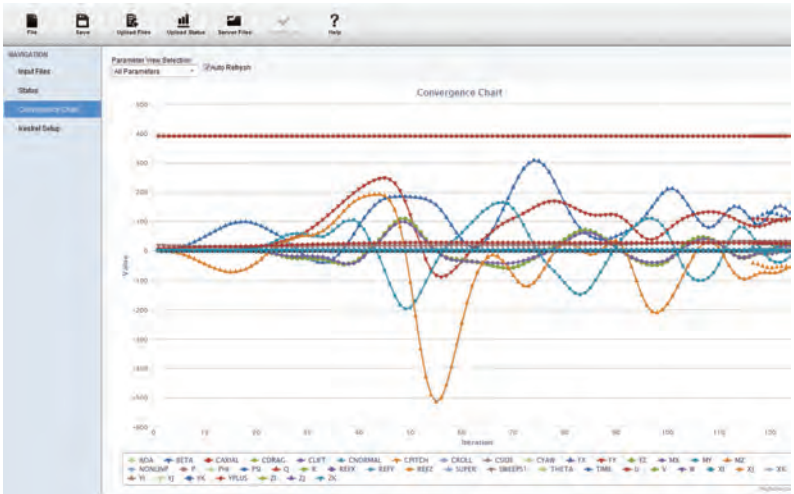


Figure 10. Dynamic convergence display

9. MHPCC MATLAB Development

The goal of the MHPCC MATLAB development effort is to provide the user a nearly effortless way to effectively utilize HPC resources using their existing codes. Mechanisms for both naturally parallel and deep parallel MATLAB will be provided that do not require any HPC or HPC programming knowledge.

Naturally Parallel Concept

Figure 11 shows, at a high-level, how this concept works for a naturally parallel problem. A specific use case of satellite image processing is given to help explain the utility of a real-world example, but is easily generalized to any

naturally parallel problem set. The user supplies a simple "run-function" and the number of iterations N. The run function maps the input parameter n in the range [1, N] to the desired function behavior. Typically, 'n' is used to index a particular set of parameters or set of input files, and the run-function is a trivial MATLAB function typically less than 10 lines long.

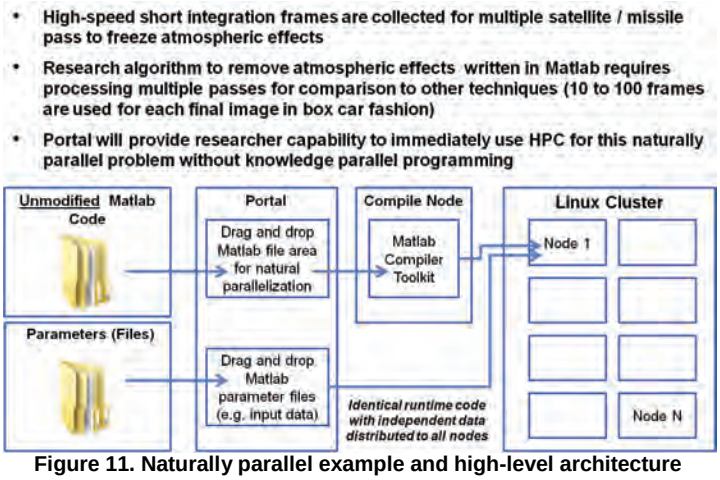


Figure 11. Naturally parallel example and high-level architecture

Since many naturally parallel problems occur in engineering and science, e.g., monte carlo parameter studies, etc., and because such natural problems provide optimal efficiency, this portal service should provide immediate benefit to the engineering community. When this option is selected and any iteration is complete, the results can be inspected immediately from within the portal environment.

Figure 12 shows a real-world example of running a relatively large Matlab code (>500 source files) integrated with the DoD Time-domain Analysis Simulation for Advanced Tracking (TASAT). TASAT itself includes compiled FORTRAN modules which are used to accurately simulate reflectance and scattering of satellite materials.

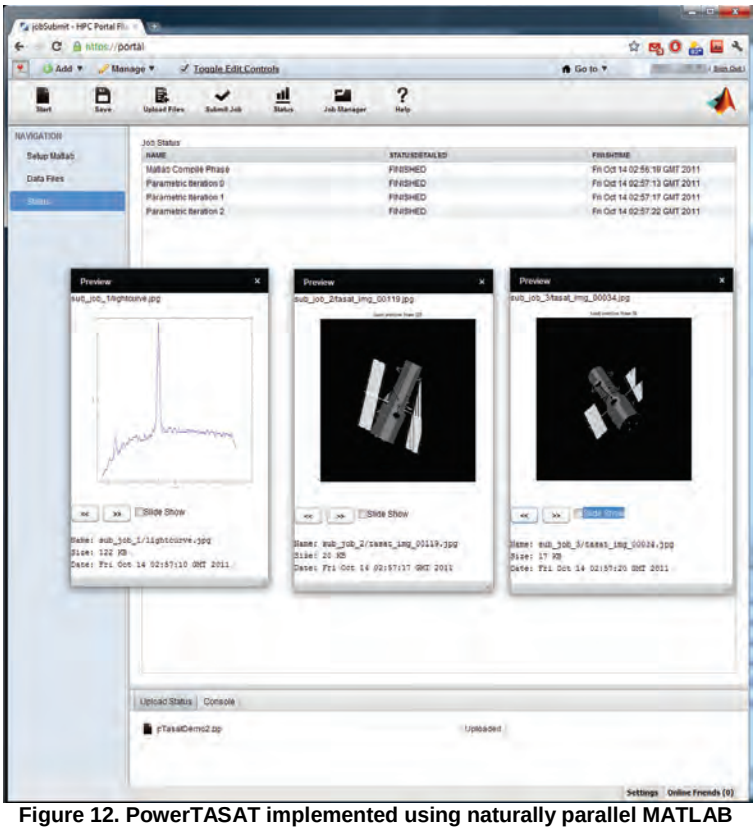


Figure 12. PowerTASAT implemented using naturally parallel MATLAB

10. Deep Parallel Concept

For problems that are not easily solved with simple natural parallel techniques, the MHPCC Matlab solution will also provide a deep parallel capability. No modification to the user code will be required. The technique exploited is to overload key mathematical operations using the MATLAB Mex interface to run on a cluster of nodes. A simple two-dimensional fast-Fourier transform (fft2 in MATLAB) use case is shown in Figure 13. Analogous to the use case for the naturally parallel example, a specific use case of multi-frame blind deconvoloution (MFBD) image processing is given to help explain the utility of a real-world example, but any sufficiently large data set requiring 2D FFT processing would benefit from this processing.

For the deep-parallel capability, the near-term plan is to provide the generic interface and implement several mathematical operations as a concept demonstration. Since a large code-base of open-source software already exists for optimized parallel processing (e.g., LAPAC, BLAS, etc.), a significant portion of the key MATLAB numerical functions could be (easily) included after the concept is demonstrated using the same interface. Additionally, using the Mex interface, such deep parallel libraries, can be used to extend MATLAB beyond its basic functionality to include a wider range of computational capabilities, the domain of HPC, integrating HPC with the most common engineering programming language world-wide.

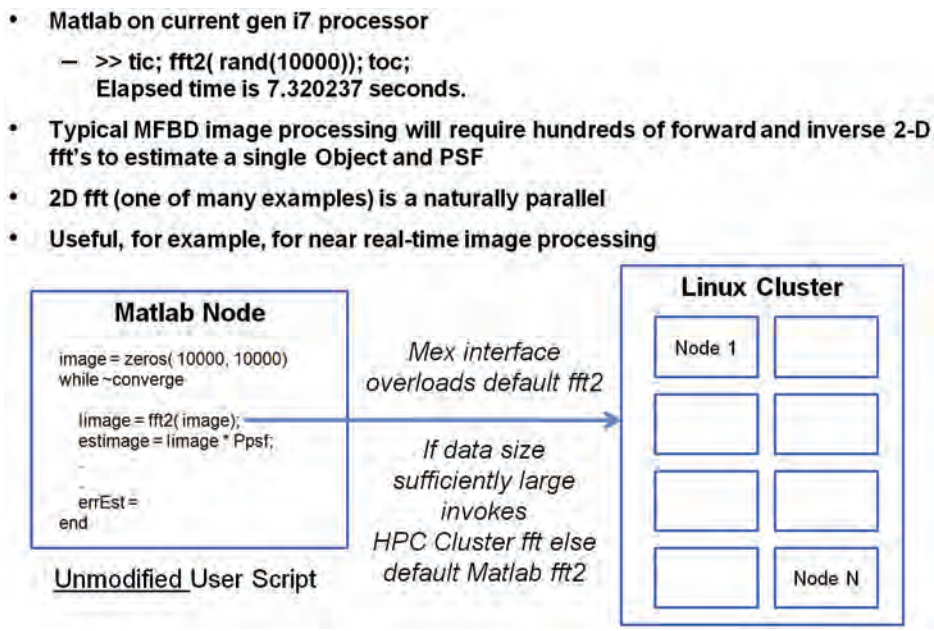


Figure 13. Deep parallel example and high-level architecture

11. Future Capabilities

A number of future capabilities are planned to be implemented in the next 12 months:

- Additional CREATE applications live including DaVinci, Helios, NavyFoam, and NESM.
- An Operational Pilot based on the Kestrel capability.
- Concept demonstration of how portal could automatically scale using resources across DSRCs (ERDC and MHPCC initially) without requiring user interaction.
- Enhanced Data Management will investigate more sophisticated data sharing with the storage initiatives such as the HEUE, Center-Wide File System (CWFS), and SRB implementations within the HPCMP.
- Release of enhanced Portal SDK which will provide an automated stand-alone build capability. A VM test harness will be provided so individual product teams can build the necessary server-side components without standing up a full Portal infrastructure. This will also ensure the UI and workflow framework will integrate easily with the Portal for future builds.

12. Summary

CREATE and MATLAB HPCMP applications are important to the DoD acquisition and engineering communities. The concept demonstration Portal is on track to conclusively demonstrate the high-utility a unified distribution mechanism provides. A number of collaboration services along with the ability to seamlessly access HPC resources within a browser should provide substantial ease-of-use and workflow improvements for these application areas. Three prototype projects using Kestrel at MHPCC have been identified within the US Air Force Aeronautical Systems Center (ASC). These projects will be provided access to both the stand-alone and Portal versions of Kestrel, which should provide detailed insight into the true utility of Portal-based applications and help shape and improve Kestrel and future portal-based, CREATE applications. For MATLAB, after successfully showing seamless integration of TASAT using our MATLAB interface, discussions with the AF Space Vehicles Directorate were initiated to explore the potential of permanently providing TASAT/PowerTASAT capabilities within the portal environment. The goal of this effort is to provide seamless access to HPC capability to the larger TASAT community. The goal of our longer term effort is to work closely with other Government sponsors and targeted HPC user communities to identify additional applications that should be migrated to the portal environment.

6.3 Advanced Technology Development

Army

Accelerating Chemical Kinetics Calculations with GPUs

Christopher P. Stone
Computational Science and Engineering (LLC), Chicago, IL
chris.stone@computational-science.com

Professor Roger R. Davis
University of California, Davis, CA
davisrl@ucdavis.edu

Abstract

A new set of chemical kinetics integration methods using programmable graphics processing units (GPUs) are presented. Implicit and explicit integration methods implemented for CUDA GPUs were used to solve the stiff system of ordinary differential equations (ODEs) arising from complex combustion kinetics problems. The performance impact of mapping one thread to each ODE was compared to mapping one thread-block per ODE problem. The GPU-enabled 4th-order-accurate adaptive Runge-Kutta ODE solver showed speed-ups of 14x over the baseline implicit 5th-order-accurate DVODE CPU run-time for larger numbers of ODEs. A GPU implementation of the implicit DVODE ODE solver achieved a speed-up of 7x over the DVODE CPU run-time. Mapping one thread per ODE problem achieved greater overall speed-up compared to the one block per ODE problem design, but required over 10k ODE problems for parallel efficiency. The one thread-block implementations achieve speed-ups of 10x over the baseline DVODE, but only require 10's of ODEs to outperform the baseline DVODE solver. The new GPU ODE solvers demonstrate a method of significantly accelerating detailed combustion reactions, thereby reducing the simulation turn-around time by an order-of-magnitude.

1. Introduction

The solution of finite-rate chemical kinetics problems in computational fluid dynamics (CFD) combustion simulations can overwhelm the available computational resources. Finite-rate kinetics combustion simulations must solve N_s additional species conservation equations (convection-diffusion-reaction) on top of the traditional mass, momentum and energy conservation equations. Detailed chemical mechanisms can consist of 100's of chemical species with 1000's of elemental reactions leading to intractable storage and computational costs. The CFD algorithm can be simplified by solving the chemical mass action separately from the fluid transport. This approximation allows the chemical kinetics problem at each grid element (cell or vertex) to be integrated independently over the specified time-step size (δt). At each element, the N_s mass fractions and enthalpy (or temperature) are solved as a set of N_s+1 stiff ODE's. Even with the above simplification, the cost of solving the ordinary differential equation (ODEs) systems can be prohibitive requiring massive parallel computing platforms. An alternative approach would be exploiting the available fine-grain parallelism using *graphics processing units* (GPUs).

GPUs contain immense computational power compared to their CPU counterparts. Recently, this power has been exploited to solve computationally-intensive problems of interest to the scientific community. While CPUs may contain, at most, 10's of cores per package, high-end programmable GPUs commonly contain many 100's and can sustain substantially higher memory bandwidths. However, algorithms must be adapted to efficiently fit into the massively-parallel, multi-threaded environment of the GPUs. For NVIDIA *Compute Unified Device Architecture* (CUDA)-based GPUs, blocks of threads are mapped to the multi-processors. The threads within these thread-blocks execute in a single-instruction, multiple-thread (SIMT) paradigm, similar to the SIMD paradigm of vector processors. Large vector operations can be efficiently implemented by mapping thousands of these thread-blocks to the hundreds of cores concurrently. In addition to supporting basic parallel vector operations out of the GPU main memory, thread-blocks have access to a separate shared-memory cache as well as an efficient synchronization mechanism. These two features enable complex iterative and hierarchal algorithms to be implemented within thread-blocks that can significantly increase the computational throughput.

The current GPU application seeks to accelerate the chemical kinetics integration encountered within the *counter-flow linear-eddy model* (CF-LEM). The CF-LEM is used to simulate the turbulent combustion found in opposed-flow configurations (e.g., augmenters) (Calhoon, 2011). The combustion is modeled using finite-rate kinetics to capture complex reaction-diffusion processes such as extinction and re-ignition. The three-dimensional (3D) turbulent fluid mechanics is modeled using a high-resolution 1d turbulent mixing model (i.e., the *linear-eddy model*). In this fashion, the turbulent reaction zone is captured at *direct numerical simulation* (DNS) resolution. An example of a CF-LEM flame profile is shown in Figure 1. There, the major fuel and oxidizer species are shown along with the temperature. Note that only the reaction zone is shown; the full CF-LEM domain is significantly larger. In this example, the combustion kinetics is modeled using a 19-species reduced C_2H_4 (*ethene*) mechanism. See Calhoon, et al. (Calhoon, 2011) for further details on the combustion model.

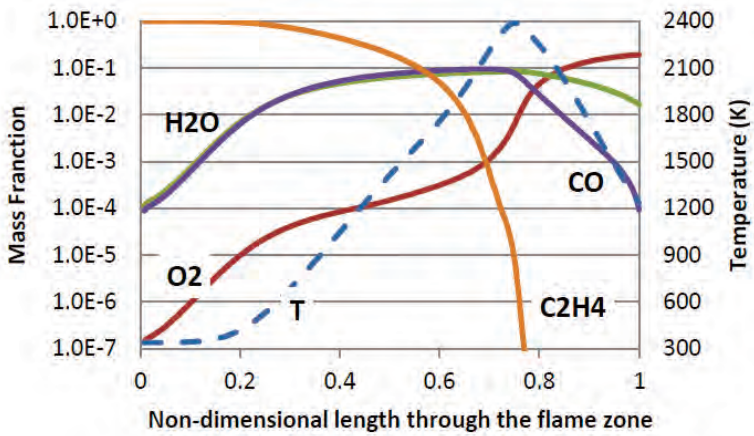


Figure 1. Example LEM flame profile. Only combustion zone of total domain is shown.

GPUs have recently been investigated for stiff chemical kinetics applications. Shi, et al. (Shi, 2011) studied the performance of the GPUs for detailed combustion kinetics. There, reaction mechanisms with over 1000 species were investigated. They reported speed-up only for mechanisms exceeding 300 species. Linford, et al. (Linford, 2009) investigated atmospheric chemistry problems on CUDA GPUs and Cell broadband engines (CBE). They found poor scalability on GPUs compared to CBE due to thread divergence and the large per-problem memory requirement. Both of these previous works will be used as a guide in the current effort. While similar, the present application differs significantly in several key aspects from these two previous investigations: mechanism size and the number of independent ODEs. As mentioned above, the chemistry at each LEM grid-point can be integrated independently. The reaction mechanism of interest is also significantly smaller than those investigated by Shi, et al. Therefore, a GPU method is needed that can rapidly integrate 100's or 1,000's of modestly-sized but independent ODEs. This readily exposes a baseline, coarse-grain level of parallelism that can be exploited.

The rapid integration of combustion kinetics problems is of specific interest to the Department of Defense (DoD). Programmable GPUs have the potential to accelerate specific applications by an order-of-magnitude or more. The kinetics integration cost is a prohibitive and dominate cost to combustion CFD presently. A 10x reduction in the kinetics integration run-time directly translates into a 10x reduction in the simulation turnaround-time along with a sizeable cost reduction.

The following sections provide details on i) the mathematical basis of the kinetics problem, ii) the parallel GPU implementation strategies, and iii) the performance and accuracy of the GPU implementations.

2. Mathematical Model

The LEM combustion model requires the time integration of the N_s species mass fractions (y_k) and temperature (T) equations under the constant pressure assumption at every grid-point. This can be expressed mathematically in the following set of coupled ODEs,

$$\dot{y}_k = \frac{\dot{\omega}_k W_k}{\rho} \tag{1}$$

$$\dot{T} = -\frac{1}{c_p}\sum_i^{N_s} h_i \dot{\omega}_k \tag{2}$$

There, W_k, h_k, and $\dot{\omega}_k$ are the molar mass, enthalpy and molar production rate for species k, and c_p and ρ are the specific heat at constant pressure and mass density of the system. The net molar production rate from combustion is a highly non-linear function of pressure p, T, and the species molar concentrations c_i. It is also the primary source of the ODE stiffness.

The reaction rate terms are computed from a reduced 19-species C_2H_4-Air (*ethene*) reaction mechanism with N_k=167 reversible elemental reactions and N_{qss}=10 quasi-steady-state (QSS) species (i.e., they are not solved in time). Each elemental reaction rate is calculated from a set of temperature- and concentration-dependent factors. The temperature-dependent forward k_f and reverse k_b rate coefficients for each reaction are computed as

$$k_f = A\,T^{\alpha} \exp\left(\frac{-\mathrm{E_a}}{\mathrm{RT}}\right) \tag{3}$$

$$k_b = \frac{k_f}{\exp\left(K_c\right)} \tag{4}$$

where E_a is the activation energy, K_c is the equilibrium constant, A is the pre-exponential rate constant and α is the temperature-dependency. Note, these constants are different for each elemental reaction. The equilibrium constant is computed as a function of the Gibbs energy ($\Delta g_k = \Delta h_k - T\Delta s_k$), the elemental reaction stoichiometric coefficients υ_i, and the reference pressure p_o:

$$K_c = \sum\nolimits_i^{N_\upsilon} \upsilon_i \left(-\Delta G_i\right) / RT + \frac{p_o}{RT} \sum\nolimits_i^{N_\upsilon} \upsilon_i \tag{5}$$

where the summation is over the total number of products and reactants in the given elemental reaction.

The net molar rate-of-progress for each elemental reaction is computed from the above temperature-dependent rate coefficients and the concentrations as

$$\begin{aligned} r_f &= p_{FO} c_{TB} \left[k_f \prod\nolimits_i^{N_p} c_i^{\upsilon_i} \right] \\ r_b &= p_{FO} c_{TB} \left[k_b \prod\nolimits_i^{N_r} c_i^{\upsilon_i} \right] \end{aligned} \tag{6}$$

There, N_p and N_r are the number of products and reactants for the given elemental reaction, c_{TB} is the *third-body* concentration and p_{FO} is the low-pressure factor (i.e., fall-off effect). Of the 167 reactions in the current mechanism, 24 involve *third-body concentrations* and 18 *fall-off* reactions factors. The *third-body concentration* effect for each relevant reaction is modeled as

$$c_{TB} = \sum\nolimits_i^{N_s} c_i + \sum\nolimits_i^{N_{ctb}} \left(A_i - 1\right) c_i \tag{7}$$

The fall-off reaction rates account for low-pressure effects and are themselves functions of the third-body concentrations. Most of the current fall-off reactions follow the 4-parameter *Troe* form of the correction. These all require the calculation of an additional low-pressure forward Arrhenius rate coefficient k_f^o along with a polynomial of temperature:

$$\begin{aligned} p_r &= \frac{k_f^o c_{TB}}{k_f} \\ f_c &= \left(1 - P_1\right) \exp\left(\frac{-T}{P_2}\right) + P_1 \exp\left(\frac{-T}{P_3}\right) + \exp\left(\frac{-P_4}{T}\right) \\ p_{FO} &= g\left(f_c\right) \frac{p_r}{1 + p_r} \end{aligned} \tag{8}$$

The above sequence of rate calculations has neglected the N_{qss} quasi-steady-state species concentrations. These are computed as a complex non-linear combination of r_f and r_b only. The forward and reverse rates are then updated with the new QSS concentrations following the same form as Equation 5.

Finally, the net molar production rate for species k is computed as the sum over all N_k elemental reactions

$$\dot{\omega}_k = \sum\nolimits_i^{N_k} \delta\upsilon_i r_i \tag{9}$$

where $\delta\upsilon_i$ is the net stoichiometric coefficient for species k in reaction i with the net rate-of-progress r_i (i.e., $r_i = r_{f_i} - r_{b_i}$).

All temperature-dependent thermodynamic variables for the chemical species (e.g., c_p, h, and s) are computed from the NASA 7-term polynomial fit database, i.e.,

$$\begin{aligned} c_p / R &= a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4 \\ h / R &= a_0 T + \frac{a_1 T^2}{2} + \frac{a_2 T^3}{3} + \frac{a_3 T^4}{4} + \frac{a_4 T^5}{5} + a_5 \\ (h - s) / RT &= a_0 \left(\log(T) - 1\right) + a_1 T + + a_2 T^2 + a_3 T^3 + a_4 T^4 - \frac{a_5}{T} + a_6 \end{aligned} \tag{10}$$

The parallel implementation of Equations 1 through 10 will strongly influence the computational performance of the ODE integration. Multiple sources of parallelism exist within the LEM application and the governing equations. Notably, each LEM cell represents an independent ODE that can be solved in parallel (i.e., coarse-grain parallelism). On a finer level, the forward and reverse elemental rate coefficients can be calculated in parallel for each individual ODE. The optimal balance between coarse- and fine-grain parallelisms is strongly dependent upon the target hardware. Specific details on the implementation of the above RHS equations and the ODE solver implementations will be given later.

The mass fraction and temperature equations are easily combined into a single vector form suitable for solving with a generic ODE integration algorithm. The following form will be used throughout the remainder of this article:

$$\begin{aligned} \frac{d\vec{u}}{dt} &= \vec{f}\left(\vec{u}, t, \vec{q}\right) \\ \vec{u}\left(t_o\right) &= \vec{u}_o\left(\vec{q}\right) \end{aligned} \tag{11}$$

where $\vec{f}$ is the right-hand-side (RHS) vector function, $\vec{u}$ is the vector of unknowns, $\vec{q}$ is a vector of non-integrated parameters (e.g., pressure), $\vec{u}_o$ is a vector of initial conditions, and t is the independent variable (i.e., time). The length of $\vec{u}$ is N_{eq} (i.e., N_s+1) unknowns. In the current LEM combustion model context, the integration time in Equation 11 is taken as the diffusion time-step size. That is, the ODEs at each grid-point are independently integrated over the same time interval to maintain time-accuracy within the LEM model. Note that a different number of integration steps may be required for each different ODE problem. The ODE integration methods for Equation 11 employed in this study are introduced below.

3. Computational Approach

The DVODE package is commonly used to solve the stiff kinetics ODEs. This algorithm uses a 5th-order-accurate implicit *backwards difference formula* (BDF) to integrate Equation 11. DVODE is a legacy FORTRAN77 application and has no parallel computing capabilities. The CVODES ODE solver (Hindmarsh, 2005), part of the SUNDIALS non-linear solver package from Lawrence Livermore National Laboratory (LLNL), is an updated and enhanced version of DVODE. CVODES is written in C and uses a modular object-oriented programming style. Both the C language and object-oriented design will greatly simplify the implementation of CVODES in CUDA, as detailed later. Note that CVODES is algorithmically equivalent to DVODE; they differ only in their implementation details.

In this study, the DVODE solver will be used as the baseline for both solution accuracy and for CPU (serial) performance. The combustion kinetics is modeled using the reduced 19-species ethene mechanism. Thermo-chemistry properties are computed using the CANTERA (CANTERA, 2010) computational library. CANTERA provides a large set of functions and databases useful for calculating thermodynamic state properties.

A traditional CPU implementation of the above ODE equations would use a single-threaded (i.e., sequential) algorithm, and implements those equations in a fashion that minimizes the number of operations. The minimum number of operations tends to equate to the minimum wall-clock time, the actual goal, with a sequential algorithm. Modern CPUs have large on-chip memory (for both instruction and data), high clock-rates, and relatively large near-chip caches, so this approach is generally optimal. Minimizing total number of sequential operations does not necessarily equate to the minimum time on parallel hardware such as a GPU.

The dominant costs incurred at each ODE integration step with DVODE and CVODES is the RHS evaluation and the solution of the Newton iteration matrix, a square $N_{eq} \times N_{eq}$ matrix formed from the system Jacobian (i.e., $\partial\vec{f} / \partial\vec{u}$). DVODE and CVODES contain many logical tests on convergence and refinement so as to minimize the number of necessary RHS evaluations and matrix computations. Many of these sequential optimizations come at the expense of extra storage

requirements. This is optimal for CPUs with large amounts of on-chip memory and only a few cores. However, it can lead to poor parallel efficiency in a GPU environment where the computational cores are plentiful but the memory is scarce.

GPUs perform most efficiently when each thread executes the same instruction on different data (i.e., single-instruction, multi-data). Each system of ODEs will have a unique set of initial conditions (ICs). Due to the non-linearity of the ODEs, each individual kinetics problem will progress differently through the implicit solvers. This, in turn, can lead to major thread divergence and instruction serialization. An alternative strategy is to integrate the ODE problems using a high-order explicit algorithm such as an adaptive Runge-Kutta method. This approach is typically slower than an equivalent implicit algorithm for stiff problems, due to the necessarily smaller integration step-sizes (i.e., more steps and more RHS evaluations). However, this type of algorithm requires minimal logic and may perform better in a parallel environment. Whether or not an explicit algorithm is numerically stable for the current application must also be investigated.

We have implemented a 4th-order-accurate *adaptive Runge-Kutta-Fehlberg* (ARK) algorithm. This algorithm has the useful property that both a 4th-order- and 5th-order-accurate scheme can be constructed using the same six intermediate RHS evaluation points. The difference between these two schemes gives an estimation of the *local truncation error* (LTE). The solution is monitored for acceptance and the step-size (h) is coarsened or refinement based on the magnitude of the LTE. In this approach, a 4th-order-accurate solution can be *guaranteed*. A solution acceptance criterion similar to that used by DVODE was implemented to ensure the ARK solutions match closely to the baseline implement ODE solver. Refinement is selected when the *weighted-root-mean-square* (WRMS) truncation error is greater than 1. The WRMS error function is defined as

$$WRMS = \sqrt{\frac{1}{N_{eq}} \sum_{k}^{N_{eq}} \frac{\vec{\varepsilon}_k}{rtol\left|\vec{u}_k\right| + atol}} \tag{12}$$

where *rtol* and *atol* are the user-specified relative and absolute tolerance parameters, $\vec{\varepsilon}_k$ is the difference in the 4th- and 5th-order-accurate solutions, and $\left|\vec{u}_k\right|$ is the absolute value of the previous solution. Combining the relative and absolute tolerances allows the species and temperature to be compared more fairly, even through their values differ by many orders-of-magnitude. When WRMS is less than 1, the solution is accepted, h is coarsened and the next integration may proceed. Otherwise, h is reduced and the integration step is repeated. h is modified by the factor $\sqrt[4]{1/2\left(1/WRMS\right)}$. Note that the same adaption factor is valid for both refinement and coarsening. The leading ½ factor in the adaption formula biases h towards a smaller value and avoids stagnation when WRMS≈ 1. The adaption factor is also bounded by 4 and ¼ to avoid large swings and potential overshoots. Finally, h is forced to be no greater than 10% of the total integration time (i.e., a minimum of 10 integration steps).

The initial step-size for ARK is estimated similar to DVODE as well. The initial step-size h_o is estimated such that the WRMS of the 2nd-order expansion term is less than 1, i.e.,

$$\left[\frac{h_o^2}{2}\ddot{y}\left(t\right)\right]_{WMRS} \leq 1 \tag{13}$$

The first step in DVODE is limited to 1st-order accuracy so this estimation ensures the initial step does not produce excess errors. This approach is likely overly conservative for ARK which is 4th-order-accurate for all steps; however, this conservative approach can only benefit the solution accuracy.

The new ARK algorithm was implemented to run on the CPU and applied to 50k ODE chemical integration problems. These problems were randomly chosen from a larger dataset containing over 19 million ICs. Note that these ODE ICs are taken from actual CF-LEM calculations. The relative and absolute tolerances were 10^{-11} and 10^{-13}, respectively. The L_2-norm of the difference between the DVODE and ARK solutions for each solution component is shown in Figure 2. Note that these values have been normalized by the square of reference DVODE values. The maximum normalized difference for any one component is on the order of 10^{-7} and the combined difference over all active species is 10^{-12}. This small difference suggests that the ARK algorithm is suitable for this application.

DVODE, as expected, is faster than ARK. For this benchmark, the DVODE run-time is approximately 50% faster, despite taking twice the total number of integration steps. The cost saving in DVODE comes largely from the reduced number of RHS evaluations per integration step: DVODE required only 1.78 RHS evaluations per-step on average while ARK always requires 6 RHS evaluations. The number of failed integration steps in ARK was minor in this study: only 5.6% of the integration steps failed forcing refinement. As noted above, the sequential cost saving measures taken by DVODE may prove counterproductive in the many-core GPU environment. A more thorough performance analysis of the ARK algorithm compared to DVODE is presented later.

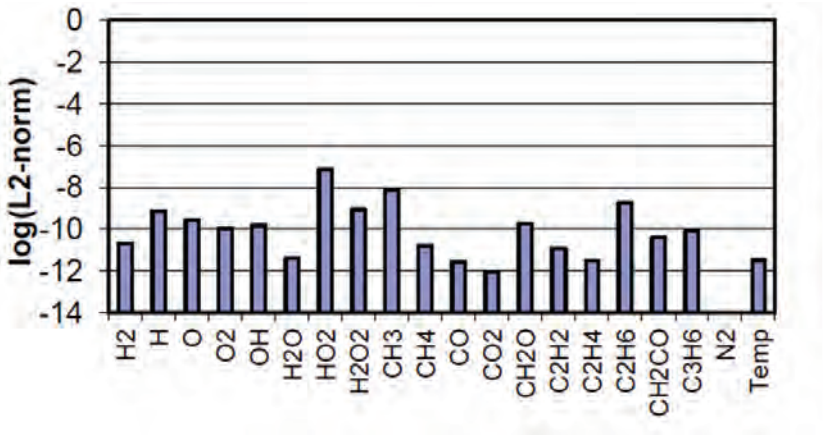


Figure 2. Log of the L2-norm difference between DVODE and ARK on 50k ODE problems

4. GPU Implementations

An efficient GPU implementation must seek to minimize the total run-time and requires a different approach to an optimal sequential algorithm. For this work, the ODE solvers and RHS function were ported to run on NVIDIA CUDA GPUs. The CUDA processor has very limited on-chip memory and a much lower clock-rate than typical CPUs. But, it has a very wide vector processing capability—essentially 1,024 vs. 128 bit vector hardware—and significantly higher memory bandwidth. The high bandwidth must be combined efficiently with the wide vector hardware to offset the lower clock-rate and limited on-chip storage. That is, the parallel CUDA hardware must offset its lower sequential processing power. To achieve this, the reaction mechanism and thermo-chemistry calculations were re-designed to maximize the number of instructions that can be executed in parallel (i.e., *vectorized*), and to maximize the bandwidth from global memory (off-chip) or shared memory (near-chip). A similar strategy was implemented for the ODE solvers themselves.

Two different parallel CUDA versions of the RHS function were implemented. The first method assigns one CUDA thread to each independent RHS evaluation: the *one-thread* algorithm. The second assigns one CUDA thread-block to each independent RHS evaluation: the *one-block* algorithm.

4.1 One-thread Parallel Algorithm

The one-thread algorithm vectorizes across many separate ODE systems, and uses essentially a sequential implementation of the RHS function. However, each instruction is executed by all threads in the thread-block concurrently. This approach exploits the parallelism from having many independent ODE problems that can be computed concurrently. The memory required per-problem far exceeds the on-chip and shared-memory limits; therefore, global memory is the only viable memory space for the unknowns and the elemental reaction rates. All memory loads and stores must be *coalesced* to give optimal bandwidth in this scenario. Coalesced memory is easily achieved when the offset between vector elements is equal to the number of threads in the thread-block, i.e., the stride equals the thread-block size. To this end, a vector storage class was created using C++ templates that automatically adjusts the array *stride* to maximize both the bandwidth and vectorized operations. The reaction mechanism and thermo-chemistry constants were stored in the CUDA *constant* memory. CUDA provides specialized hardware for the *constant* memory, including a dedicated on-chip cache, but it is only suitable when each thread reads the same element. This is easily satisfied in the one-thread approach.

Good performance is only expected with the one-thread approach when the number of independent ODEs is large (e.g., 10,000's) in order to fully saturate the available parallel hardware. This approach is very dependent upon the efficiency of both coalesced memory transactions and vectorized calculations. Both thread divergence and a small number of problems could be very detrimental to the computational throughput. The alternative thread-block approach is designed to address these potential deficiencies.

4.2 One-block Parallel Algorithm

The one-block approach seeks to exploit parallelism within the reduced mechanism itself in addition to the parallelism from multiple ODEs. Instead of mapping one thread to each ODE system, the thread-block approach allocates all the threads in a thread-block to a single ODE problem. Note that many blocks, and ODE problems, may still execute concurrently in parallel. In the one-thread approach, each *for* loop iteration is vectorized across many different ODEs allowing a sequential algorithm to execute in parallel. In the one-block approach, all threads in a thread-block operate on the same ODE's data so vectorization is applied across only a single array to expose the fine-grain parallelism.

As noted earlier, Equations 3–8 individually contain a significant amount of fine-grain parallelism, though they must be computed as a sequence. Each of these equations is vectorized in the one-block approach. For example, the 167 forward elemental reaction rates in Equation 3 are computed concurrently by a separate thread. If viewed as a *for* loop, the loop stride equals the thread-block size (e.g., 64).

The fine-grain vectorized loop is most efficient on large loops. Some operations have smaller vector lengths but can still be executed in parallel using the same *strided* loop method. For example, the Gibbs energy calculation has a vector length of N_s+N_{qss} (i.e., 28) and the final $\dot{y}_k$ calculation has a vector length of only N_s. Each of these vector lengths is less than the thread-block size, so some threads will have no work to perform. In these situations, the computational efficiency may suffer due to the small loop size.

Many other RHS operations are more complicated and cannot be implemented as a simple vectorized loop operation in the one-block approach. The two most prominent are those requiring a *reduction* operation and those with variable vector lengths. A reduction operation computes a single value based on all elements in a vector (e.g., the sum or product of all elements of a vector). For example, the evaluation of $\dot{T}$ requires the sum over all $\dot{y}_k$. A single-thread (sequential) implementation requires N_s operations and N_s steps (i.e., one operation per-step). This algorithm would waste most of the parallel hardware on the CUDA processor and will degrade performance in the one-block method. This same operation can be executed in parallel using a binary reduction algorithm in only $\log_2(N_s)$ steps. The total number of operations is greater than the sequential algorithm, but the number of steps is less. Since each step is executed in parallel, the total run-time is less. For large vectors, this approach is vastly superior on parallel vector hardware. Even though N_s is small compared to the hardware vector length, the potential speed-up is still approximately 4x. When multiple small vector reductions are required and they are independent, each warp computes a separate parallel reduction. Recall, instructions are issued to a warp of threads (i.e., 32 threads). Threads in the same thread-block but in a separate warp can diverge without a performance impact. That is, each warp can do a separate vectorized calculation. Both versions of the parallel reduction algorithm are used whenever possible in the one-block implementation to increase the performance.

Variable-length vector operations are difficult to vectorize efficiently in the one-block approach. The optimal parallel strategy strongly depends on the number of variable-length vector operations that are needed and the minimum and maximum variable-length size. The K_c, r_f, and r_b calculations all require a reduction over the stoichiometric coefficients for each elemental reaction. There are 167 elemental reactions, but only between 1 and 3 productions or reactants each. A parallel reduction over the stoichiometric coefficients would be inefficient in this situation. Instead, the stoichiometric coefficient arrays are padded with 0's and vectorization is applied directly to the 167 elemental reactions. This increases the total number of operations, but also increases the vectorization (and net performance). Note that the one-thread algorithm uses the variable-length summation since each instruction is executed in parallel.

A different approach was required for the calculation of $\dot{\omega}_k$. The number of elemental reactions contributing to the species net production rate ranged from 2 to 47, and averaged 13. The above padding approach would be overly wasteful with such wide variability. And, the standard block parallel reduction would be inefficient for the very small vector-lengths. In this instance, the multi-warp parallel reduction algorithm was used to increase the total parallelism (i.e., each warp executes a separate parallel reduction concurrently).

The one-thread algorithm uses only global memory to store the unknowns due to the large per-problem memory requirement. The one-block approach only stores one problem per thread-block, and requires much less memory (e.g., 64x less memory with 64 threads per thread-block). Because of the smaller storage size, the reaction mechanism data can fit entirely into the on-chip shared-memory in the one-block approach. Like global memory, shared-memory bandwidth is optimal when each thread reads an adjacent array element; otherwise, a *bank conflict* occurs serializing the load or store operations. The optimal vector stride is 1 in the one-block approach. The optimal stride equals the number of threads per thread-block (e.g., 64) in the one-thread approach. The same C++ vector storage class is reused with the array stride of 1 allowing the same source code to be used for both parallel implementations. The one-block algorithm does not have an optimal memory access pattern for c*onstant* memory. Therefore, *global* memory is used to store the reaction rate and thermo-chemistry coefficient data. This is expected to reduce the potential throughput of the one-block approach due to the

lower global memory bandwidth; however, newer Fermi GPUs provide a level of global memory caching which may lessen this impact.

The pseudo-code shown in Listing 1 illustrates the difference between the one-thread and one-block strategies when applied to a simple vector scaling operations a[] **:=** α * b[] for two vectors of length 256 using 32 threads. In the one-thread version, the arrays have an inner-stride equal to the number of threads to maximum memory throughput and vectorization. Each thread iterates through the entire 256 elements concurrently. The GPU processor can execute 32 operations per clock-cycle so all 32 array elements can be processed in only 256 cycles. In the one-block version, only 1 array is processed and the array inner-stride is 1. Each thread processes one element per iteration and skips 32 elements each iteration. Again, since 32 operations can be executed in parallel per cycle, the entire array is processed in only 8 cycles. The one-thread method attempts to process many arrays in parallel; the one-block attempts to process 1 array in parallel.

Listing 1. Pseudo-code demonstrating the difference between the one-thread and one-block approach to the vector scaling operation a[] := α * b[]

One-thread: 32 arrays processed concurrently	**One-block:** 1 array processed in parallel
nThreads := 32 tidx := GetThreadIndex() dim a[nThreads,256], b[nThreads,256] for k=1, 256, 1 a[tidx,k] := b[tidx,k] * alpha end	nThreads := 32 tidx := GetThreadIndex() dim a[1,256], b[1,256] for k=tidx, 256, nThreads end

Equations 1–8 must be executed in sequence due to data-dependency (e.g., K_c requires k_f). This data-dependency requires explicit thread synchronization to avoid race conditions. However, this forced synchronization also allows many intermediate vectors to be reused and safely overwritten, lowering the required shared-memory, a limited commodity.

The last distinction between the parallel algorithms is the treatment of the quasi-steady-state calculation. The QSS component solves a complicated non-linear combination of the elemental reaction rates. Unfortunately, no parallelism is discernible so the entire QSS calculation is computed on only a single thread in the one-block algorithm. The QSS calculation is also a performance limiter for the one-thread algorithm due to the high register usage. Alternative QSS algorithms are needed that are more amendable to vectorization.

4.3 ODE Parallel Implementations

Both the ARK and CVODES algorithms were implemented in CUDA following the one-thread and one-block strategies described above. The one-thread implementation of ARK (CUDA-ARK one-thread) mirrors that used previously by the one-thread CUDA RHS implementation: each thread in a thread-block integrates a separate ODE problem independently over its specified integration time. Note that each ODE problem may have different time integration limits. The same *strided* vector storage class with global memory is used to hold solution and RHS vectors and any scratch space required by the RHS function. At the end of each integration step, the WRMS truncation error is tested for suitability by all threads. If the solution is suitable, each thread updates its solution vector and updates its integration time. Thread divergence is incurred during the adaption test phase but should have only a minor performance impact compared to the 6 RHS function evaluations required each integration step. If the final integration time is reached, the thread exits the integration loop and waits. The threads within the thread-block must wait until the last thread completes its integration. It is possible that a small number of ODEs may require significantly more integration steps causing most threads to stall and, in turn, leading to poor utilization of the CUDA processor. All floating-point and memory operations are vectorized to give high-throughput with global memory.

The optimal thread-block size was found to be 64 for CUDA-ARK one-thread. The CUDA Fermi C2050/M2050 GPUs have 14 MPs and each can support a maximum of 8 concurrent thread-blocks. Therefore, a maximum of 112 thread-blocks are launched. Each thread-block solves a batch of 64 ODEs. Once all threads are complete, the thread-block proceeds to the next batch of 64 problems. Cycling through the ODEs with a fixed number of thread-blocks minimizes the amount of global memory required. At full-occupancy (i.e., 8 thread-blocks are active per MP), the one-thread approach will operate on 7,168 ODE's concurrently.

The one-block CUDA-ARK implementation uses the same vector storage class created for the one-block CUDA RHS function. Shared-memory is used for all vector storage. Explicit synchronization is enforced after each RHS function evaluation to avoid race conditions. After each integration step, the WRMS truncation error is computed using the parallel reduction algorithm. The optimal thread-block size is also 64 for CUDA-ARK one-block and the same block recycling approach is employed. Recall, each problem is solved by an entire thread-block; therefore, a maximum of 112 ODEs can be solved concurrently at full-occupancy.

CVODES was implemented in CUDA using the one-thread approach (CUDA-CVODES one-thread). As discussed above, the one-thread approach takes a nominally sequential algorithm and solves it in a fully-vectorized fashion. This approach allows for straightforward implementations from existing source code. The performance is entirely dependent upon the number of problems to be solved and the efficient coalescing of global memory operations. Much of the original CVODES source code from the SUNDIALS package was reused in the CUDA C++ implementation. All SUNDIALS software uses an object-oriented approach for the vector and matrix storage which greatly simplified the porting to CUDA C++. As before, the *strided* vector storage class was used allowing for coalesced loads and stores from the GPU global memory. The low-level vector and matrix operations used by CVODES (e.g., dot product, scaling and LU decomposition, etc.) were replaced with vectorized CUDA versions using the optimized *strided* vector class.

CUDA-CVODES one-block follows the same approach as the one-block CUDA-ARK and RHS implementations. Again, the object-oriented structure of the CVODES vector and matrix operations simplifies the implementation effort. Each of the vector operations (e.g., dot product) is vectorized across the thread-block. Thread synchronization was critical in the one-block approach so explicit synchronization was enforced on entry and exit from each vector and matrix operation. It is important to note that the number of synchronization points in the CUDA-CVODES one-block implementation is significantly higher than that required for CUDA-ARK one-block and may negatively impact the performance. Vectorized versions of the LU matrix factorization (with partial row pivoting) and back-solve routines were implemented for the CUDA-CVODES one-block algorithm. These two specialized algorithms were implemented using only one thread-warp and not the entire thread-block. This reduced the number of explicit synchronization points required during the matrix decomposition and back-solve and increased the performance.

The CUDA versions of the ODE solvers each require a different amount of memory. The RHS function requires a fixed amount of array storage per-problem, so each ODE solver must provide sufficient space for the RHS solution vector and scratch space. With double-precision, each RHS evaluation requires 3,232 bytes. The one-block version requires additional shared-memory scratch space for the parallel reduction algorithm. This amounts to an extra 512 bytes for 64 threads per thread-block. Note that these storage requirements are only for array data and exclude local scalar variables.

The CUDA-ARK implementations require storage for 6 vectors of length N_{eq} (i.e., 5 intermediate solutions and 1 RHS vector) in addition to the RHS function storage for each ODE problem. CUDA ARK one-block requires 4,704 bytes of shared memory. Recall that each CUDA MP supports up to 48k of shared-memory and 8 thread-blocks per MP. Therefore, there is sufficient shared-memory storage for full-occupancy on the CUDA MP (i.e., $4{,}704\times8<48k$). The limiting factor in the realized run-time occupancy is the register requirement per-thread. The RHS implementation requires the maximum allowable 63 registers per-thread. The high register usage limits the current CUDA implementation's occupancy to only 37.5% meaning only 3 thread-blocks is loaded per MP. Higher occupancy generally leads to higher performance for CUDA programs using global memory heavily.

The CUDA-CVODES implementations require significantly more space than the CUDA-ARK implementations. The vast majority of the extra storage is due to the Jacobian matrix with size $N_{eq}\times N_{eq}$. The register usage is also at the maximum 63 per thread. CVODES and DVODE reuse previously generated Jacobian matrices if the solution vector has not deviated significantly. This requires that the Jacobian is copied and saved each instance doubling the already high storage cost. With this enabled, the CUDA-CVODES requires 11,472 bytes per problem of which the two Jacobian matrices and its pivot array account for 57%. CUDA-CVODES one-block uses shared-memory exclusively. With the Jacobian-recycling strategy, there is insufficient shared-memory for full-occupancy. At the current memory level, the maximum potential occupancy is 50%, a possible performance limitation.

5. Performance and Accuracy

The performance and accuracy of the new GPU-based RHS function and ODE solvers are reported here. All baseline benchmarks were run on a AFRL HPC DSRC *Utility Server* GPU node. Each GPU node has one NVIDIA Fermi M2050 GPU and two 8-core AMD Opteron 6134 Magny-Cours (2.3GHz) CPU's. All source code was compiled with the GCC C++ compiler, the PGI Fortran90 compiler (v11.10) and the NVIDIA CUDA C++ compiler (v4.1.28). All computations were conducted using *double-precision exclusively*. All reported wall-clock measurements were averaged over a minimum of 10

samples, and the sample size was adjusted until the percent standard-deviation was less than 1%. Note that all baseline CPU benchmarks used only one CPU core and all GPU benchmarks used only one GPU.

5.1 RHS Function

The wall-clock time of the two CUDA RHS functions independent of the ODE integrators is shown in Figure 3. The speed-up is also given relative to the CANTERA (CPU) library. A maximum speed-up of 23.1x was obtained using the one-thread approach and 10.7x with the one-block algorithm. The one-block method is approximately 2x slower than the one-thread implementation for large numbers of evaluations. However, for a smaller number of problems, the one-block strategy is far superior. For 100 RHS evaluations, the one-block strategy is 4.3x *faster* than the one-thread method. More importantly, the one-block algorithm for 100 RHS evaluations is faster than the CANTERA CPU version. The one-thread algorithm only becomes faster than the CPU baseline when the number of RHS evaluations exceeds 1,000; whereas the one-block algorithm breaks even at less than 100 evaluations. The lower break-even point of the one-block approach must be balanced with the much faster potential throughput of one-thread for large numbers of problems. Also, the maximum RHS evaluation throughput should be an indication of the maximum ODE solver throughput since the RHS evaluation is the dominant cost.

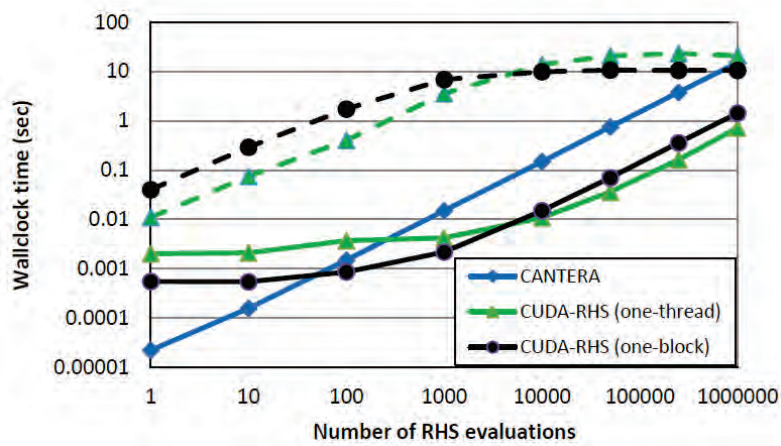


Figure 3. Run-time scaling of baseline CANTERA and CUDA RHS functions. Speed-up shown in dashed line (same scale).

The accuracy of the two CUDA RHS algorithms was also measured compared to the CANTERA baseline. The difference between the two CUDA RHS algorithms and the CANTERA library was computed over 50k randomly selected ODE problems. The maximum difference over all species and temperature equations, in the L_2-norm, was 3.7×10^{-8}. This difference is attributed entirely to the modified format used to store the forward Arrhenius rate coefficients, and is not due to round-off error in the computation. There was negligible difference between the one-thread and one-block versions; any minor difference between the two CUDA versions is due only to the different reduction algorithms. The reduction algorithm changes the order of operations; however, the difference is only on the order of the double-precision round-off error.

5.2 ODE Performance

A detailed run-time profile as a function of the number of ODE's for CPU versions of DVODE and ARK is shown in Figure 4. The CUDA versions of CVODES and ARK are also shown. Some variation in cost is seen for small numbers of ODE's due to the difference in IC's for each problem. These differences are largely averaged out after 1,000 ODE's for both DVODE and ARK. At that point, both DVODE and ARK scale linearly with problem size. ARK is consistently 40% slower over the range of problems with the same absolute and relative tolerances as DVODE. The CUDA-ARK and CUDA-CVODES solvers use the same algorithms as their CPU counterparts. Therefore, the CUDA-ARK implementations must overcome the same 40% performance penalty in order to break even with DVODE.

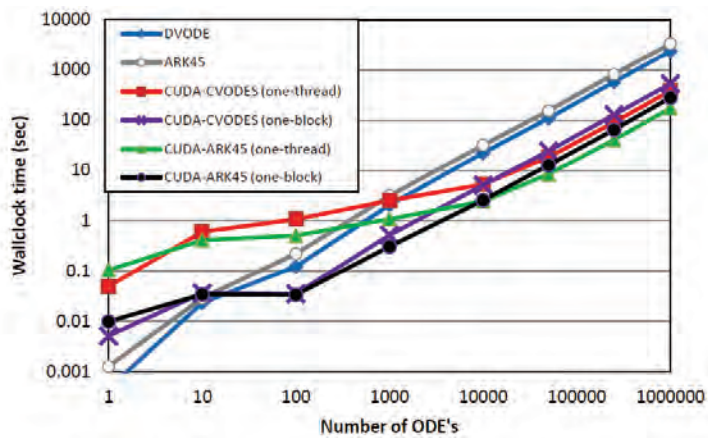


Figure 4. Total run-time (sec) of the baseline CPU DVODE and ARK solvers and their CUDA counterparts

The run-times of each solver normalized by the baseline DVODE CPU run-time are shown in Figure 5. The normalized run-times indicate the speed-up or slow-down compared to the baseline DVODE run on the CPU. CUDA-CVODES one-thread is seen to be many times slower than DVODE until the number of ODE's exceeds 1,000. After this break-even point, the speed-up with CUDA-CVODES one-thread grows slowly, eventually saturating at 7x faster than the baseline DVODE CPU solver. CUDA-ARK one-thread follows a similar scaling trend but is approximately 2.3x faster over the entire range of dataset sizes compared to CUDA-CVODES one-thread. The break-even point with the CUDA-ARK one-thread solver is between 100 and 1,000 problems; the speed-up is already 2.4x at 1,000 problems. The saturation speed-up for the CUDA-ARK one-thread solver is 14x compared to DVODE. Note that CUDA-ARK one-thread is 19x faster than its CPU version using CANTERA. This 19x speed-up matches closely to the CUDA RHS speed-up previously reported suggesting that the RHS function is the limiting factor in the throughput using the CUDA-ARK one-thread method.

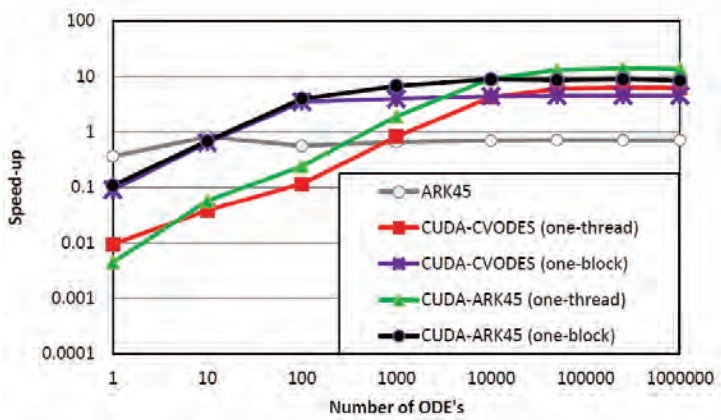


Figure 5. Speed-up of the ODE solvers relative to DVODE

Both one-thread versions of CUDA-CVODES and CUDA-ARK suffer from poor performance when the number of problems is small. This was also observed in the one-thread version of the RHS function shown in Figure 3. However, the estimated CUDA-ARK one-block break-even point with DVODE is only 19 problems; 2 orders-of-magnitude sooner than the one-thread implements. CUDA-ARK one-block quickly saturates to 9x speed-up compared to DVODE at 10k ODE's.

The CUDA-ARK one-block maximum speed-up matches closely to the one-block RHS implementation (i.e., approximately 10x speed-up) clarifying that the RHS function is the limiting factor for both CUDA-ARK implementations.

CUDA-CVODES one-block has identical performance to CUDA-ARK one-block for small numbers of problems. The estimated break-even point with DVODE is 21 problems. The CUDA-CVODES one-block maximum speed-up is 4.5x, only half that achieved by CUDA-ARK one-block. However, both of the one-block CUDA implementations have important performance implications for the LEM application: the flame region is commonly resolved by 100's of cells. That is, the number of ODEs from a single LEM simulation is expected to be on the order of 100's. Therefore, only the one-block CUDA implementations will provide a useful speed-up compared to the baseline DVODE.

6. Conclusion

This research effort has investigated the performance and accuracy of solving stiff chemical kinetics ODEs on CUDA GPUs. The mathematical basis of the kinetics ODEs and how it impacts a parallel implementation was reviewed. Details of the GPU implementation of the RHS function and two ODE solvers using two different parallel strategies, one-block and one-thread, were presented.

The two CUDA ODE solvers, CUDA-CVODES and CUDA-ARK, both achieved significant speed-up compared to the baseline DVODE CPU solver. CUDA-ARK using one-thread-per-ODE achieved a 14x speed-up over DVODE when the number of ODEs is over 10k. The one-block version of CUDA-CVODES achieved a lower speed-up value of 7x compared to DVODE for greater than 10k ODEs. The one-block-per-ODE CUDA implementations both achieved a lower maximum speed-up compared to their one-thread counterparts. However, both CUDA-ARK and CUDA-CVODES one-block reached a break-even point with DVODE at only approximately 20 ODE problems.

The typical LEM simulation uses 100's of cells to resolve the reaction zone. At this problem dataset size, the one-block CUDA ODE solver implementations offer a potential speed-up between 4 and 7x compared to DVODE. Solving multiple LEM simulations concurrently on the host CPU could provide a straightforward method of increasing the number of ODEs available for the GPU. In this manner, the combined LEM simulations can easily achieve speed-up exceeding 10x compared to DVODE.

It is interesting to note that the slower ARK algorithm outperformed the CPU-optimized DVODE and CVODES algorithms when implemented in the massively-parallel GPU environment. The simplified structure of the ARK algorithm provided greater parallel efficiency, easily overcoming a 40% slower sequential performance. The maximum throughput of the CUDA-ARK algorithm is 2x faster than the CUDA-CVODES maximum, and reached a speed-up of 19x compared to the ARK CPU implementation.

The current CUDA implementations of the RHS function and ODE solvers can be further optimized and enhanced. Effort should focus on reducing the register pressure and per-thread local memory usage. The QSS calculation requires a considerable amount of local storage and degrades both the one-thread and one-block throughputs with CVODES and ARK. Methods of reducing the required shared-memory and local-memory usage in CVODES are also necessary. Specifically, the impact of Jacobian recycling should be studied from both a performance and accuracy point of view. The Jacobian recycling method increases the memory storage by approximately 28%. Lowering the shared-memory usage can help increase the occupancy factor.

Acknowledgments

Funding for this work was provided by the DoD HPCMP's User *Productivity Enhancement, Technology Transfer, and Training* (PETTT) Program (Contract No: GS04T09DBC0017 through High Performance Technologies, Inc.) under project PP-CFD-KY02-115-P3 titled *Implementing stiff chemical kinetic ODE solvers on GPUs using CUDA*. Hugh Thornburg (DRC) is PI. DoD users are Balu Sekar, Joseph Zelina and Barry Kiel (AFRL/RZ). The views and conclusions contained herein are those of the authors, and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the US Air Force Research Laboratory or the US Government. The authors would like to thank William Calhoon (CRAFT Tech, Inc.) for his assistance with the baseline DVODE solver and providing the LEM database.

References

Calhoon, W., Zambon, A., Sekar, B., and Kiel, B., "Combustor and augmentor combustion modeling using LES based on stochastic model parameterization." *Proceedings of ASME Turbo Expo 2011*, Vancouver, 2011.

CANTERA, *code.google.com/p/cantera*, 2010.

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., and Woodward, C.S., "SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers." *ACM Transactions on Mathematical Software*, 31, No. 3, pp. 363–396, 2005.

Law, C.K., "Fuel options for next-generation chemical propulsion." *AIAA Journal*, 50, No. 1, 2012.

Linford, J., Michalakes, J., Vachharajani, M., and Sandu, A., "Multi-core Acceleration of Chemical Kinetics for Simulation and Prediction." *SC09*, Portland, 2009.

Shi, Y., Green, W.H., Wong, H.-W., and Oluwole, O.O., "Redesigning combustion modeling algorithms for the Graphics Processing Unit (GPU): Chemical kinetic rate evaluation and ordinary differential equation integration." *Combustion and Flame*, 58, No. 1, 2011.

+Augmenting Live Network Experiments with High Performance Network Simulation and Emulation

Kenneth D. Renard
Secure Mission Solutions, Reston, VA
kenneth.renard@us.army.mil

Brian J. Henz
US Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
brian.j.henz@us.army.mil

Abstract

The goal of the Mobile Network Modeling Institute (MNMI) is to improve the Army's capability for mobile network modeling by exploiting high performance computing (HPC) hardware and software. Network Simulation and Emulation are augmented with Experimentation in a cycle that integrates real-systems that provide a means for validating models of network systems. C4ISR-NM, a partner organization in the MNMI, hosts a series of events to conduct experimentation and integration testing of tactical communications systems using live and virtual entities. Soldiers, vehicles, and tactical systems are sent on "missions" at Ft. Dix, NJ to exercise and test communications technologies of interest to the Army. They interact with each other as well as virtual entities typically implemented in OneSAF, a semi-automated forces simulator. The combination of live and virtual assets allows for a richer environment to test new technologies at a lower cost.

The MNMI has built simulated and emulated network models that run on HPC systems that interact with live and other constructed entities. These models realize a high-fidelity network environment: precise propagation effects, as well as protocol and device characteristics. In this paper, we explore and demonstrate the technologies used, developed, and deployed in these efforts for the 2011 exercises, as well as the ongoing 2012 activities in the MNMI.

1. Introduction

The Mobile Network Modeling Institute (MNMI) aims to improve the Army's capability for mobile network modeling by exploiting high performance computing hardware and software. Network Simulation and Emulation are augmented with Experimentation in a cycle that integrates real-systems and provides a way to validate models of network systems. The MNMI is run by the US Army Research Laboratory (ARL) with partners in the Department of Defense (DoD), academia, and the commercial realm. C4ISR-NM (Network Modernization) is the experimental arm of the Institute and works closely with simulation and emulation efforts.

2. Event 2011 (E11)

C4ISR-NM hosts a series of events during the summer months to conduct experimentation and integration testing of tactical communications systems using live and virtual entities. Live soldiers, vehicles, and tactical systems are sent on missions around Ft. Dix, NJ to exercise and test communications technologies. They interact with each other as well as virtual entities typically implemented in OneSAF, a semi-automated forces simulator. The combination of live and virtual assets allows for a richer environment to test new technologies at a lower cost.

The OneSAF environment set-up for the events typically consists of approximately 3,000 entities. The mobility of these entities can be controlled manually or in a semi-automated fashion. These entities can send and receive certain types of messages with live systems via a middleware machine coupled with a radio transceiver. Within the simulated systems, all communications are "perfect". Every message that is sent is received with no effects considered for radio propagation through distances and terrain.

3. Goals for MNMI Participation

The simulation and emulation capabilities of the MNMI can be used to augment these live exercises in a similar fashion to OneSAF. The tools used implement high-fidelity radio models, as well as realistic communications effects (comm-effects) based on terrain and atmospheric conditions. The throughput, loss, jitter, and latency effects of real wireless networks can be realized, and their 'down-range' effects can be felt by external live systems. The goal of this effort was to establish a "proof-of-concept" and connect MNMI simulation and emulation capabilities to live tactical assets for the purpose of augmenting live exercises. This work would establish network connectivity from the MNMI laboratory to the tactical network, demonstrate basic functionality and interaction with tactical systems, and allow us to further examine what is possible with this type of integration.

4. Network Configuration

The C4ISR-NM tactical network is a "closed" network that does not attach to standard Internet/NIPRNet systems. It was necessary to virtually connect the MNMI simulation and emulation systems via VPN that is tunneled over NIPRNet/Defense Research and Engineering Network (DREN) from the MNMI labs at Aberdeen Proving Ground (APG) to the tactical network at Ft. Dix, New Jersey [see Figure 1]. C4ISR-NM has already established a tunnel from their Live/Virtual/Constructed laboratory at APG, so it was only necessary to set up a VPN between ARL-APG and C4ISR-NM-APG. Network engineers from both organizations set up an IPSec VPN between the two labs, along with a GRE tunnel that would help route multicast traffic over the VPN. The result was an extension of the tactical network into the lab at ARL-APG. The Memorandum of Agreement (MOA) between the two organizations' ISOs (Information Security Officers) was the most time-consuming part of the network setup.

A set of RFC1918 non-routable IP addresses was assigned as a MNMI network by C4ISR-NM. This was split-up into four separate subnets within the ARL lab:

1. Physical subnet to include the VPN endpoint, physical hosts, and emulated NS-3 interface
2. NS-3 subnet that exists entirely within the simulation
3. EMANE subnet for "Over-the-air" network of SRW radios
4. Unused subnet

Static routing was set up within the ARL subnets as necessary.

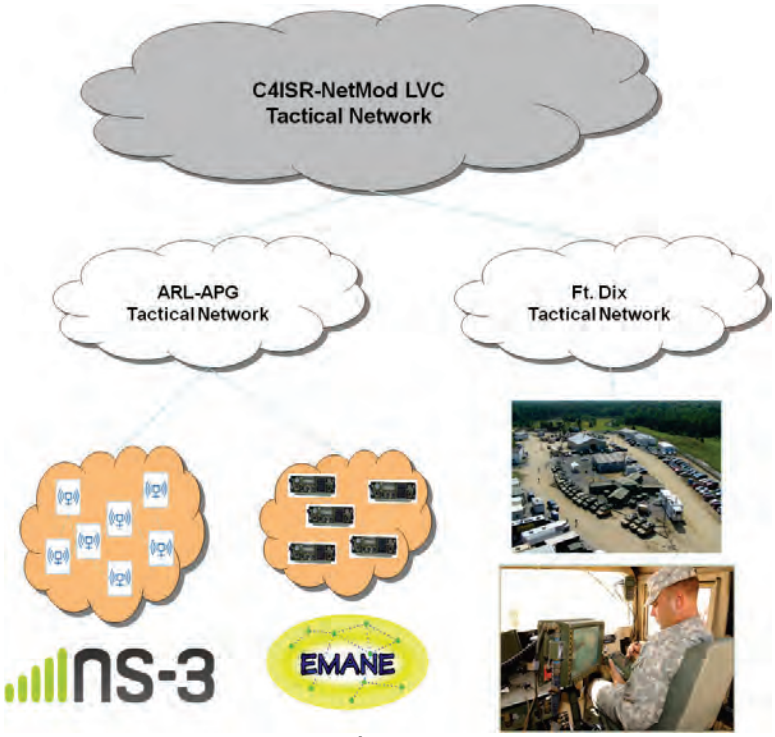


Figure 1

5. Traffic Application

Since the primary goal of this year's effort was to establish connectivity, only a basic situational awareness (SA) application was run on the simulated and emulated nodes. An application was developed for the NS-3 simulator that would poll the node's position and make a determination on whether or not to send a K05.1 VMF (Variable Message Format) packet. These messages are typically sent out after a period of time since the last message sent, or if the node has moved more than a pre-defined distance. Values are typically 3 minutes and 100 meters, respectively. The generation of the K05.1 messages was coded into 2 layers, such that other VMF messages could be built in the future while reusing basic VMF formatting routines. The VMF messages were wrapped in a MIL-STD-2045-47001 header and sent to a configurable IP address over UDP. The application development interface in NS-3 is purposely similar to the sockets-based API of Unix systems, which made the port of the application to the Linux machines running in EMANE somewhat trivial. The main difference was the default MTU that applied to multicast packets in NS-3 versus Linux machines. While NS-3 will send multicast packets with a default MTU of 64, a standard Linux stack uses an MTU of only 1. This was initially not considered in the port to EMANE and resulted in EMANE SA packets not getting all the way through the VPN/tunnel to the Ft. Dix tactical network.

6. NS-3 Simulation

The NS-3 simulator (http://www.nsnam.org/) is a collaborative development effort among a large, international community. This project is funded by a grant from the National Science Foundation (NSF) to the University of Washington and Georgia Tech. The NS-3 simulator has been in use within the MNMI, and has the benefits of being an open-source project and can scale to support very large simulations (hundreds of millions of nodes). There are currently no military radio models supported by NS-3, but the modularity of the code will certainly allow integration of new and ported models. The simulator has an 'emulation' component which can allow it to send and receive packets on real network interfaces that interact with external systems. When used with the NS-3 real-time scheduler, simulated and live systems can interact and exchange packets.

For E11, an NS-3 was set-up to simulate 10 nodes that would exchange SA data with live and other virtual entities. Limiting the simulation to 10 nodes achieves the primary goal of proving that these systems can interact, although NS-3 can scale too much larger node-counts. C4ISR-NM assigned 10 Unit Reference Numbers (URNs) to the simulated NS-3 nodes and designated a geographical area to the northeast of Fort Dix, Range 1 for the entities to navigate in. The NS-3 nodes were restricted to a 100m by 100m box within this designated area, and their movements were independent random walks (select random waypoint within box, move at walking speed to that waypoint, then select a new waypoint). The 10 nodes all have 802.11 interfaces running in ad-hoc mode. One of the 10 nodes is designated as a "gateway" node and has an additional wired interface that connects to external networks and routes traffic to and from the E11 network. OLSR (Optimized Link State Routing) is used over the 802.11 ad-hoc network to provide routing among the 802.11 nodes, and to advertise the route of the gateway node to the rest of the E11 network. The simulation was run on a low-end Linux PC with an Ethernet connection to the VPN endpoint at ARL-APG.

7. EMANE Emulation

EMANE is the Extendable Mobile Ad-hoc Network Emulator developed by CENGEN (http://labs.cengen.com/emane). Development of the EMANE emulator is funded by NRL (US Naval Research Laboratory) and ARL. With emulation, the network stack is modeled in software at the physical and MAC layers with the rest of the stack executed natively on hardware, in this case within a virtual machine running Linux. Emulation provides for the execution of unmodified applications in a virtual, repeatable environment. Currently, EMANE radio models include civilian waveforms such as 802.11 a/b/g and the military's SRW (Soldier Radio Waveform). For this effort the SRW model is used to provide a military-based platform for executing applications and providing VMF situational awareness messages.

For the NM E11, 8 EMANE nodes were created using Linux virtual machines running CentOS. Each of the virtual machines had multiple network interfaces for a control network, OTA (Over-the-Air) network and an NM E11 interface for VMF SA messaging. The emulation nodes were controlled through a JavaScript interface that provides drag-and-drop interactivity with EMANE. In order to tunnel the VMF messages to the DREN connection to Ft. Dix the VMF interface was forwarded to a second port on the physical switch that connects the cluster together and then sent to the DREN uplink.

The EMANE cluster in located in a computing enclave separated from the production ARL network in order to prevent leaking of traffic from the emulated radios onto the production network. The cluster consists of 8 servers with dual quad cores processors, 32GB of RAM and 10x 1 GbE interfaces for high speed connectivity. The cluster also contains a single control server and a workstation with a high-end graphics card. This graphics card is used for real-time RF propagation which is used to compute path loss between the EMANE radios and creates the virtual physical environment for the emulated radios. This path loss can also be used to control the packet loss rate and delay from experimental and simulated devices within the NM E11 environment.

8. Testing

Initial tests were performed on SA packet data being emitted from the NS-3 simulation to assure that the packet format was correct. The VMF Test Tool (VTT) was used by ARL and C4ISR-NM staff to import recorded packet data from the simulation. The VTT read and parsed the packet and affirmed that the binary format of the messages was correct and that the correct versions of the VMF protocols were being used.

9. Results

Once the VPN was established and multicast routing was enabled over the VPN, the NS-3 simulation was started and multicast traffic was verified to be coming from the simulation host to the specified multicast address for SA. C4ISR-NM staff verified that they were able to see the traffic at their lab at APG. Further analysis showed that the traffic was correctly being forwarded to the Ft. Dix tactical network. Finally, a visit was conducted on 21 July 2011 to Range 1 at Ft. Dix where ARL staff was able to see the NS-3 entities on an FBCB2 display in the Tactical Operations Center (TOC) as well as in a remote HMMWV. The FBCB2 was able to "see" the NS-3 nodes due to the VMF SA messages being sent from the simulation. The EMANE nodes were not visible at the time of the visit due to a suspected TTL issue in the application deployed for EMANE. The TTL issue was confirmed and fixed the following day.

The MNMI (NS-3 and EMANE) nodes were not seen in the OneSAF view of the battlefield. This was due to the version of OneSAF being used for the scenario and the fact that a full "C2 adapter" was not available to sufficiently bridge the VMF traffic in and out of the OneSAF simulation.

Several ideas came out of the E11 participation to enhance the scenario and demonstrate more capabilities for high-fidelity, constructed network nodes. The areas of interest for next exercise are:

1. Increase the size of the simulations, potentially to brigade-size
2. Add new applications that generate different VMF message types such as Spot Reports (K04.1), free text (K01.1), MedEvac Request (K07.1)
3. Web and/or scripted interface to control applications and mobility during scenario
4. Using DIS or other data feed from OneSAF to control positions of MNMI nodes
5. Using DIS or other data feed from OneSAF to control application traffic from MNMI nodes
6. Support for IPv6 traffic messaging

Isolating Runtime Faults with Callstack Debugging using TAU

Sameer Shende, Allen D. Malony, and John C. Linford
ParaTools, Inc., Eugene, OR
{sameer, malony, jlinford}@paratools.com

Andrew Wissink
US Army Aeroflightdynamics Directorate (ARMDEC), Ames Research Center, Moffett Field, CA
andrew.m.wissink@us.army.mil

Stephen Adamec
University of Alabama at Birmingham, AL
stephen.a.adamec.ctr@hpcmo.hpc.mil

Abstract

Traditional debugging tools do not fully support state inspection while examining failures in multi-language applications written in a combination of Python, C++, C, and FORTRAN. When an application experiences a runtime fault, such as numerical or memory error, it is difficult to relate the location of the fault to the original source code and examine the performance of the application. We present a tool that can help identify the nature and location of runtime errors in a multi-language program at the point of failure. This debugging tool, integrated in the TAU Performance System®, isolates the fault by capturing the signal associated with it and reports the program call-stack. It captures the performance data at the point of failure, stores detailed information for each frame in the call-stack, and generates a file that may be shipped back to the developers for further analysis. The tool works on parallel programs, providing feedback about every process regardless of whether it experienced the fault. This paper describes the tool and demonstrates its application to the multi-language CREATE-AV applications Kestrel and Helios. The tool is useful to both software developers and to users experiencing runtime software issues as the file output may be exchanged between the user and the development team without disclosing potentially sensitive application data.

1. Introduction

When application software experiences a runtime failure or performance problem, it is important for concise information about the error to be communicated to the development team. Although the next generation of high performance computing (HPC) engineering simulation tools for the Department of Defense (DoD) use a multi-language implementation[1,2], with numeric kernels written in C, C++, or FORTRAN driven by scripting or interpreted languages like Python or Java, current debugging tools do not operate well in this multi-language environment. A multi-language paradigm facilitates modular software and a more straightforward exchange of software between different development groups. However, it introduces a number of complexities in terms of debugging and memory management.

Determining execution context in relation to program parts and the language features used to create them is non-trivial. At present, if software code experiences a runtime issue (e.g., a numerical or memory error), the program generates a core file which, while useful for debugging, contains limited information about the status of the execution, the memory profile, the process call stack, and the system resources in use. In order to resolve the problem, software developers need to understand the nature of the exception, where it occurred, how long the program executed, and the routines invoked prior to the error. A full view of the multi-language execution state, with the attribution of low-level faults to higher-level software context, is required to gain better insight into the layered workings of the application. The same reasoning extends to the importance of retaining other runtime program information that might be useful in understanding what led up to the error. In the case of parallel programs, for example, the ability to save parallel performance measurements after a long-running execution encounters an exception could be extremely valuable, instead of losing all performance information because the tools are unable to retain it. Just being able to properly manage an orderly shutdown of a faulty parallel application when only a few (or even one) process experiences the fault is a challenge. Also, exchanging large core files (generated by each core) with

the developers is cumbersome, especially when working with parallel applications that use dynamic shared objects (DSOs). It is also of limited value when the fault occurs in a DSO where addresses are relative to an offset at which the shared library is loaded at runtime.

Another complexity in HPC software used by the DoD is the mechanism for reporting the software issue. Security requirements in the DoD often preclude the use of forums to discuss software issues if geometry or input conditions are proprietary, sensitive, or classified. Software issues are generally independent of geometry or inputs but the development team must be able to reproduce the issue in order to effectively fix it, and this is difficult for cases with limited distribution rights.

This paper presents a diagnostic tool for runtime faults that aims to replace the standard core file output with more effective and concise information gathered during execution and at the time of the errant behavior, including data about the computational performance, memory usage, call stack, and input/output (I/O) operation. The diagnostic file is self-contained and interchangeable between machines in the TAU standard profile format, so a user that experienced an execution problem can send the diagnostic file to the development team for further analysis to determine the source of the problem. The format for this file is human readable and open.

The next section presents the design approach and technical issues. Section 3 describes the development of the tool for multi-language applications based on a Python-driver model. We have successfully applied this tool to the Helios[1] and Kestrel[2] codes and we report our findings. Related work is presented in Section 4. We conclude the paper with a discussion of the next development and evaluation steps.

2. Design Approach

When an application experiences a runtime error such as a segmentation violation (e.g., caused by an access through a null pointer), execution of an illegal instruction, or floating-point exception (e.g., division by zero), it is critical to diagnose the problem with respect to the execution context. Merely reporting the text output of the execution is rarely sufficient to fix the problem, particularly in multi-language applications. The developers typically need to understand the nature of the exception, where it occurred, how long the program executed, and the routines invoked prior to the error. This requires a full view of the execution state to gain a better insight into the program's layered workings of the multi-language program. Additionally, system information such as the operating system kernel version, the extent of heap memory utilization, number of cores, and other application specific parameters are important in fully understanding the error.

The key to solving the problem of observing execution state is in capturing (unwinding) the call stack at the location of the error across the many layers of languages and libraries. The diagnostic tool we created integrates a call stack capture module in TAU to accomplish this[20]. TAU unwinds the calling stack of each thread of execution and records it in the profile file format with context specific information such as the calling routine name, file name, and source line number for each frame in the program's calling stack, when available. The tool operates at runtime, automatically interposing with the application to generate diagnostics. This involves registering signal handlers, examining the program callstack using GLIBC library *backtrace* Application Programming Interface (API)[15] at the point of failure, maintaining and updating address maps for DSOs as they are loaded and unloaded, and translating the address from each frame in the callstack using the GNU *Binutils*[13] to useful program information before the information is lost and the program terminates. However, a tool to recover state information from an errant parallel program must consider the cases where only a subset (perhaps just one) of the threads experiences a failure. Here, it is necessary to inform the other processes to gracefully terminate, capturing relevant diagnostic information as they do so. The design choices for this tool are described in Reference 20.

3. Implementation

We have developed a callstack capture module in TAU that works with Python and can help us understand the I/O and memory usage in a mixed language application at the point of failure. The user activates this module by setting the *TAU_TRACK_SIGNALS* environment variable prior to executing the program using *tau_exec*. The *tau_exec* tool preloads a TAU DSO in the address space of the executing application and registers the handlers for capturing a signal at the point of failure. Additionally, a TAU Python wrapper helps to instrument all Python functions. Figure 1 shows how the application is invoked with the I/O and memory inspection command line options. No changes are required in the application source code or binaries. If and when a runtime fault occurs, TAU isolates it by capturing the callstacks for each rank of the Message Passing Interface (MPI) program. There are two distinct callstack entities—the system callstack and the callstack maintained by TAU. The TAU callstack represents the sequence of events that are recorded by entry and exit instrumentation in interval timers. Because TAU supports a wide variety of programming languages, this callstack is

independent of the language and we can see a combination of events from C, C++, FORTRAN, and Python in this view. The Python events represent routine names in the Python scripts that TAU gets from the interpreter. These events are typically not visible to traditional debuggers such as *gdb*[14]. By combining the two callstacks consistently within one tool, TAU permits debugging of system backtraces along with performance data generated by higher language level interval and atomic events providing a consistent view that will allow developers to get an accurate view of the program execution with or without fault diagnostic information.

TAU is integrated in the runtime layer of the *PToolsRTE* package[21] that is used by the CREATE-AV codes. This allows a user to source a single configuration file or load an environment module to set all paths needed for launching the codes using TAU, as shown in Figure 1.

```
[sameer@m0297 example]$ source $PET_HOME/pkgs/ptoolsrte/etc/ptoolsrte.cshrc
[sameer@m0297 example]$ which pyMPI
/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/pyMPI-2.5b0/bin/pyMPI
[sameer@m0297 example]$ which tau_exec
/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/tau-2.21.2/x86_64/bin/tau_exec
[sameer@m0297 example]$ setenv TAU_TRACK_SIGNALS 1
[sameer@m0297 example]$ setenv TAU_CALLPATH_DEPTH 100
[sameer@m0297 example]$ cat wrapper.py
import tau

def OurMain():
    import samarcrun

tau.run('OurMain()')

[sameer@m0297 example]$ mpirun -np 2 tau_exec -T icpc,python,mpi,pdt -io -memory pyMPI ./wrapper.py

initializing SAMARC
SAMINT::initialize()
building SAMARC initial AMR mesh
SAMINT::buildGrids()
printing output
SAMINT::getGlobalNumberPatches()
SAMINT::getLocalNumberPatches()
SAMINT::writePlotData()
-----------------------------------------------------
            step:  1     time:  0.0
SAMINT::timestep()
initializing SAMARC
SAMINT::initialize()
building SAMARC initial AMR mesh
SAMINT::buildGrids()
printing output
SAMINT::getGlobalNumberPatches()
SAMINT::getLocalNumberPatches()
SAMINT::writePlotData()
-----------------------------------------------------
            step:  1     time:  0.0
SAMINT::timestep()
TAU: Caught signal 8 (Floating point exception), dumping profile with stack trace: [rank=1, pid=25501, tid=0]...
TAU: Caught signal 8 (Floating point exception), dumping profile with stack trace: [rank=0, pid=25502, tid=0]...
--------------------------------------------------------------------------
mpirun has exited due to process rank 1 with PID 25428 on
node m0297.mana exiting without calling "finalize". This may
have caused other processes in the application to be
terminated by signals sent by mpirun (as reported here).
--------------------------------------------------------------------------
[sameer@m0297 example]$ paraprof --pack create_io_mem.ppk
Loading data...
Packing data...
[sameer@m0297 example]$
```

Figure 1. Executing an application with the tau_exec tool captures the program state when a runtime fault occurs

Launching TAU's *ParaProf* profile browser allows the user to examine the callstacks. Figure 2 shows the system callstack at the point of failure in Kestrel initialization code. This data is captured in the BACKTRACE metadata fields. Figure 3 shows a small segment of the callstack, zoomed in for better readability. By clicking on an entry, TAU's source browser can show the source code at the location of the segmentation fault for any frame.

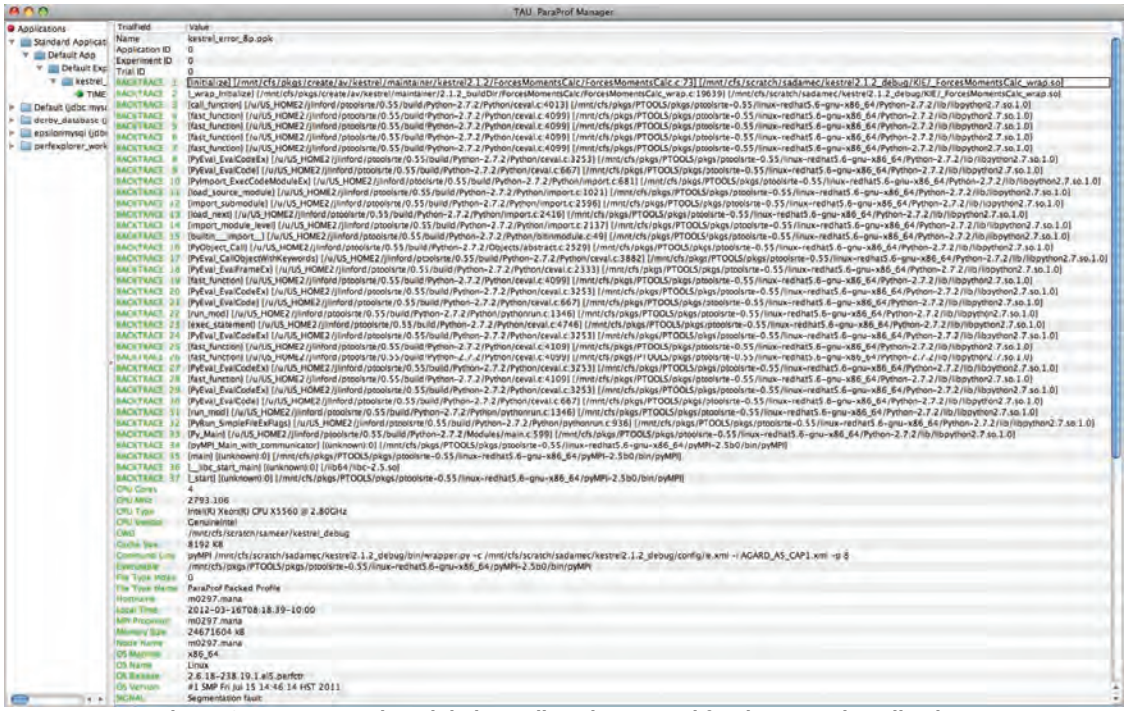


Figure 2. A segmentation violation callstack captured for the Kestrel application

Figure 3. Zooming into a segment of the callstack shows the routine name and the source code location

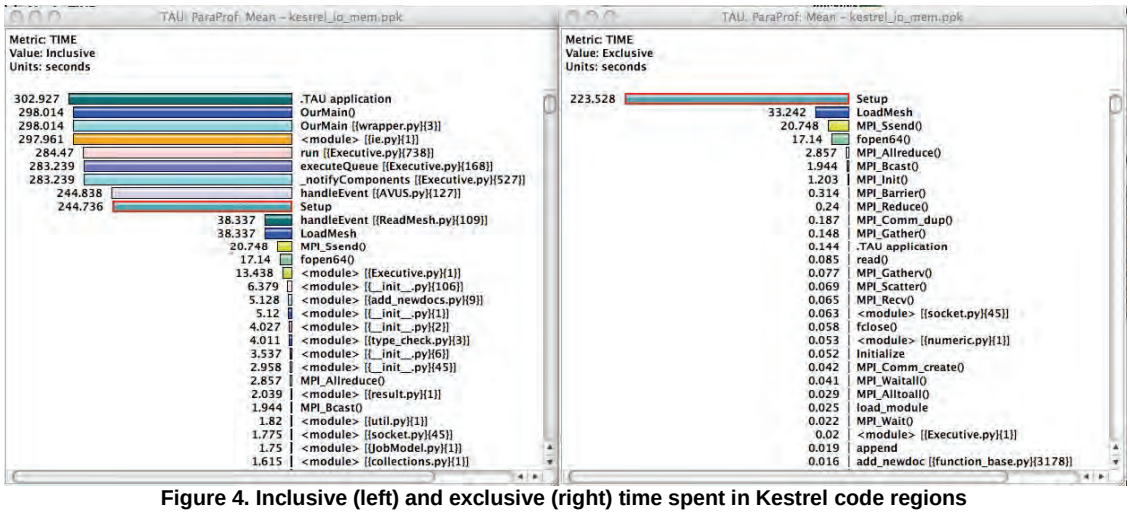
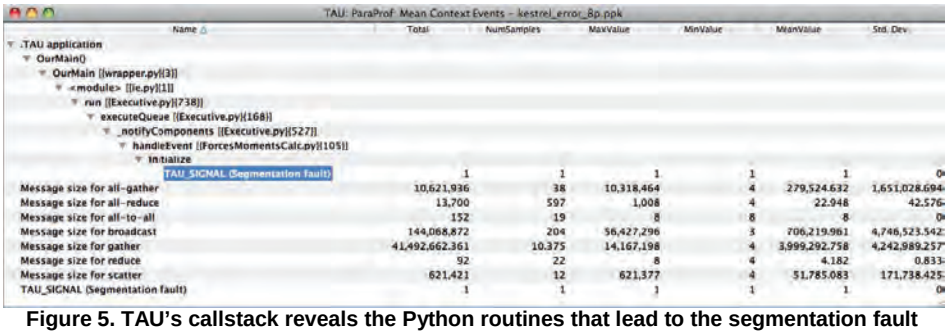


Figure 4. Inclusive (left) and exclusive (right) time spent in Kestrel code regions

In addition to capturing the frames, the tool retains the performance data TAU has collected for the run. Figure 4 shows the exclusive and inclusive time spent in code regions. We see that the application executed on 8 processors for 302. Nine (9) seconds before terminating with the error. In addition to the code region performance data, the TAU callstack information reveals the Python routines that were called before the segmentation fault. Because I/O activity is also measured, the message sizes for communication operations can be record. Figure 5 highlights these features for the Kestrel testcase.



TAU: ParaProf: Mean Context Events - kestrel_error_8p.ppk

Name	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
.TAU application						
OurMain()						
OurMain [{wrapper.py}{3}]						
<module> [{ie.py}{1}]						
run [{Executive.py}{738}]						
executeQueue [{Executive.py}{168}]						
_notifyComponents [{Executive.py}{527}]						
handleEvent [{ForcesMomentsCalc.py}{105}]						
Initialize						
TAU_SIGNAL (Segmentation fault)	1	1	1	1	1	0
Message size for all-gather	10,621,936	38	10,318,464	4	279,524.632	1,651,028.694
Message size for all-reduce	13,700	597	1,008	4	22.948	42.576
Message size for all-to-all	152	19	8	8	8	0
Message size for broadcast	144,068,872	204	56,427,296	3	706,219.961	4,746,523.542
Message size for gather	41,492,662.361	10.375	14,167,198	4	3,999,292.758	4,242,989.257
Message size for reduce	92	22	8	4	4.182	0.833
Message size for scatter	621,421	12	621,377	4	51,785.083	171,738.425
TAU_SIGNAL (Segmentation fault)	1	1	1	1	1	0

Figure 5. TAU's callstack reveals the Python routines that lead to the segmentation fault

Metadata for n,c,t 7,0,0

Name	Value
BACKTRACE 1	[Initialize] [/mnt/cfs/pkgs/create/av/kestrel/maintainer/kestrel2.1.2/ForcesMomentsCalc/ForcesMomentsCalc.c:73] [/mnt/cfs/scratch/sadamec/kestrel2.1.2_debug/KIE/_ForcesMomentsCalc_wrap.so]
BACKTRACE 2	[_wrap_Initialize] [/mnt/cfs/pkgs/create/av/kestrel/maintainer/2.1.2_buildDir/ForcesMomentsCalc/ForcesMomentsCalc_wrap.c:19659] [/mnt/cfs/scratch/sadamec/kestrel2.1.2_debug/KIE/_ForcesMomentsCalc_wrap.so]
BACKTRACE 3	[call_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4013] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 4	[fast_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 5	[fast_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 6	[fast_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 7	[fast_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 8	[PyEval_EvalCodeEx] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 9	[PyEval_EvalCode] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 10	[PyImport_ExecCodeModuleEx] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/import.c:681] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2...
BACKTRACE 11	[load_source_module] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/import.c:1021] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 12	[import_submodule] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/import.c:2596] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 13	[load_next] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/import.c:2416] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 14	[import_module_level] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/import.c:2127] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 15	[builtin___import__] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/bltinmodule.c:49] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 16	[PyObject_Call] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Objects/abstract.c:2529] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 17	[PyEval_CallObjectWithKeywords] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:3882] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython...
BACKTRACE 18	[PyEval_EvalFrameEx] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:2333] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 19	[fast_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 20	[PyEval_EvalCodeEx] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 21	[PyEval_EvalCode] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 22	[run_mod] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/pythonrun.c:1346] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 23	[exec_statement] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4746] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 24	[PyEval_EvalCodeEx] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 25	[fast_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4109] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 26	[fast_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 27	[PyEval_EvalCodeEx] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 28	[fast_function] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:4109] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 29	[PyEval_EvalCodeEx] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 30	[PyEval_EvalCode] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 31	[run_mod] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/pythonrun.c:1346] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 32	[PyRun_SimpleFileExFlags] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Python/pythonrun.c:936] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7...
BACKTRACE 33	[Py_Main] [/u/US_HOME2/jlinford/ptoolsrte/0.55/build/Python-2.7.2/Modules/main.c:599] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 34	[pyMPI_Main_with_communicator] [(unknown):0] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/pyMPI-2.5b0/bin/pyMPI]
BACKTRACE 35	[main] [(unknown):0] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/pyMPI-2.5b0/bin/pyMPI]
BACKTRACE 36	[__libc_start_main] [(unknown):0] [/lib64/libc-2.5.so]
BACKTRACE 37	[_start] [(unknown):0] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/pyMPI-2.5b0/bin/pyMPI]
CPU Cores	4
CPU MHz	2793.106
CPU Type	Intel(R) Xeon(R) CPU X5560 @ 2.80GHz
CPU Vendor	GenuineIntel
CWD	/mnt/cfs/scratch/sameer/kestrel_debug
Cache Size	8192 KB
Command Line	pyMPI /mnt/cfs/scratch/sadamec/kestrel2.1.2_debug/bin/wrapper.py -c /mnt/cfs/scratch/sadamec/kestrel2.1.2_debug/config/ie.xml -i AGARD_AS_CAP1.xml -p 8
Executable	/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/pyMPI-2.5b0/bin/pyMPI
File Type Index	0
File Type Name	ParaProf Packed Profile
Hostname	m0297.mana
Local Time	2012-03-16T08:18:39-10:00
MPI Processor Name	m0297.mana
Memory Size	24671604 kB
Node Name	m0297.mana
OS Machine	x86_64
OS Name	Linux
OS Release	2.6.18-238.19.1.el5.perfctr
OS Version	#1 SMP Fri Jul 15 14:46:14 MST 2011
SIGNAL	Segmentation fault
Starting Timestamp	1331921919260893
TAU Architecture	x86_64
TAU Config	<wrapper -bfd=/mnt/cfs/pkgs/PTOOLS/pkgs/tau-2.21.2/x86_64/binutils-2.20 -pythoninc=/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-gnu-x86_64/Python-2.7.2/include/python2.7 -c++=icpc -c
TAU Makefile	/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte/linux-redhat5.6-gnu-x86_64/tau-2.21.2/x86_64/lib/Makefile.tau-icpc-mpi-python-pdt
TAU Version	2.21.2

Figure 6. System callstack leading up to the fault on the MPI rank 7

It is important to understand that a distinct callstack and performance profile is captured for each rank. Different ranks may have different faults (or none at all) and it is necessary to see all states. It is possible to analyze any particular rank, such as rank 7 in Figure 6. The location of the error can also be seen across all ranks in Figure 7 (the bar graph indicates the number of samples of the error event on each rank).

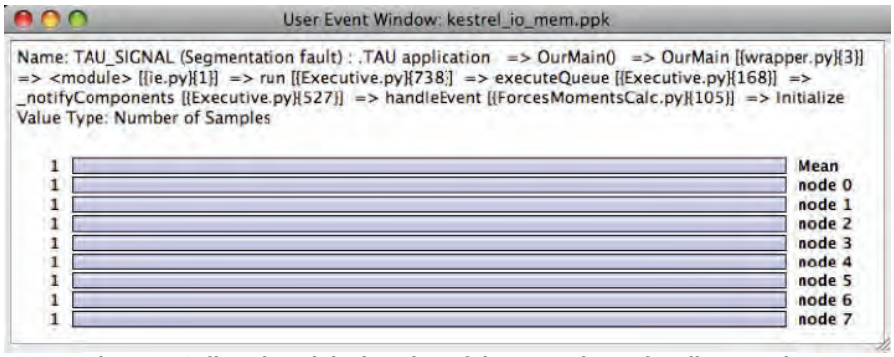


Figure 7. Callstack and the location of the error shown for all MPI ranks

The same application can be re-executed to generate I/O and memory data using *the -io -memory* command line options for *tau_exec*. Figure 8 shows the peak heap memory usage (~1.3GB) for the application on rank 0. Figure 9 shows the TAU callstack with the I/O and memory statistics leading up to the point of failure on rank 0. The total memory allocated till the point of failure was ~3GB and the total de-allocated was 2.24GB.

TAU: ParaProf: Context Events for: n,c,t 0,0,0 – kestrel_io_mem.ppk

Name	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
.TAU application						
Memory Utilization (heap, in KB)		47,690,869	1,319,090.592	0.023	275,509.524	38,071.609
malloc size (bytes)	3,012.988,071	23,869,030	496,523,280	1	126.23	113,322.074
free size (bytes)	2,248.780,956	23,821,839	496,523,280	1	94.4	108,796.005
Bytes Read	158.530,329	19,180	1,126,076	2	8,265.398	9,607.032
Read Bandwidth (MB/s)		19,156	10,982	0.051	1,209.371	277.131
Bytes Read <file=/mnt/cfs/pkgs/create/av	151.920,360	18,545	8,192	8,172	8,191.985	0.543
Read Bandwidth (MB/s) <file=/mnt/cfs/pk		18,545	2,048	327.68	1,238.092	153.524
Message size for all-reduce	13,700	597	1,008	4	22.948	42.576
Bytes Written	64,053	238	4,161	2	269.13	973.394
Write Bandwidth (MB/s)		212	1,040.25	0.085	46.81	174.011

Figure 8. Memory and I/O diagnostic information for rank 0

4. Inclusion of Performance Analysis

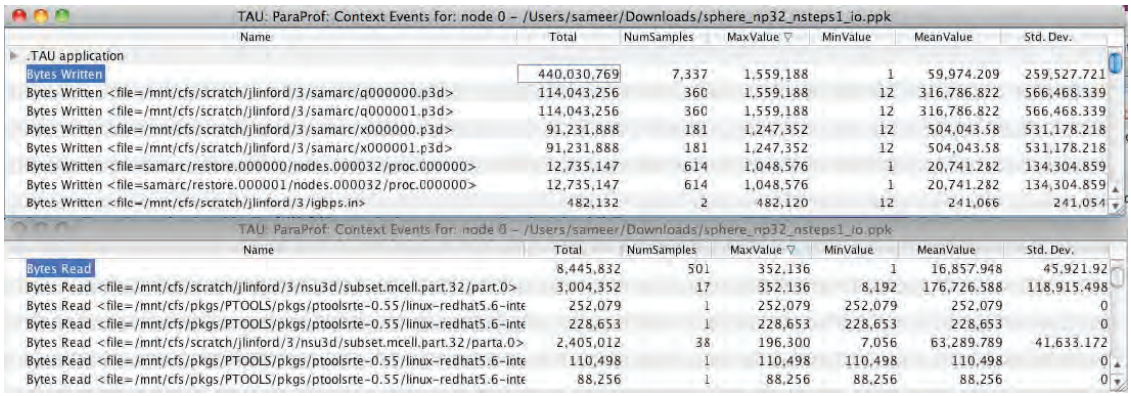
The new debugging support in TAU discussed above works seamlessly with the performance measurement and analysis that TAU has traditionally been used for. Performance testing is not impeded in any way by having debugging enabled. To demonstrate this, we included performance evaluation of the CREATE-AV rotary-wing code Helios[1], which performs high-fidelity modeling of rotorcraft aero and structural dynamics. Helios consists of multiple components performing different parts of the multi-disciplinary application—computational fluid dynamics (CFD), computational structural dynamics (CSD), six degree-of-freedom (6DOF) dynamics, etc. The different components are written in different languages—FORTRAN90, C, and C++—which are integrated through a high-level Python-based infrastructure. Further details on the implementation and validation of Helios are included in Reference 1. Details of prior work integrating TAU with Helios are included in Reference 11.

TAU: ParaProf: Context Events for: node 0 – kestrel_io_mem.ppk

Name	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
.TAU application						
MPI_Init()						
OurMain()						
OurMain [[wrapper.py](3)]						
<module> [[io.py](1)]						
<module> [[Executive.py](1)]						
fclose()						
fopen64()						
read()						
Read Bandwidth (MB/s)		1	755.538	755.538	755.538	0
Read Bandwidth (MB/s) <file=/mnt/cfs/scratch/sadamec/kestrel2.1.2_debug/KIE/Executive.pyc>		1	755.538	755.538	755.538	0
Bytes Read	19,644	1	19,644	19,644	19,644	0
Bytes Read <file=/mnt/cfs/scratch/sadamec/kestrel2.1.2_debug/KIE/Executive.pyc>	19,644	1	19,644	19,644	19,644	0
run [[Executive.py](718)]						
__init__ [[ElementTree.py](607)]						
__init__ [[Event.py](19)]						
__init__ [[EventQueue.py](14)]						
__init__ [[Executive.py](69)]						
__init__ [[JobModel.py](19)]						
addComponents [[JobModel.py](49)]						
allreduce						
append						
bcast						
checkResources [[Executive.py](103)]						
close						
executeQueue [[Executive.py](168)]						
_notifyComponents [[Executive.py](527)]						
bcast						
gather						
handleEvent [[AVUS.py](127)]						
handleEvent [[ForcesMomentsCalc.py](105)]						
initialize						
fclose()						
fopen()						
fopen64()						
fprintf()						
read()						
TAU: SIGNAL (Segmentation fault)	1	1	1	1	1	0
free size (bytes)	1,095,600	3,069	231,808	2	356.989	7,021.732
malloc size (bytes)	10,372,529	17,625	1,126,076	2	588.512	12,114.147

Figure 9. Location of error with memory and I/O combined with the callstack data

To demonstrate use of the new tool to assess multi-language performance, we executed Helios using *tau_exec*. Figure 10 shows the profile from Helios with TAU instrumentation at the Python, mpi4py, MPI, C++, C, and FORTRAN. It shows an integrated view of fault diagnostic information with the TAU callstack that shows Python entities clearly in the profile. PToolsRTE v0.55 supports both pyMPI and mpi4py based executions using TAU. While the earlier example with Kestrel shows the performance data truncated at a location of failure, the user may use TAU to investigate the performance characteristics of the test case that executes without an error as well, as shown in Figure 10. Here we see the I/O characteristics of the application executed on Mana, a Dell system at the MHPCC DSRC. It shows that MPI rank 0 wrote ~440MB and read ~8MB of data. It also shows the extent of I/O performed on each file. Figure 11 shows the peak read bandwidth obtained for all ranks of the application. Here we see that during the course of the entire execution, rank 18 had a peak bandwidth of 4,089 MB/s compared to nearly 8,178 MB/s for other ranks. Having access to such detailed information about each rank and file can help developers identify sources of performance bugs.



TAU: ParaProf: Context Events for: node 0 - /Users/sameer/Downloads/sphere_np32_nsteps1_io.ppk

Name	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
.TAU application						
Bytes Written	440,030,769	7,337	1,559,188	1	59,974.209	259,527.721
Bytes Written <file=/mnt/cfs/scratch/jlinford/3/samarc/q000000.p3d>	114,043,256	360	1,559,188	12	316,786.822	566,468.339
Bytes Written <file=/mnt/cfs/scratch/jlinford/3/samarc/q000001.p3d>	114,043,256	360	1,559,188	12	316,786.822	566,468.339
Bytes Written <file=/mnt/cfs/scratch/jlinford/3/samarc/x000000.p3d>	91,231,888	181	1,247,352	12	504,043.58	531,178.218
Bytes Written <file=/mnt/cfs/scratch/jlinford/3/samarc/x000001.p3d>	91,231,888	181	1,247,352	12	504,043.58	531,178.218
Bytes Written <file=samarc/restore.000000/nodes.000032/proc.000000>	12,735,147	614	1,048,576	1	20,741.282	134,304.859
Bytes Written <file=samarc/restore.000001/nodes.000032/proc.000000>	12,735,147	614	1,048,576	1	20,741.282	134,304.859
Bytes Written <file=/mnt/cfs/scratch/jlinford/3/igbps.in>	482,132	2	482,120	12	241,066	241,054

TAU: ParaProf: Context Events for: node 0 - /Users/sameer/Downloads/sphere_np32_nsteps1_io.ppk

Name	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
Bytes Read	8,445,832	501	352,136	1	16,857.948	45,921.92
Bytes Read <file=/mnt/cfs/scratch/jlinford/3/nsu3d/subset.mcell.part.32/part.0>	3,004,352	17	352,136	8,192	176,726.588	118,915.498
Bytes Read <file=/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-inte	252,079	1	252,079	252,079	252,079	0
Bytes Read <file=/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-inte	228,653	1	228,653	228,653	228,653	0
Bytes Read <file=/mnt/cfs/scratch/jlinford/3/nsu3d/subset.mcell.part.32/parta.0>	2,405,012	38	196,300	7,056	63,289.789	41,633.172
Bytes Read <file=/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-inte	110,498	1	110,498	110,498	110,498	0
Bytes Read <file=/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/linux-redhat5.6-inte	88,256	1	88,256	88,256	88,256	0

Figure 10. I/O performed by MPI rank 0

5. Related Work

Debugging mixed language parallel programs that use Python is a daunting task. Commercial debuggers such as TotalView[16], DDT[17], and open source debuggers such as gdb[14] excel at generating backtraces for compiled executions. It is difficult if not impossible at this time to stop a program at a breakpoint, and move up or down the frames traversing Python and C boundaries in the same debugger, and examining and invoking Python routines and data structures. Python level entities are visible to performance evaluation tools that operate at the Python interpreter level. By merging the backtrace operations traditionally in the debugging domain with performance introspection, we create a hybrid tool capable of diagnosing fault information based on performance instrumentation. Extensive work has been done in the area of callstack unwinding and sampling based measurements in the DyninstAPI, TAU, and HPCToolkit projects[18].

To better understand the performance characteristics of an application, both profiling and tracing are relevant. While profiling shows summary statistics, tracing can reveal the temporal variation in application performance. Among tools that use the direct measurement approach, the VampirTrace[12] package provides a wrapper interposition library that can capture the traces of I/O operations using the Linux preloading scheme used in *tau_exec*. Scalasca[7] is a portable and scalable profiling and tracing system that can automate the detection of performance bottlenecks in message passing and shared memory programs. Like many other tools, including VampirTrace, it uses library wrapping for MPI. TAU may be configured to use Scalasca or VampirTrace internally. TAU, VampirTrace, and Scalasca use the PAPI[4] library to access hardware performance counters present on most modern processors. However, only the *tau_exec* scheme provides the level of integration of all sources of performance information—MPI, I/O, and memory—of interest to us, with the rich context provided by TAU. With this support, we can utilize the VampirServer[5] robust parallel trace visualization system to show the performance data through scalable timeline displays the state transitions of each process along a global timeline. Profile performance data can also be easily stored in the PerfDMF database[7]. TAU's profile browser, ParaProf, and its cross-experiment analysis and data-mining tool PerfExplorer[6] can interface with the performance database to help evaluate the scalability of an application.

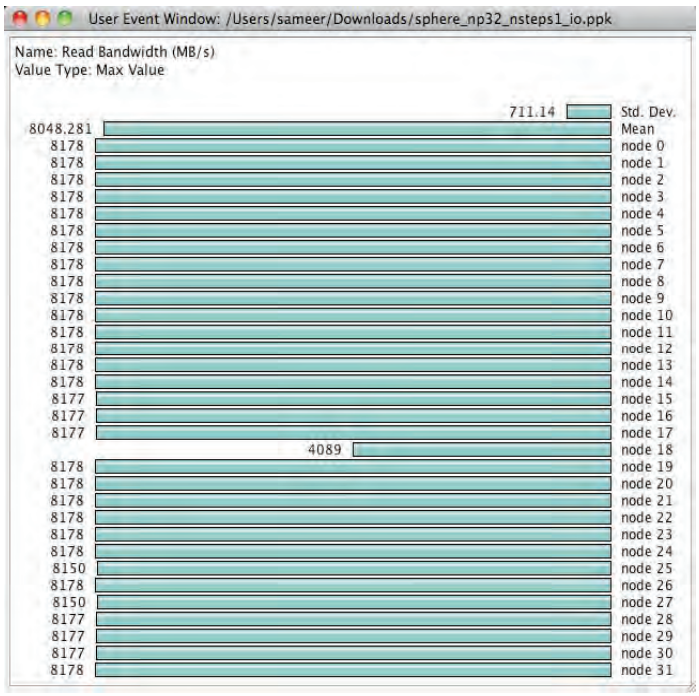


Figure 11. Peak read bandwidth by MPI rank

6. Significance to DoD

When application software experiences a runtime failure or performance problem, it is important for concise information about the error to be communicated to the development team. Particularly for multi-language applications that use a mix of Python, FORTRAN, C, and C++, current solutions are inadequate. This leaves a gap in communication between users experiencing bugs and/or performance issues and the code development team. This project delivers a tool that consolidates the execution data required for diagnostic purposes by utilizing powerful techniques for comprehensive measurement in the presence of execution errors. The goals of the project are twofold; first, to develop a runtime fault reporting tool to assist with debugging multi-language applications, and second, to close the loop with developers for more rapid turnaround of bug fixes. Absent any errors, the tool will report diagnostic information to users about the computational performance, memory usage, and IO. Such information is useful for users to understand the computational characteristics of their application and for planning their computing requirements.

The new tool addresses security concerns by avoiding the need for the user to provide the problem geometry and inputs to the development team diagnosing problems. The diagnostic file contains only runtime information so it is more easily exchangeable to members of the development team that may not have the requisite permissions to see the problem data. This is particularly important for the large amount of classified or proprietary work that takes place within the DoD.

The extensions to TAU described in this paper -simplified assessment of error diagnostics coupled with I/O and memory inspection for un-instrumented and instrumented applications expand the available capabilities, allowing users to ask questions such as:

- Where and when did the program experience an anomalous operation?
- What was the nature of the fault?
- What is the heap memory utilization in the application at the time of failure?
- Were there any memory leaks in the application?
- What was the level of nesting of the callstack?
- What was the routine name, source file name, line number and module name at the fault location?
- What were the performance characteristics of the application at that time?
- How much time did the application spend in I/O, and communication operations at the time of fault?

without re-compiling or re-linking the application and evaluate the fault diagnostics and performance of codes that use multiple languages such as Python, FORTRAN, C, and C++.

7. Conclusion

Modern scientific software requires software components written in different languages to interact with one another. For instance, software being developed by the CREATE air vehicles program involves high-level Python scripts executing lower-level C, C++, and FORTRAN90 software components. While this multi-language paradigm enhances the re-usability and extensibility of software, programming it is challenging due to a lack of debugging tools available for inter-language execution and memory leak analysis. Also, the software is intended to run on high-end parallel computer systems that demand a high level of sophistication from performance evaluation tools. In this paper, we describe a new diagnostic tool under development for multi-language applications. The tool, which builds on our previous efforts using TAU for performance and memory analysis, reports memory, and IO information as well as useful diagnostic call-stack information when the code experiences some form of anomalous operation, such as a segmentation fault. As CREATE software gets deployed for classified and proprietary projects within the DoD, users that experience bugs or anomalous behavior can provide details on the code execution through the diagnostic report generated by the tool, without revealing details about the application that the code is being applied to.

Acknowledgements

This work was supported by the DoD High Performance Computing Modernization Program (HPCMP) User Productivity Enhancement, Technology Transfer and Training (PETTT) program and through support provided by the DoD HPCMP to the HIARMS Institute and the CREATE program.

References

1. Wissink, A.M., V. Sankaran, B. Jayaraman, A. Datta, J. Sitaraman, M. Potsdam, S. Kamkar, D. Mavriplis, Z. Yang, R. Jain, J. Lim, and R. Strawn, "Capability Enhancements in Version 3 of the Helios High-Fidelity Rotorcraft Simulation Code." *AIAA-2012-0713*, AIAA 50th Aerospace Sciences Meeting, Nashville, TN, January 2012.

2. Morton, S.A., B. Tillman, D.R. McDaniel, D.R. Sears, and T.R. Tuckey, "Kestrel – A Fixed Wing Virtual Aircraft Product of the CREATE Program." *Proc. DoD HPCMP UGC 2009 Conference*, IEEE Computer Society, 2009.

3. Shende, S. and A.D. Malony, "The TAU Parallel Performance System." *Int'l Journal of High Performance Computing Applications*, SAGE Publishers, 20(2), pp. 287–311, http://tau.uoregon.edu, Summer 2006.

4. Browne, S., J. Dongarra, N. Garner, G. Ho, and P. Mucci, "A Portable Programming Interface for Performance Evaluation on Modern Processors." *Int'l Journal of High Performance Computing Applications*, 14(3), pp. 189–204, http://icl.cs.utk.edu/papi/, 2000.

5. Knupfer, A., R. Brendel, H. Brunst, H. Mix, and W. Nagel, "Introducing the Open Trace Format (OTF)." *Proc. ICCS 2006*, LNCS 3992, Springer, 2006.

6. Huck, K. and A. Malony, "PerfExplorer: A Performance Data Mining Framework for Large-Scale Parallel Computing." *Proc. ACM/IEEE Conference on Supercomputing (SC'05)*, 2005.

7. Huck, K., A. Malony, R. Bell, L. Li, and A. Morris, "PerfDMF: Design and Implementation of a Parallel Performance Data Management Framework." *Proc. ICPP 2005*, IEEE Computer Society, 2005.

8. Knupfer, A., H. Brunst, and W. Nagel, "High Performance Event Trace Visualization." *Proc. PDP 2005*, IEEE, http://www.vampir.eu, 2005.

9. Geimer, M., F. Wolf, B. Wylie, and B. Mohr, "Scalable Parallel Trace-Based Performance Analysis." *Proc. EuroPVM/MPI 2006*, LNCS 4192, Springer, pp. 303–312, http://www.scalasca.org, 2006.

10. SourceForge, "pyMPI: Putting the py in MPI." http://pympi.sourceforge.net, 2012.

11. Wissink, A. and S. Shende, "Performance Evaluation of the Multi-Language Helios Rotorcraft Simulation Software." *Proc. DoD HPCMP UGC 2007 Conference*, 2007.

12. Shende, S., A.D. Malony, A. Morris, and A. Wissink, "Simplifying Memory, I/O, and Communication Performance Assessment using TAU." *Proc. DoD HPCMP UGC 2010 Conference*, 2010.

13. *GNU Binutils*, http://www.gnu.org/software/binutils/, 2012.

14. *GDB – The GNU Project Debugger*, http://www.gnu.org/software/gdb/, 2012.

15. *Backtraces – GLIBC*, http://www.gnu.org/software/libc/manual/html_node/Backtraces.html#Backtraces, 2012.

16. *TotalView*, http://www.roguewave.com/products/totalview-family.aspx, 2012.

17. *DDT*, http://www.allinea.com/products/ddt/, 2012.

18. *DyninstAPI*, http://www.dyninst.org, 2012.

19. Malony, A.D., J. Mellor-Crummey, and S. Shende, "Measurement and Analysis of Parallel Program Performance using TAU and HPCToolkit," *Performance Tuning of Scientific Applications*, (eds. D. Bailey, R. Lucas, and S. Williams), CRC Press, pp. 49–86, 2010.

20. Shende, S., A.D. Malony, and A. Wissink, "A Fault Detecting Diagnostic Tool for Python-driven Multi-language Scientific Code." *Proc. DoD HPCMP UGC Conference*, IEEE Computer Society, 2012.

21. ParaTools, Inc., *PToolsRTE -Parallel Tools Runtime Environment*, http://www.paratools.com/ptoolsrte, 2012.

LWIR Background Signature Prediction for Very Large Scenes using the High-Fidelity Background Thermal Solver (HBTS)

Denise Talcott and Kyle Rokos
Signature Research, Inc., Calumet, MI
{talcott, rokos}@signatureresearchinc.com

Udaya Ranawake
High Performance Technologies Group, Dynamics Research Corporation, Andover, MA
uranawake@drc.com

Randal Cannon
US Army Missile Research Development and Engineering Center (AMRDEC), Redstone Arsenal, AL

Frank Carlen
US Army Aberdeen Test Center (ATC), Aberdeen Proving Ground, MD

Abstract

The state-of-the-art in long-wavelength infrared (LWIR) background signatures for very large terrestrial scenes is currently limited not by rendering technology, but by the large memory footprint and data dependencies in the heat-transfer algorithm used for physical temperature prediction. EOView is a physics-based, radiometrically-correct signature model which, at the time of first release in 2008, was processing scenes with around 100 million thermal nodes. The purpose of this project was to adapt the EOView code to accommodate much larger scenes for Department of Defense test and evaluation and research and development users who require physical temperature predictions to feed advanced scene-generation tools.

Algorithmic improvements introduced for this effort resulted in a decrease in memory usage and runtime of more than 50% for most scenes. The HBTS program expanded the scene-size supported by EOView temperature prediction to 500 million thermal nodes, exceeding the proposed metric by more than 150%. Efficiency and scalability were greatly improved, and a clear path forward was identified for computation on an even more massive scale.

This paper will describe the architectural and algorithmic changes implemented to achieve these goals, as well as how this program helped lead to generation of scenes with several billion polygons for government sponsors.

Keywords: thermal solver, heat transfer, scene generation, signature, modeling, simulation

1. Introduction

EOView is a large-scale, radiometrically-correct, physics-based background signature model that outputs hyper-spectral imagery for use in acquisition algorithm development and sensor/seeker performance assessment via simulation. The parallel scientific applications in *EOView* include a Thermal Solver, which solves the heat transfer portion of the problem, and the Renderer, which uses a ray tracing process to solve the rendering equation. *EOView* Version 2.2 (released in 2008) was demonstrated on scenarios with 100 million facets in high-fidelity terrestrial background scenes on the order of one to two square kilometers. Current simulation requirements call for scenes of 10 to 100 square kilometers and more. Given that these scenes are modeled with the same spatial complexity, this will result in orders-of-magnitude increases in overall scene facet counts. As demonstrated in the TRMC-funded MSP2 program (*EOView* 3.0), advances in rendering technology put hundreds of billions of facets well within our reach. In order to support this advanced rendering capability, we required a thermal solver capable of estimating physical temperatures for much larger background scenes.

Very large scenes present phenomenological, computational and logistical challenges to temperature prediction. The *EOView* Thermal Solver solves the transient heat equation using an implicit numerical technique over a user-defined period of time, generally a diurnal cycle. The implementation uses an iterative process with an outer loop over time-steps and an inner-loop over convergence within each time-step. During the thermal solver computation, processes exchange

temperatures and perform other synchronization via MPI (Message Passing Interface). Each process is also multi-threaded, with each thread using shared scene data and working independently on a collection of facets. The numerical solution imposes data dependencies and synchronization overhead that make load-balancing a challenge. In addition, memory required to store the temperatures, view factors and shadow factors required for the heat transfer prediction is a function of the number of facets in the scene and can quickly become a limiting factor. At the outset of this program these were the major challenges to larger scale scene generation.

The goal of this project was to address these challenges and provide the Department of Defense (DoD) test and evaluation (T&E) community with the capability to predict accurate background thermal signatures for environments and scenarios much larger and more complex than was possible at the time. High performance computers were a necessary component in the solution. The enhanced code developed for this program is referred to here as $EOView_{\mathrm{HBTS}}$ and is now fully-integrated into *EOView*.

2. Development of the HBTS

Scene size refers to both the spatial extent of the scene and the number of facets used to represent it. Increased complexity can be a result of such things as moving objects or strong radiation sources (e.g., exhaust plumes). An important criterion for this effort was that code modifications would not compromise fidelity in the *EOView* temperature prediction. Most code changes made under this program were related to:

- Reduced memory usage, with a goal of 50%
- Reduced data dependencies, with a goal of 10%
- More effective load-balancing
- Reduction of communication bottlenecks
- Support for increased scenario complexity

2.1 The Memory/Computation Trade-off

Architectural changes introduced for this program represented tradeoffs between memory, computation and communication. Preference was given to modifications that reduced memory, provided that computational overhead did not increase dramatically. A major component of this was a move away from conventional data structures and pointers to data in packed binary form that would be un-packed as needed during the computation. Instancing of scene objects was added to reduce the scene geometry footprint. These changes reduced memory sufficiently to allow us to store the full-scene geometry on each compute node rather than partitioning it, which in turn simplified the view factor calculation process, improving load-balancing and reducing communication overhead. The acceleration structure used in the view factor calculation was modified to handle much larger data sets and for more efficient ray traversal. These changes, along with the need to reduce the disk footprint and input/output (I/O) cost, led to computing view factors on-the-fly rather than pre-computing and storing them. On-the-fly computation also readily supports dynamic events such as moving objects.

2.2 Load-Balancing with a Modified Job Allocation Strategy

Another major change to the code was in the re-definition of ‘jobs’ and job ‘chunks’. Job chunks are groups of facets to be processed by a single thread. In the baseline code, job chunks corresponded to terrain and background objects (e.g., trees, shrubs or buildings), confounding the load-balancing process as the size of these objects varied. The biggest difficulty was presented by terrain data which, when necessary, was broken up prior to run-time in a manner not amenable to changing processing configurations or load-balancing.

The modified approach was to compute a chunk size at run-time based on the processing configuration (e.g., number of threads and processes) and to allocate jobs based on these chunks. This introduced some additional complexity, as job chunks were no longer neatly defined as a single object (and in fact a single job chunk could span multiple objects); however, the result was a much simpler and more natural way of balancing the work load among processors and threads. Load-balancing (along with increased multi-threading) is a major driver of reduced run-times relative to the baseline code.

2.3 Region of Influence Implementation

A region of influence (ROI) is the area around a facet within which it is expected to exchange energy. Limiting the ROI is one practical way to limit facet dependencies and is a reasonable concept when considering very large scenes. In the baseline code, no limitation is placed on a facet ROI. In order to explore the impact of limiting ROI, the modified code used a simple implementation based on range. When view factor intersection tests were truncated because of the ROI limitation, a simple approach was used to account for the solid angle it represented. In this case, we considered that the solid angle was exposed to sky temperature if the associated ray was directed above the horizon and air temperature if the ray was directed at or below the horizon.

The changes described above are manifested in differences in performance patterns in the code. The chart in Figure 1 provides an overview of the memory, network and I/O, load-balancing, and computational effects of the algorithmic and data structure changes implemented for this project. In each category, a '-1' represents a penalty, a '0' represents a neutral change and a '1' represents an advantage.

Algorithmic Feature	Memory Usage	Network, I/O	Load Balancing	Computation
Thermal object instancing	+1	0	+1	-1
Packable temperatures	+1	0	0	-1
Modified job chunks	0	0	+1	0
Multi-threading	+1	+1	0	+1
On-the-fly computation	+1	+1	0	-1
Eliminate spatial partitioning	-1	+1	+1	+1
Acceleration Structure	0	0	0	+1
Region of Influence	0	+1	+1	+1
Terrain representation	+1	0	+1	0

Figure 1. Algorithmic feature performance impact

3. Test Overview

The results of tests done for this program illustrate the impact of algorithmic changes and inform a path forward to temperature prediction on a much larger scale. For these tests, the Signature Research *Kaleva* cluster was used in addition to the ARL *Harold* cluster, where a comparison to the baseline code was required because the baseline code was not ported to Harold. The two systems are described in Figure 2, with the most notable difference being the network: Infiniband on Harold and gigabit ethernet on Kaleva. Where appropriate, results from both Kaleva and Harold were included to confirm that code behavior and trends were consistent. Tests on larger scenes, including scalability tests, were done on Harold. The results of fidelity tests are not included here, but all results were within the required tolerance.

	Harold	**Kaleva**
Manufacturer	SGI	Dell
Model	Altix ICE 8200	n/a
Theoretical Peak	109.3 TFlops	n/a
Cores	10,752 2.8 GHz Intel Nehalem	128 2.66 GHz Intel Xeon E5430
Memory	32 TB	384 GB
Local System Disk	600 TB	No local disks
Operating System	SLES 10 SPA	Centos Release 5
Internal Network	4X DDR Infiniband	Gigabit Ethernet

Figure 2. Overview of two clusters used for testing

4. Memory, Run-time Reduction, and Performance Trends

It was expected that memory usage patterns would vary with scene size and density: terrain imposes a disproportionate memory overhead because the facets in the terrain aren't instanced, and because multiple-layer temperatures must be stored for each terrain facet. For this series of tests, each test scene consisted of a ground plane and objects such as trees and other vegetation totaling between 1 and 100 million facets. Two variations of the 50 and 80 million facet scenes with different object densities were included. In the first case, the ground plane was a 100×100 meter area (higher density). In the second case, the ground plane was a 1,000×1,000 meter area (lower density).

Figure 3 and Figure 4 show a reduction of between 60% and 95% in the peak *per node* memory usage for scenes of 100×100 meters and 1,000×1,000 meters. The significant improvement seen in even the low-density scenes is due to a more efficient terrain representation in the modified code. Similar gains were seen in peak *total* memory usage (not shown here), with a reduction of over 50% for all scenes, and 80% in many cases. The small differences in memory usage between the two clusters were not precisely identified but are likely related to system differences (e.g., MPI[1] implementation, operating system, standard library) and/or load variations.

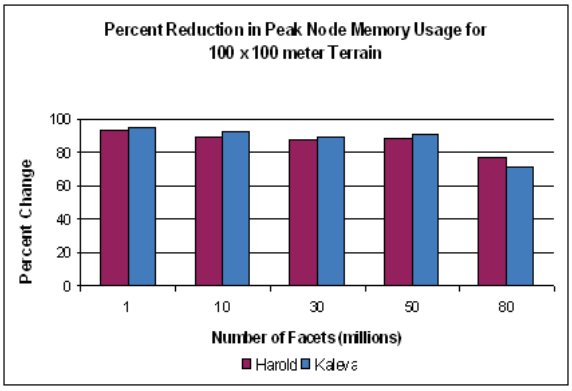


Figure 3. On 16 compute nodes, peak memory per node was reduced by more than 70% for scenes with increasing density

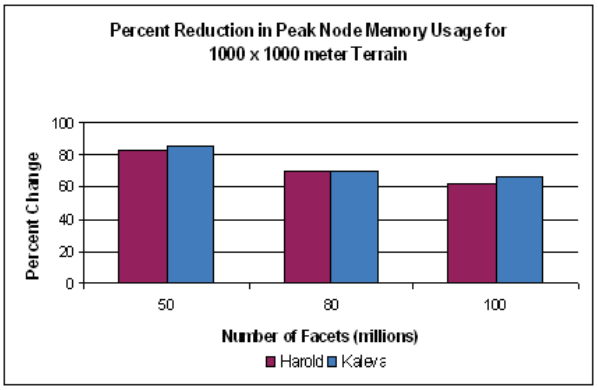


Figure 4. On 16 compute nodes, peak memory per node was reduced by 60% to 80% for scenes with larger terrain and lower density

A second set of tests took a closer look at per facet memory and code behavior when using different numbers of threads and compute nodes; in this case up to 128 threads on 16 nodes on the Harold cluster. Again, scenes of different densities were used. These included three scenes with no objects (the extreme low density case) and higher density scenes with 52% and 69.5% object facets. As shown in Figure 5 and Figure 6, the memory required to store the full scene geometry on each node caused per facet memory to increase as nodes were added. We also see per facet memory decrease in Figure 6 when objects are added to the scene due to the advantage gained by instancing.

In contrast, adding nodes lowers *per node* memory usage because the bulk of memory is consumed by view factors and terrain layer temperatures, and these are amortized over compute nodes. Figure 7 and Figure 8 show per node memory decreasing as the number of nodes increases. For these small scenes and small numbers of compute nodes, the trends are promising.

Trends for small scenes can be deceiving; however, as the relative size of constant (non-amortized) overhead to view factor and layer temperature data changes for larger scenes. In the 200 million facets case executed on Harold and shown in Figure 9, little per node memory reduction is seen between processing configurations of 128 and 256 nodes because a high-percentage of the memory consumed on each node is constant data.

[1] The SGI MPI implementation was used on *Harold* while OpenMPI was used on *Kaleva*.

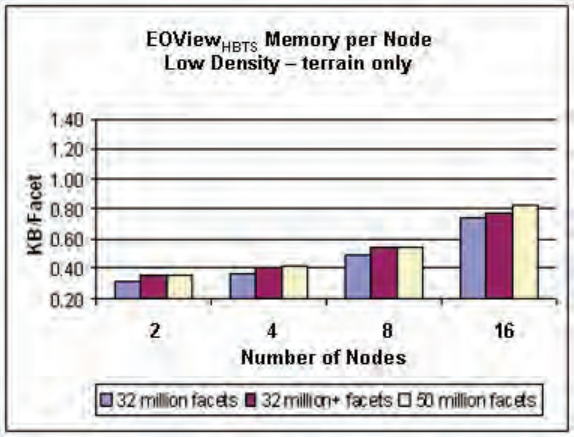


Figure 5. Memory per facet increases as node-count increases (Harold cluster)

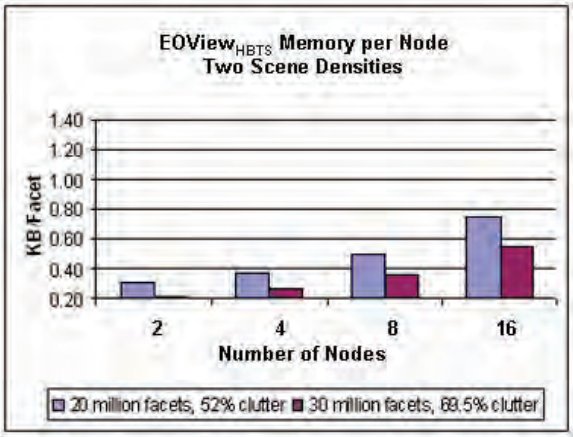


Figure 6. Memory per facet can be less for higher density scenes (Harold cluster)

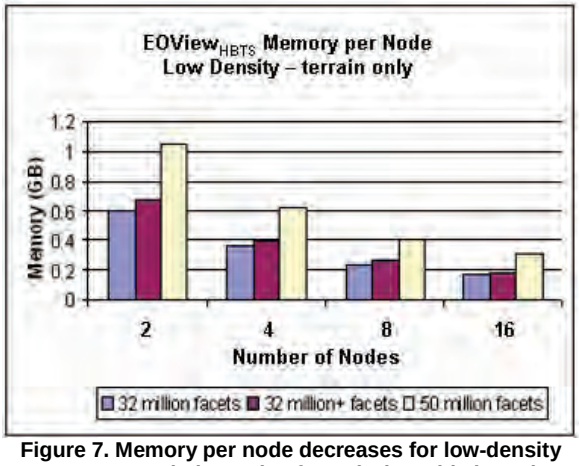


Figure 7. Memory per node decreases for low-density scenes consisting only of terrain (Harold cluster)

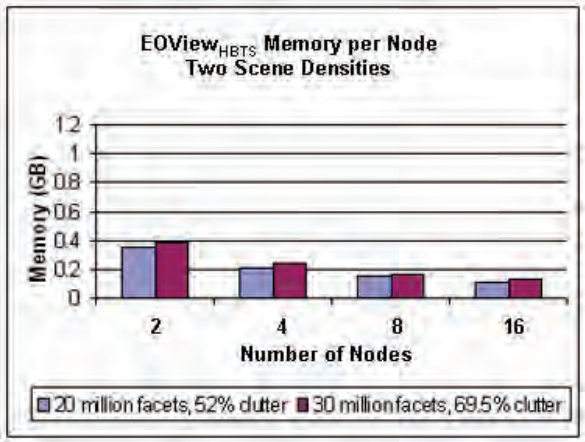


Figure 8. Memory per node decreases for scenes with higher facet densities (Harold cluster)

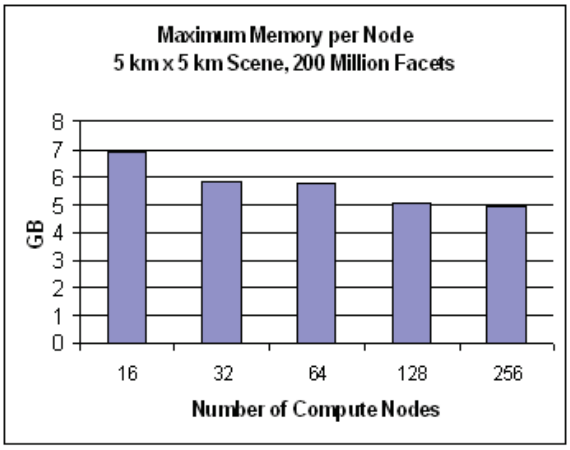


Figure 9. At 256 nodes and 200 million facets the large constant overhead on each compute node dominates memory usage. (Harold cluster)

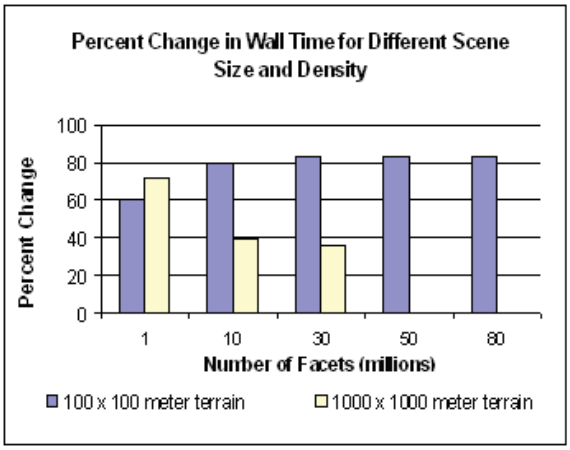


Figure 10. Wall-time is reduced by over 50% for higher density scenes, less for lower density scenes, over the baseline code when executed on 16 compute nodes on the Kaleva cluster. (Only three scenes were included for the larger terrain.)

As shown in Figure 10, wall-time decreased as well, by 60% to 80% for higher density scenes, and 30% to 70% for lower density scenes. In this example, all scenes were executed on 16 compute nodes on the Kaleva cluster. The primary reasons for the reduction in wall-time are 1) better use of multi-threading, 2) elimination of spatial partitioning, and 3) more efficient acceleration structure. Figure 11 shows the standard deviation in memory usage for the baseline code and the modified code (also on the Kaleva cluster with 16 nodes) for scenes of 50, 80, and 100 million facets and of different densities. Memory usage is also a reasonable proxy for load, as the amortized portion of the memory footprint on each node is a function of the number of facets being solved. The labels on the horizontal axis in Figure 11 indicate the code used (BL = baseline and HBTS = modified code), the number of facets in each scene and the scene dimensions.

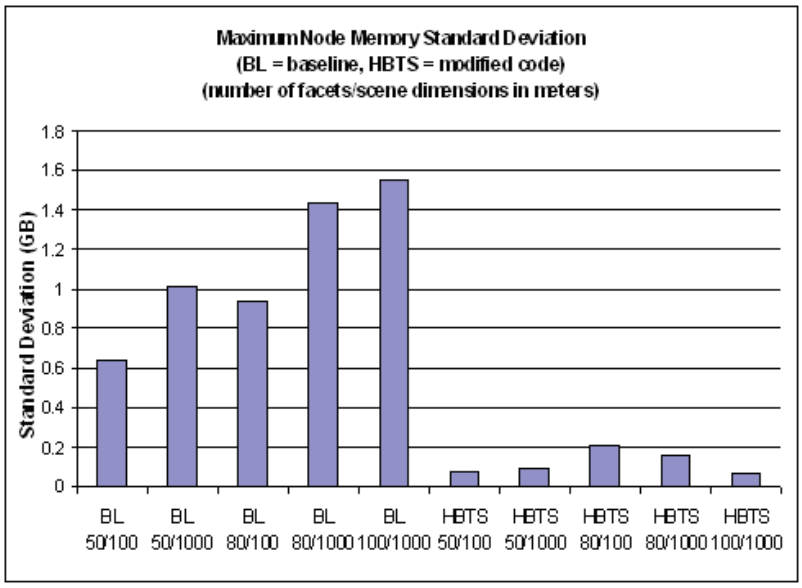


Figure 11. The memory standard deviation is an indicator of load balance in the thermal solver

5. Performance Trends in Larger Scenes

As seen above, characteristics such as scene density play a large role in thermal solver computing performance. Other features such as dynamic events affect performance as well. For example, time-step size is, in part, a function of how quickly scene temperatures are expected to change. Solve functions, temperature synchronization and convergence synchronization must be done at each iteration within each time-step. View factors may need to be re-computed as scene elements (e.g., vehicles) move. Shadow factors must be re-computed as the solar angle changes. An additional set of tests considered larger scenes and a break-down of the calculation into these major tasks to assess how well the code would perform in more complex, dynamic scenarios. These tests were done on the Harold cluster.

Figure 12 and Figure 13 breakdown the average per facet iteration time between times spent executing the solver function ("solveTime"), time spent on convergence tasks ("convergeSync"), and temperature synchronization ("tempSync"). One striking observation was how much faster, per facet, the simulation ran on the largest scene. Another observation is that the temperature synchronization is taking longer, per facet, than the solve-time. The most telling result is that the amount of time spent on convergence synchronization varies from scene-to-scene. The convergence synchronization timer starts when the first compute node begins waiting for synchronization, and stops when the last compute node has sent its results. The MPI transaction involved in this step should take the same amount of time for runs using the same number of processing nodes. The overall conclusion is that the urban scene run is more load balanced than the others. View factor and shadow factor compute times in Figure 14 and Figure 15, where times were lowest for the largest scene, are consistent with this conclusion.

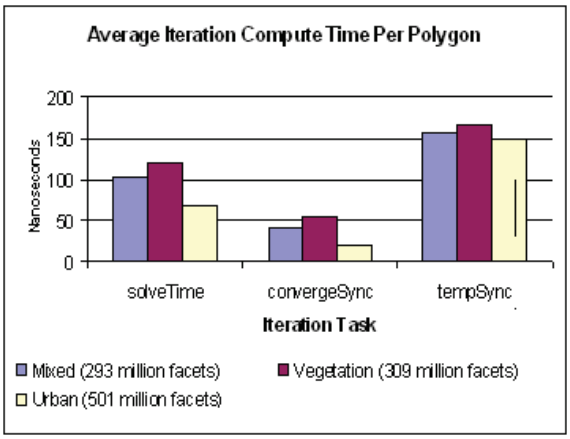


Figure 12. A difference in convergence synchronization time indicates differences in load balancing

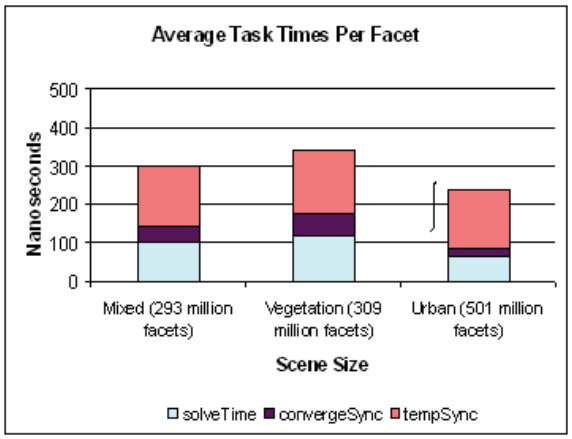


Figure 13. Both solve-time and convergence synchronization are less for the largest scene, indicating better load balancing

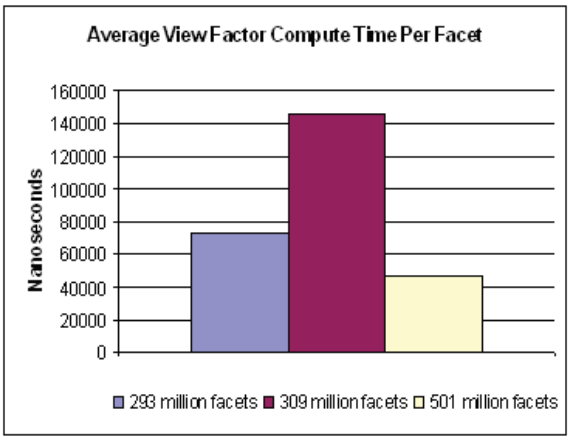


Figure 14. Relative per facet view factor calculation times are consistent with differences in load balance

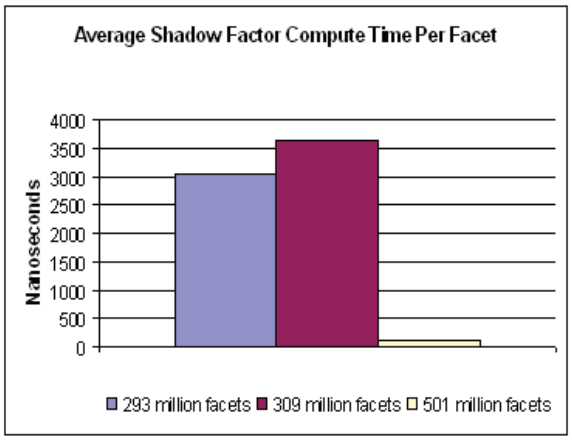


Figure 15. Relative shadow factor calculation times are consistent with differences in load balance

6. Investigation into Scalability

It is clear from previous results that the behavior of the code changes as data sets become larger, and as more compute nodes are added. For this investigation we looked at the contribution of each thermal solver task to overall run-time and which task(s) might pose scalability issues when executed on larger numbers of compute nodes. Tests were done on the Harold cluster with processing configurations of 16 to 256 compute nodes using 128 to 2,048 cores. Two scenes were used consisting of a 1 km×1 km area with 100 million facets (higher density) and a 5 km×5 km area with 200 million facets (lower density). Timing results shown below are averages over all iterations for each run.

The results for both scene densities followed the same trends (results for the larger, lower density case are shown here). Run-times shown in Figure 16 for the view factor and shadow factor calculation reflect the fact that there is no communication overhead for these tasks. As a result, we expect run-times to decrease proportionately as the number of compute nodes increases.

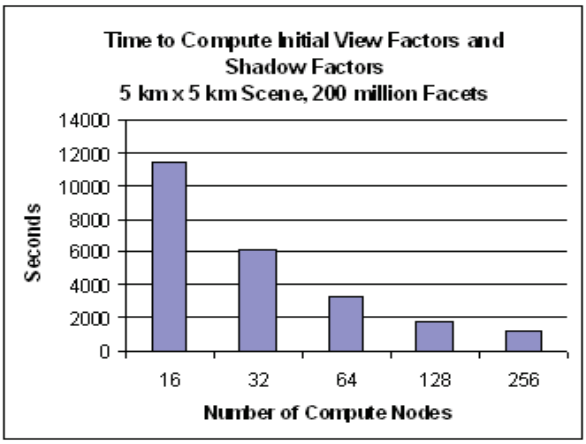


Figure 16. Time to compute initial view factors and shadow factors scales well: decreasing proportionately as the number of compute nodes increases

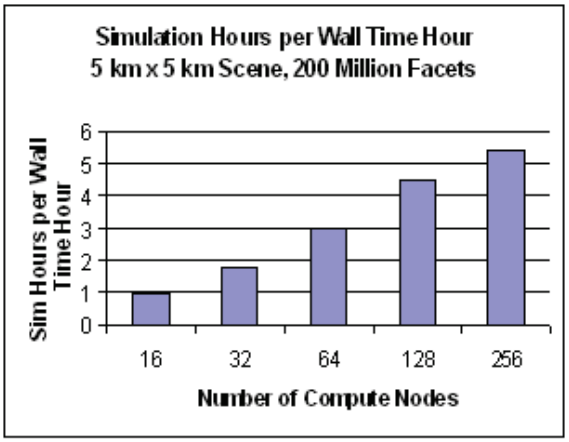


Figure 17. Simulation hours per wall-time hour increases initially; improvement becomes marginal after 128 nodes

The number of simulation hours per wall-hour shown in Figure 17, where adding more than 128 compute nodes provides only marginal improvement in wall-time, reveals that there are scaling issues related to synchronization. Figure 18 expresses this as an *efficiency* metric: simulation hours per wall-time hour per compute node decreases steadily until the efficiency at 256 nodes is about one-third of the efficiency for 16 nodes. The portion of each iteration time consumed by temperature synchronization in Figure 19 supports this. When 64 compute nodes are used, and thereafter, synchronization time dominated each iteration of the solver. For the 100 million facet case (not shown), *overall* iteration time actually increased due to this overhead when moving from 128 to 256 nodes. The trend shown in the 200 million facet case suggests the same behavior will be seen when more compute nodes are added. This overhead may be a result of the actual time spent exchanging temperatures, insufficient load-balancing, or both.

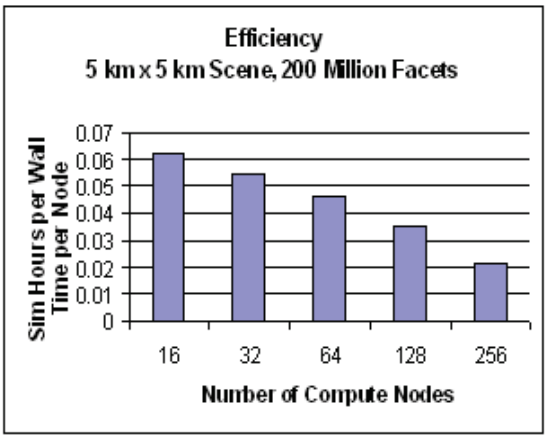


Figure 18. As compute nodes are added efficiency begins to fall off due to synchronization overhead

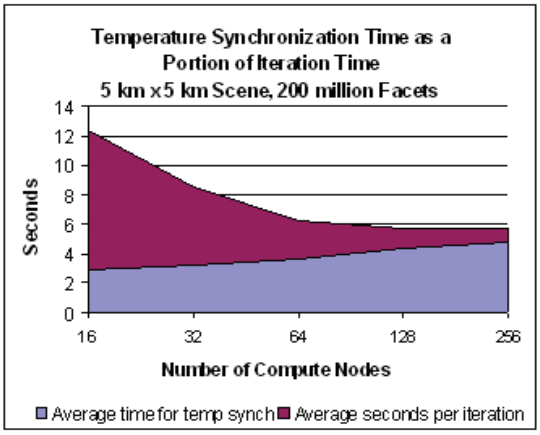


Figure 19. Synchronization overhead increases with the number of compute nodes

7. Data Dependency Reduction Using Region of Influence

Efforts to improve memory and processing were successful, but scaling will remain an issue as long as synchronization is imposed on a global level. The only way to scale the heat transfer algorithm on a massive level is to find some reasonable separability to reduce synchronization. As a first step (to be followed later by more sophisticated approaches), a series of tests were conducted to evaluate the effect of limiting facet ROI on the number of data dependencies and on the temperature prediction.

Figure 20 lists the five scenes used for these tests, representing various combinations of urban and non-urban environments. A target model was included in two scenes to look at the impact of ROI limitations on target signature. Comparison of the temperature prediction was done by inspecting the mean temperature in selected areas of interest (AOI) in the scene on objects such as buildings, soils, vegetation, and target facets. In each case, the delta T temperature difference and percent dependency reduction were recorded for the limited ROI compared to an unlimited ROI.

Test Scene	Type	Area	# Objects	# Facets
Urban	High density, urban	5 km x 5 km	7,626	501 million
Industrial	Low density urban	1 km x 1 km	214	40 million
Mixed	Low density, primarily vegetation	5 km x 5 km	7,038	293 million
Test range	Mixed density, vegetation	860 m x 590 m	738	74.5 million
Vegetation	Low density, vegetation	5 km x 5 km	3,345	309 million

Figure 20. A variety of scenes types and sizes were used for region of influence tests

Figure 21 shows that the scenes exhibited very different behavior with respect to ROI limitations. The industrial scene had far fewer dependencies than the other scenes, and demonstrated a much larger proportional reduction. The industrial scene was the only case in which a 50 meter ROI produced a 10% reduction in dependencies. The 10% goal was reached for the other scenes only when the ROI was limited to 10 meters. The impact of ROI limitation on the temperature prediction is similarly diverse, exceeding 10 degrees for the industrial scene, even for the least restrictive ROI. These investigations led to successful follow-on efforts using more sophisticated approaches to dependency reduction.

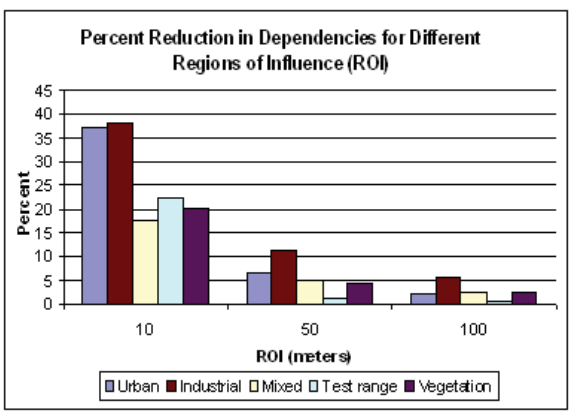


Figure 21. Only the 10 meter ROI limitation consistently reduced dependencies by 10%

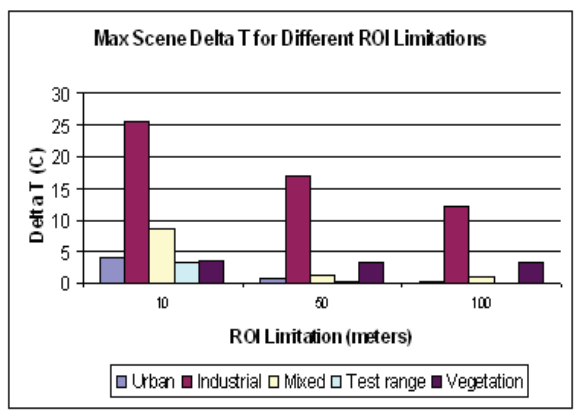


Figure 22. The low density industrial scene is the most sensitive to ROI limitation

8. Conclusion

This project significantly expanded temperature prediction capabilities and informed the path to even larger and more complex scenes. All success criteria were met while maintaining fidelity. The scene size that was achieved ultimately exceeded the goal by more than 150%. Figure 23 provides a summary of increased capabilities as a result of this program.

	Goal	Actual
Reduction in per node memory usage	50%	65% to 95%
Reduction in node idle time	50%	Over 63% for large scenes of typical density
Dependency reduction	10%	Varied, up to 35% (scene dependant)
Increased scene size	200 million facets	501 million facets
Reduction in run time	n/a	34% to 77%

Figure 23. Performance improvements in *EOView* as a result of this program

In addition, the following were important accomplishments:

- A plan for scaling to massively-larger scenes on HPC systems via a 'divide and conquer' approach was identified and subsequently implemented under funding from government sponsors.
- The architecture is now much more suited to dynamic scenarios
- Advances made under this program enabled a proof-of-principal demonstration for a large Department of Defense (DoD) program in April 2011. This was followed by a major milestone in spring of 2012 when *EOView* was used to compute physical temperatures for a scene of about 11 km×17 km and 16 billion polygons as part of a program deliverable for government sponsors.

Acknowledgement

This work was done under funding from the High Performance Computing Modernization Program (HPCMP) User Productivity Enhancement, Technology Transfer, and Training (PETTT) Program.

6.3 Advanced Technology Development

Navy

Towards Prediction of Hydro-Acoustics using Large Eddy Simulations

Peter A. Chang III, Michael D. Mattson, Joseph Slomski, and Hua Shan
US Naval Surface Warfare Center, Carderock Division, Computational Hydromechanics Division (NSWCCD), West Bethesda, MD
{peter.chang, michael.d.mattson, joseph.slomski, hua.sha}@navy.mil

Krishan Mahesh
Department of Aerospace Engineering, University of Minnesota, Minneapolis, MN
mahesh@aem.umn.edu

Abstract

In this paper we describe two examples of the use of incompressible Large-Eddy Simulations (LES) to provide data for hydro-acoustic evaluations. LES temporally and spatially resolves the energy-containing scales of turbulence, meaning that it is the appropriate methodology for hydro-acoustics, where fidelity over a range of frequencies is necessary. HPC resources are invaluable for providing the large number of cores and CPU time, post-processing tools, high data-transfer bandwidth, and archive storage space necessary for running these simulations and processing enormous amounts of data.

If the trailing-edge of a hydrofoil is sufficiently blunt, the flow separates and Kelvin-Helmholtz instabilities cause vortex shedding. The fluctuating wall pressures on the trailing-edge surfaces are the cause of narrow-band dipole hydro-acoustic noise. Using incompressible LES, we would like to predict whether shedding occurs, and if it does occur, the strength of the shedding, as measured by its peak in the wall pressure power spectral density (PSD). Problems of Navy interest necessitate resolving the flow over the entire chord, in order that relatively weak flow separations, and vortex shedding, are predicted accurately. This means that problem sizes for trailing-edge vortex shedding are quite large, typically 20 to 50 million control volumes, require 512 or more processors and at least a week of computing time. In this paper we show preliminary validation results for a symmetric airfoil at chord-based Reynolds number, $Re_c=1.2\times10^6$. We show that the developing attached boundary-layers compare well with the literature.

The turbulent boundary-layer flow over a forward-facing step (FFS) can also be the cause of radiated noise. We have developed a methodology where the far-field radiated noise power spectral densities (PSD) are predicted using the Ffowcs Williams-Hawkings solution to Lighthill's equation assuming stationary, acoustically-compact source surfaces, where the local force intensities are computed using incompressible LES. A spectral velocity field perturbation method provides an incipient boundary-layer with resolved turbulent fluctuations. In this paper, we show an FFS validation for a Reynolds number, $Re_h=21{,}904$ (based on step-height and free-stream velocity). The numerical predictions of fluctuating wall pressure and radiated noise PSD are in good agreement with experimental measurements.

1. Introduction

Broadband hydro-acoustic sources, associated with a boundary-layer, are diffracted by surface discontinuities, generating dipole noise, which are the cause of narrow-band radiated noise. Two examples of discontinuities are flow over an airfoil with a beveled trailing-edge and the flow over a forward-facing step.

A. Trailing-Edge Vortex Shedding

If the trailing-edge (TE) of a hydrofoil is sufficiently blunt, the flow separates and Kelvin-Helmholtz instabilities give rise to a vortex street in the wake. Trailing-edge vortex shedding, or TEVS, causes periodic pressure fluctuations on the trailing-edge surfaces, which in turn excite structural vibrations. The radiated noise frequency spectra due to wall pressure fluctuations can be shown to be related to the convolution of the wall pressure frequency spectrum and the structural mode shape functions.[1] We consider hydrofoils at design condition, with small angles-of-attack, where the

flow over the foil, up to the trailing-edge is fully-attached. Only at the trailing-edge is the flow separated. It has been shown in Chang, et al.[2,3] that the resolution of the upstream boundary-layer is necessary for the accurate prediction of vortex shedding. Therefore, fully-resolved large-eddy simulations (LES), where the flow over the upstream boundary-layer, as well as in the trailing-edge region is resolved, are necessary. Lower-fidelity methods, such a Reynolds-averaged Navier-Stokes (RANS) and detached-eddy simulations (DES) cannot accurately predict vortex shedding signatures because they do not resolve the upstream boundary-layers. Chang, et al.[4] performed simulations over a small submersible showing fully-resolved LES can accurately predict the evolution of spatially-developing attached boundary-layers.

LES simulations for TEVS for chord Reynolds numbers over 1×10^6 can be computationally expensive, and early work was limited to lower Reynolds numbers. Oberai, Roknaldin and Hughes[5] computed the trailing-edge noise for an Eppler 387 airfoil at the moderate Reynolds number of 1×10^5 with transitional boundary-layers. Wang, Moreau, Iaccarino and Roger[6] performed an LES of a non-vortex shedding, cambered airfoil at $Re_c=1.5\times10^5$ showing good comparison with experimental wall pressure spectra.

In order to simulate higher Reynolds numbers, one approach has been to perform LES of just the aft 40% of a flat strut utilizing flat plate boundary-layer simulations to provide the time-dependent turbulent inflow boundary conditions and Reynolds-averaged Navier-Stokes (RANS) methods to obtain the correct global flow and pressure distribution[7–9] The possibilities of using this method for predicting vortex shedding were investigated by Wang[9], where an LES of a flat strut with a 45° beveled TE at $Re_c=1.9\times10^6$ was performed; results showed clear evidence of vortex shedding in the wake and predicted sound pressure spectra that compares well with experiments.

Some of the first attempts at LES of entire airfoils at higher Reynolds numbers have been reported by Chang, Wang, and Gershfeld.[2] They performed LES on a NACA 0016 foil with a 41° trailing-edge apex angle and a slightly convex suction-side. They report on simulations performed at $Re_c=1.2\times10^6$ and $Re_c=3.2\times10^6$. The convex suction-side has the effect of diffusing the pressure-side boundary-layer, weakening the vortex shedding. In fact, the far-field radiated noise measurements of Gershfeld, Blake and Knisely[10] show a vortex shedding peak of only a few decibels. Thus, this geometry provides a very rigorous test case. Gershfeld, et al.[10] also measured the WPFs in the suction-side TE region. While they did not obtain similar measurements on the pressure-side, these can be inferred from the far-field radiated noise. These calculations were performed using a high-fidelity C-grid LES code that was parallelized using OpenMP. The code had a maximum speedup factor of four, and simulations where the flow traversed only a few chord lengths took six months using an SGI Altix cluster circa 2005.

In the first part of this paper, we discuss TEVS LES computations that have been performed using an LES code which has been highly-parallelized. Simulations using 512 processors take approximately one day per chord length, and a little over a week to complete a simulation. With this type of turnaround-time these LES computations can be used for design evaluation. We show some preliminary results on a NACA 0030 airfoil with a symmetric, beveled, trailing-edge demonstrating that the flows over the foil compare reasonably well with the literature and resolve the necessary turbulence structures important for accurate prediction of vortex shedding.

B. Forward-Facing Step

The noise caused by turbulent flow over a forward-facing step is a problem that is quite representative of flow noise-generating mechanisms in engineering practice. Surface unevenness, for example, in the form of a panel edge that protrudes into the oncoming turbulent boundary-layer flow, can be a significant source of flow-induced noise for aircraft, automobiles, and marine vehicles. Such a protruding panel-edge resembles a step which faces forward into the oncoming turbulent boundary-layer. In addition, a gap can typically occur between the panel-edge and the surface that surrounds or adjoins it. Experimental investigations of forward-facing step flows at low Mach number by Farabee and Casarella[11,12] have characterized the wall-pressure fluctuations accompanying these flows. Further experimental studies by Farabee and Zoccola[13] and Catlett, et al.[14] have characterized the radiated sound associated with such flows.

For prediction of far-field radiated noise, we employ a hybrid method where velocity field sources are computed using an incompressible LES, and then Lighthill's acoustic analogy[15] is utilized to obtain the far-field acoustic pressure. In this hybrid approach, the flow and sound computations are decoupled, implying that the acoustic pressure does not need to be numerically-resolved at every point in the flow field. Rather, it can be calculated at arbitrary far-field points, either numerically or in integral form, using just the fluid dynamic information obtained from the incompressible turbulent flow adjacent to the surface in question. The sound is assumed to radiate directly from the source or scattered wave region to the far-field. This is a relatively economic approach to predicting flow-induced noise, since it dispenses with the need for high-resolution techniques required to resolve acoustic waves as they propagate from the turbulent flow region.

In the second part of this paper, we discuss the mathematical formulations, and then we show the computational grid, and results for the forward-facing step problem.

2. Trailing-Edge Vortex Shedding

A. MPCUGLES Large-Eddy Simulation Code

The numerical methods used to solve the RANS equations are not directly applicable to LES. RANS typically either uses up-winded numerical methods or central-differencing with a high-order dissipation term. Numerical dissipation makes the solution procedure robust. However, when used for LES, numerical dissipation compromises accuracy,[16] since it competes with, and often overwhelms the physical dissipation that the sub-grid model is trying to provide.

By definition, the dissipative scales are not resolved by the grid in an LES. In practice, this leads to numerical instability if straight-forward non-dissipative central-difference schemes are used. One solution to this problem is to develop non-dissipative numerical schemes that discretely conserve not only first-order quantities such as momentum, but also second-order quantities such as kinetic energy. Discrete energy conservation ensures that the flux of kinetic energy only has contributions from the boundary faces. This makes the solution robust without the use of numerical dissipation. Note that satisfying one constraint discretely, does not ensure the other—both constraints have to be simultaneously enforced when deriving the algorithm. The Harlow-Welch algorithm[17] possesses this property on structured grids, and has therefore been widely used for LES on structured grids in simple geometries.

Mahesh, et al.[17] have developed an algorithm, "MPCUGLES," for LES that is discretely energy-conserving on unstructured grids with arbitrary elements. Also, a novel least-squares formulation is employed for enforcing continuity, and is essential for obtaining solutions on highly-skewed grids. Skewness in this context refers to the amount of distortion or stretching in the grid elements. The algorithm has been validated for a variety of flows including the temporal decay of isotropic turbulence (DIT), flow over a circular cylinder, flow in a coaxial combustor, flow in a gas turbine combustor,[18] and initial crash-back validations.[19,20] The DIT results are very important, as they show that the code has the minimally-dissipative numerical scheme that can give the correct Reynolds number dependence, i.e., when increasing the Reynolds number of the turbulence while keeping the grid density constant, the temporal decay rate without a sub-grid scale stress (SGS) model goes to zero as the grid no longer supports the increasingly smaller scales of turbulence responsible for decay. The practical ramification of this result is that relatively few grid points may be necessary to initiate and maintain turbulent vortices, possibly allowing for crash-back simulations to be run with relatively coarse grids.

In MPCUGLES, the SGS stresses are parameterized using a dynamic Smagorinsky SGS model. The pressure Poisson equation is solved using the algebraic multi-grid (AMG) code BoomerAMG from the hypre library (version 2.2) developed at Lawrence Livermore National Laboratory or a conjugate gradient (CG) technique. BoomerAMG and CG have been parallelized and have been found to be scalable on massively-parallel computers.[20a] The code has been parallelized with the Message Passing Interface application programmer interface (API) library specification using METIS, developed at the University of Minnesota for automated, load-balanced partitioning of computational domains consisting of large numbers of unstructured elements. Mesh and data files are written in netCDF a machine-independent, self-describing, binary data format standard.

We are using MPCUGLES to predict trailing-edge vortex shedding over hydrofoil sections. These sections are at a low angle-of-attack to the flow, and therefore the flow is attached except for at the trailing-edge. Therefore, the flow over the entire chord length needs to be resolved. For even model-scale Reynolds numbers, these result in very large problem sizes. Fully-resolved LES simulations are being used because it has been shown by Chang, et al.[3] that the turbulence structures in the wake are highly dependent upon the resolution of upstream wall-bounded flows.

B. Computational Grid and Run Conditions

The airfoil is a modified NACA 0030, which is extruded in the span-wise (z) direction to obtain a three-dimensional (3D) domain from a two-dimensional (2D) airfoil shape. The airfoil dimensions are shown in Figure 1, with x the stream-wise direction. The leading- and trailing-edge sections are cut from a NACA 0030 airfoil section, with a flat section sandwiched between them. The origin of the computational domain is located at the airfoil trailing-edge. The airfoil thickness h is $0.05833C$, where C is the chord length. The leading-edge (l_{LE}) and trailing-edge (l_{TE}) stream-wise lengths are $0.05833C$ and $0.14167C$, respectively. The angle Φ of the trailing-edge from the horizontal plane, defined as $\tan(\Phi)=h/(2l_{TE})$, is 11.6°.

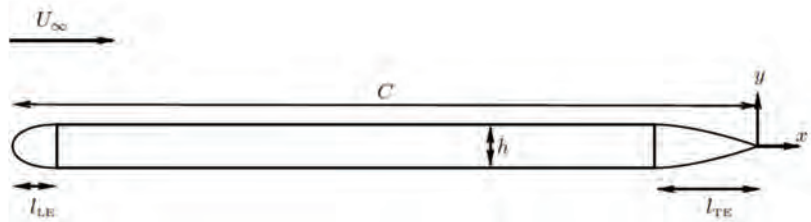


Figure 1. Schematic of the modified NACA 0030 airfoil

Figure 2a shows a span-wise cut (z=0) of the computational domain, where the flow travels from left to right. The upstream domain boundary condition is set to a constant, uniform velocity U_∞. The inflow velocity is uniform and oriented such that the airfoil angle-of-attack is zero: $(u, v, w)_{\text{inflow}}=(U_\infty,0,0)$. A zero-gradient boundary condition is applied at the downstream boundary.

A boundary-fitted C-grid computational mesh is used. The meshing tool GRIDGEN[a] was used to create the one-dimensional (1D) mesh on the airfoil surface and downstream of the trailing-edge, which was then grown smoothly into a 2D planar mesh using HYPGEN[b]. Once the 2D mesh was created, an in-house code was used to extrude the mesh into a 3D domain, using uniform grid spacing in the span-wise direction. The span-wise width of the full computational domain is L_z=0.04C, which is discretized into N_z=96 grid points, such that $\Delta z=4.1667\times10^{-4}C$. The grid spacings were chosen such that at the wall, the viscous spacings (normalized by friction velocity, $u_\tau(x)$, and kinematic viscosity, ν) were $\Delta_x^+ \approx 100$, $\Delta r_{\min}^+ \approx 1$, and $\Delta_z^+ \approx 25$, where r is the wall-normal distance to the control volume centroid. Figure 2b shows the distribution of the computational mesh near the trailing-edge.

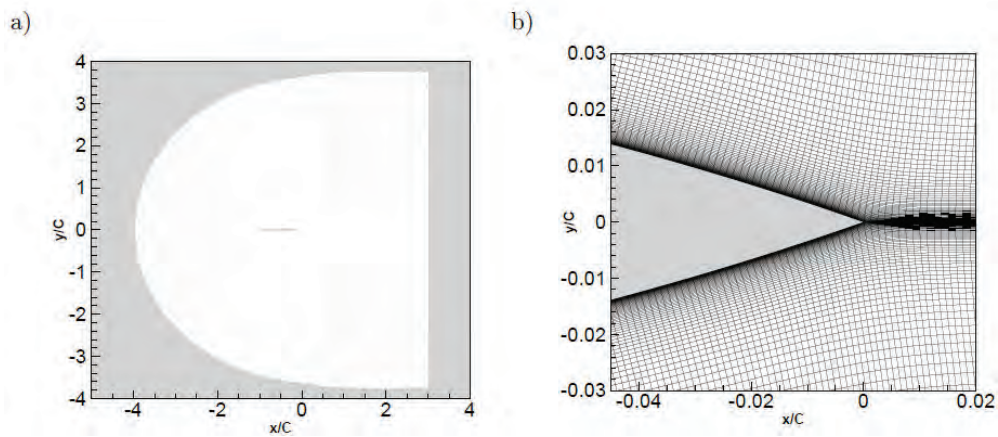


Figure 2. a) A span-wise cut (z=0) of the computational domain, denoted by the white region of the figure; b) Span-wise cut (z=0) of the computational mesh, near the trailing-edge

The computational mesh is composed of 32×10^6 control volumes, and is broken into 512 equal-sized partitions. Each partition runs on its own CPU, using MPI to communicate data between processors. Figure 3 plots the viscous grid spacing as a function of stream-wise position on the airfoil. Since the behavior of the flow near the trailing-edge is sensitive to upstream turbulence, obtaining sufficient grid resolution of the transition to turbulent flow (downstream of the leading-edge) and the subsequent evolution of the turbulent boundary-layer is important. The viscous grid spacings are largest near the stagnation point at the leading-edge and are (except for the span-wise grid spacing Δz^+ near the leading-edge) less than the target spacing ($\Delta x^+\approx100$, $\Delta y^+\approx1$, $\Delta z^+\approx25$) typically used for LES.

The Reynolds number of the flow, based on the chord length C, U_∞, and ν (Re_C) is $\sim10^6$. The flow solution was advanced with a constant time-step of $\Delta tU_\infty/C$=0.0001. To ensure initial transients were removed, the simulation was run at a Reynolds number of $\text{Re}_C=1.0\times10^6$ for 5 chord-based timescales ($t_C=C/U_\infty$), then for $2t_C$ at the target Reynolds number $\text{Re}_C=1.2\times10^6$. Statistics are then obtained over the next $3t_C$ at the target Reynolds number.

[a] Gridgen is a registered trademark of Pointwise, Inc.
[b] William M. Chan, NASA

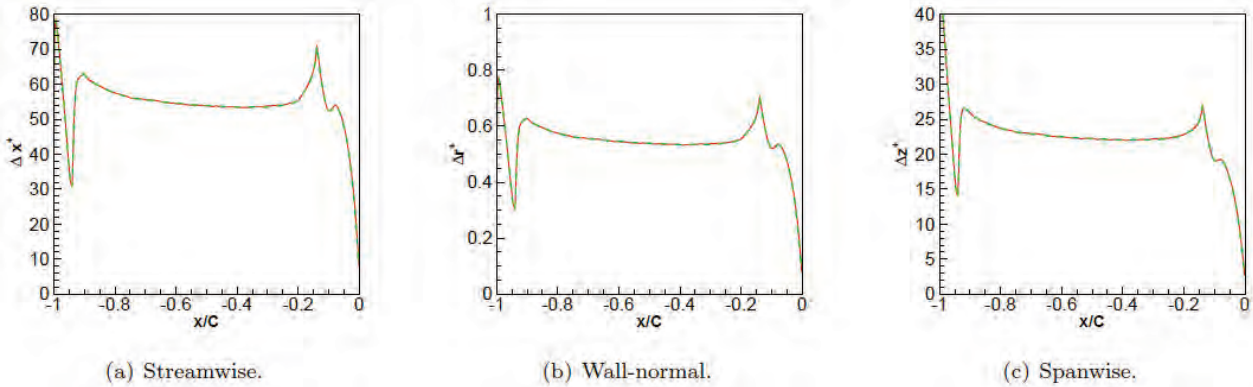


Figure 3. Time-averaged viscous spacings as a function of stream-wise position on the airfoil surface.; —: Suction-side; --: Pressure-side

On the Open Research System machine Chugach, a Cray XE6 with 728 compute nodes of 2×8-core AMD Magny-Cours Processors, each running at a core-speed of 2.64 GHz, the simulation was able to run through $1t_C$ in 26 hours on 512 CPUs, thus for a run of length $10t_C$ (enough time to remove transients and obtain statistics) uses 0.133×10^6 CPU hours.

C. Results

Below are preliminary results, which will (in the near future) be compared to currently ongoing experiments. Figure 4 plots the color contour of instantaneous stream-wise velocity, along with positions from which line plots of the flow statistics are taken. These line plots extend in the wall-normal direction from the airfoil surface. The flow travels from left-to-right and transitions naturally from laminar flow (starting at the stagnation point on the leading-edge, then around the nose to the flat section of the airfoil) to turbulent flow ($x/C>-0.95$). The turbulent boundary-layer grows with increasing downstream distance until it reaches the knuckle before the trailing-edge ($x/C=-0.15$), at which the flow is accelerated by a favorable pressure gradient, reducing the boundary-layer thickness. The adverse pressure gradient downstream of the knuckle reduces the near-wall momentum of the flow, creating a wake downstream of the trailing-edge. Figure 5a plots the instantaneous stream-wise velocity in the trailing-edge region. The increase in size of the flow length-scales are apparent as the trailing-edge is approached. The flow does not separate, as is shown by the mean (time-averaged) velocity streamlines in Figure 5b, though the near-wall momentum is reduced as compared to the upstream flow.

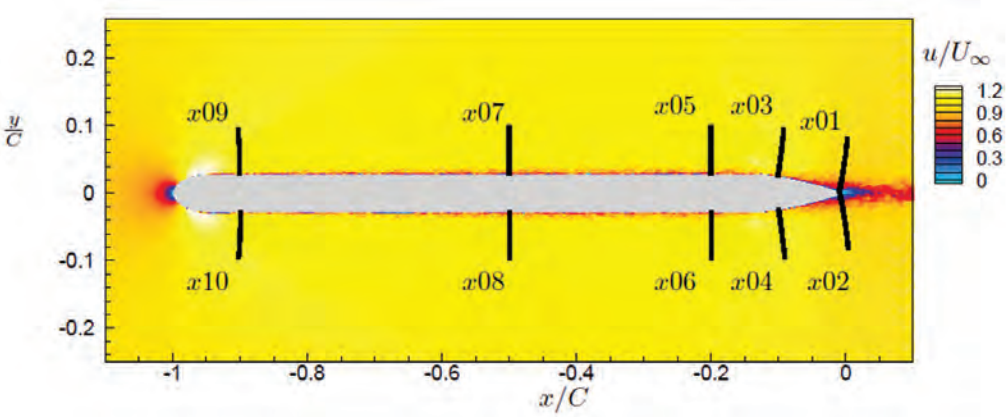


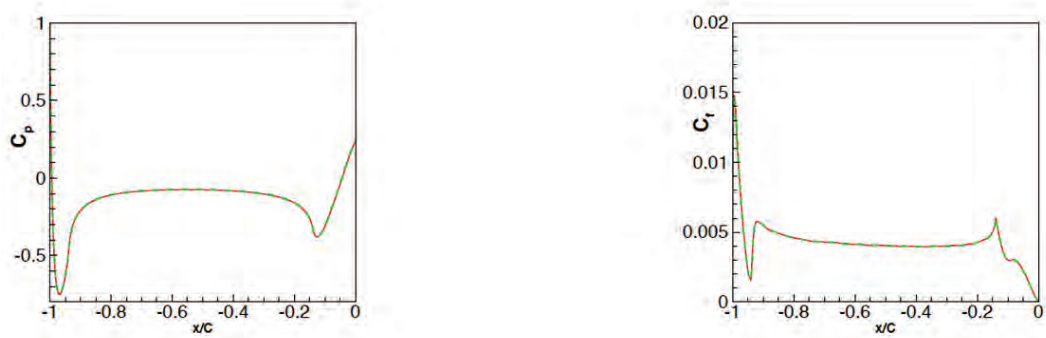
Figure 4. Color contours of instantaneous stream-wise velocity, taken from *z*=0 plane at tU_∞/C=10. Positions of probe stations are also shown (black lines).

(a) Instantaneous streamwise velocity at $tU_\infty/C = 10$

(b) Mean streamwise velocity along with mean streamlines

Figure 5. Color contours of stream-wise velocity in the *z*=0 plane

Time-averaged flow statistics on the pressure- and suction-sides are statistically similar, since the airfoil shape is symmetric about the y=0 plane. Figure 6a plots the mean pressure coefficient, $C_p(\vec{x}) = \left(p(\vec{x}) - p_\infty\right)/\left(1/2\rho U_\infty^2\right)$, as a function of stream-wise distance. Comparing the pressure- and suction-sides, the mean behavior is clearly equivalent. As typical of airfoil flows, the pressure coefficient decreases downstream from the stagnation point at the leading-edge (x=0), then increases as the flow diverts around the body ($0.96<x/C<0.7$). The stream-wise pressure gradient is nearly zero for much of the flat section of the airfoil, with the flow accelerating ($\partial C_p/\partial x<0$) near the knuckle, then decelerating ($\partial C_p/\partial x>0$) as the trailing-edge is approached. The skin friction coefficient, defined as $C_f(x) = \tau_w(x)/\left(1/2\rho U_\infty^2\right)$ where τ_w is the mean wall-shear stress, is plotted as a function of stream-wise distance in Figure 6b. The maximum shear stress is located at the stagnation point at the leading-edge (x=0). Once the flow has become turbulent, the mean shear gradually decreases, then increases as the flow accelerates near the knuckle, and finally, decreases as the adverse pressure gradient region is reached. This indicates that the grid spacing ($\Delta x^+=\Delta x/\Delta^+$) is naturally less-restrictive near the trailing-edge, since $\Delta^+ \nu/u_\tau$ is increasing as $x \rightarrow x_{TE}$.



(a) Pressure coefficient.

(b) Skin-friction coefficient.

Figure 6. Time-averaged pressure and skin friction vs. stream-wise location; —: Suction-side; --: Pressure-side

Line plots of the flow statistics are taken from lines that extend normal from the airfoil surface. For each station on the pressure-side, an equivalent station was placed on the suction-side, for a total of 10 wall-normal lines that extend from the airfoil surface ($x01-x10$, see Figure 4). The stream-wise locations of the stations are given in Table 1.

Table 1. Stream-wise location of each statistical probe station

Station	x/C
x01, x02	-0.01
x03, x04	-0.1
x05, x06	-0.2
x07, x08	-0.5
x09, x10	-0.9

At station $x07$, the flow is similar to that of a zero pressure-gradient turbulent boundary-layer. Here, the Reynolds number based on the momentum thickness (Re_θ) is equal to 1,488. Our LES results at $x07$ are compared to a fully-resolved, zero pressure-gradient direct numerical simulation (DNS) performed by Jiménez, et al.[22] at Re_θ=1,551. Figure 7(a) plots

the mean stream-wise velocity, normalized by u_τ, versus wall-normal position in viscous units. Also plotted is the log-layer profile $\left(u^+ = \frac{1}{\kappa}\ln\left(r^+\right) + B\right)$, where κ=0.5 and B=7.0. The log-law coefficients obtained by Jiménez, et al.[22] for this profile were κ=0.41 and B=5.1. The log-law intercept B and slope coefficient κ are slightly larger than that observed in zero pressure-gradient turbulent boundary-layers. Reasonable agreement is obtained in the inner-layer and log-law regions, though our LES predicts a lower outer-layer mean velocity and a different slope in the log-law region. Figure 7(b) compares the Reynolds stress term $\overline{u'u'}$, normalized by u_τ^2 versus wall-normal position. Reasonable agreement is obtained, though our LES predicts a larger peak of $\overline{u'u'}/u_\tau^2$.

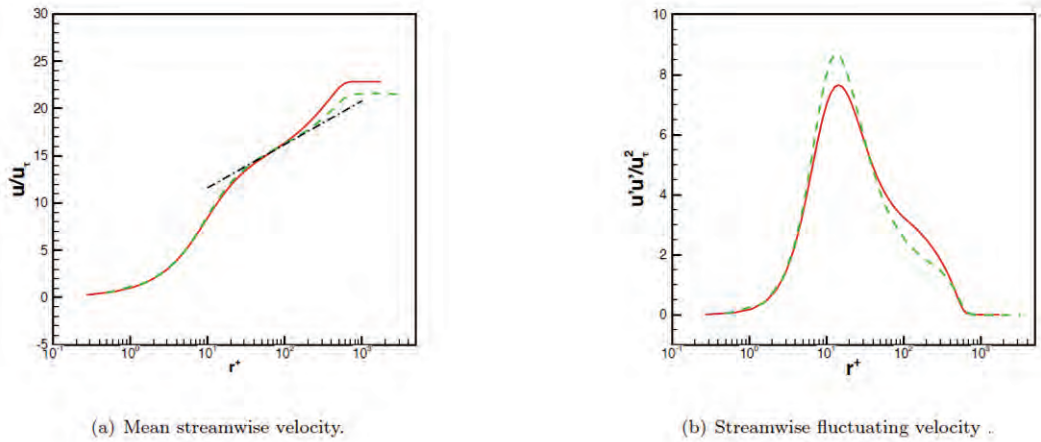


Figure 7. Time-averaged, viscous-scaled, wall-normal profiles as a function of viscous spacing from wall. Each symbol represents the values at the control-volume centroid. --, LES; —, Jiménez, et al.[22]; -···-, $\bar{u}/u_\tau$=ln(r^+)/0.5+7.0.

Profiles of mean stream-wise velocity as a function of stream-wise and wall-normal position are shown in Figure 8a. Only the results from the suction-side stations are shown, since the pressure-side statistics are similar. Most profiles follow the inner-law scaling and the log-law scaling, though $\bar{u}/u_\tau$ at station x01 (nearest the trailing-edge) increase in magnitude since the wall-shear τ_w, and thus u_τ, are decreasing.

The $\overline{u'u'}$ Reynolds stress term is plotted as a function of stream-wise and wall-normal position in Figure 8b. Only the results from the suction-side stations are shown, since the pressure-side statistics are similar. The values at stations x09 through x05 nearly collapse on a single curve, though significant differences are observed for stations x01 and x03, due to the adverse pressure gradient near the trailing-edge.

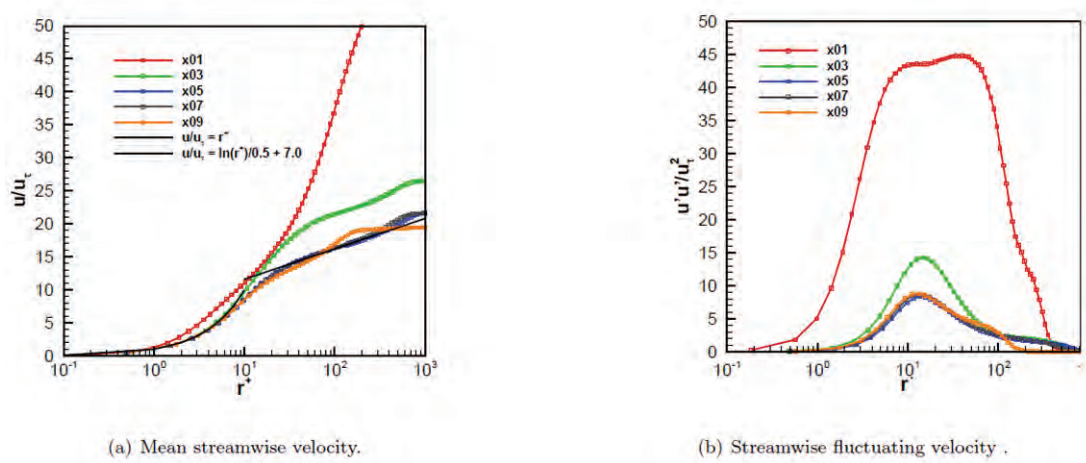


Figure 8. Time-averaged, viscous-scaled, wall-normal profiles

Figure 9 plots the evolution of the shape factor H, defined as the ratio of the displacement thickness to the momentum thickness ($H=\delta^*/\theta$). The shape factor is plotted as a function of Reynolds number (Re_x) based on stream-wise distance x, boundary-layer edge velocity U_e and ν. Near the leading-edge (low-Re_x) the flow is transitioning from laminar to turbulent

flow and the shape factor is ~1.6. With increasing downstream distance, the flow transitions to become a zero-pressure gradient boundary-layer and a shape factor approaches H~1.3, which is consistent with other zero-pressure gradient boundary-layer experimental results.[23] Near the knuckle (station x03), the favorable pressure gradient accelerates the flow and the value of H decreases to ~1.15. Near the trailing-edge, the edge velocity decreases and the length scales increase due to the adverse pressure gradient, and the shape factor increases. The LES results are also compared to an LES of flow over a flat plate with a semicircular leading-edge, performed by Yang, et al.[24] For the Yang, et al.[24], simulation, the Reynolds number is lower (Reynolds number (Re_h) based on h, U_∞ and ν is 3,450, where for our LES Re_h=70,000), thus the curve H vs. Re_x is shifted to the left. The comparison of H versus Re_x show qualitative agreement, where the shape factor is large near the leading-edge, then relaxing to the zero pressure gradient boundary-layer value.

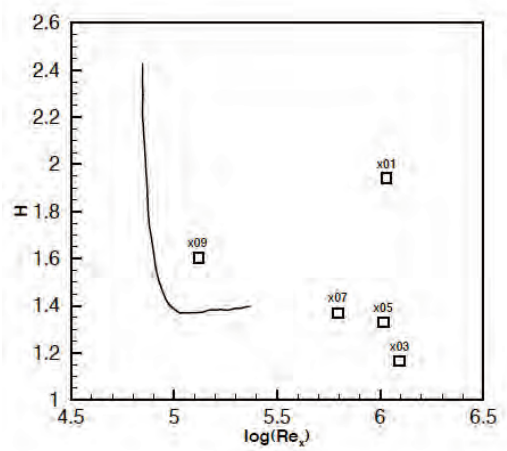


Figure 9. Evolution of shape factor *H* as a function of Re$_x$. — Yang, et al.[24]; □, LES.

3. Forward-Facing Step

A. Mathematical Formulation

The large-eddy simulation (LES) form of the NavyFOAM[25] solver is used to solve the grid-filtered, incompressible form of the Navier-Stokes equations:

$$\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} = \frac{1}{\rho}\frac{\partial p}{\partial x_j} + \nu \frac{\partial^2 u_j}{\partial x_i \partial x_i} + \frac{\partial \tau_{ij}^r}{\partial x_i} \tag{1}$$

In Equation 1, u_j is the grid-filtered velocity vector which resolves the energy containing scales of turbulence; p is the filtered pressure field; ρ is the density; ν is the kinematic viscosity; x_i are the Cartesian coordinates; t is time; and τ_{ij}^r is the residual stress tensor. The residual stress tensor is modeled using the dynamic Smagorinsky formulation,[26] in which τ_{ij}^r is defined as

$$\tau_{ij}^r = -2c_s\Delta^2 |S| S_{ij} \tag{2}$$

In Equation 2, S_{ij} is the resolved rate of strain tensor, $|S|$ is the magnitude of S_{ij}, Δ is the average mesh-spacing (filter width), and c_s is the dynamic Smagorinsky model coefficient, which is evaluated as a field variable through a double-filtering process. In the NavyFOAM LES framework, the convective terms are discretized via a central difference formulation which is second-order in space, and the time derivative is discretized using a second-order, implicit formulation.

Far-field sound pressures caused by turbulent flow over surface imperfections are determined in NavyFOAM using the Ffowcs Williams-Hawkings[27] (FWH) generalization of Lighthill's acoustic analogy. Lighthill[15] decomposed the Navier-Stokes equations into a linear wave propagation operator and its corresponding sound sources, which is called Lighthill's equation, and is written here in terms of the acoustic pressure, $p'=p-p_o$, where p_o is the free-stream reference pressure, and the far-field speed of sound, c_o:

$$\frac{1}{c_o}\frac{\partial^2 p'}{\partial t^2}-\frac{\partial^2 p'}{\partial x_i \partial x_j}=\frac{\partial^2 T_{ij}}{\partial x_i \partial x_j} \tag{3}$$

In Equation 3, T_{ij} is the Lighthill stress tensor

$$T_{ij}=\rho u_i u_j + P_{ij} - \delta_{ij} c_o^2 \left(\rho - \rho_o\right) \tag{4}$$

where the compressive stress tensor, P_{ij}, is given by

$$P_{ij}=p\delta_{ij}-2\mu\left(S_{ij}-\frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \tag{5}$$

where

$$S_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{6}$$

is the strain rate tensor. The numerical solution to Equation 1 provides the fluid dynamic information needed to evaluate the Lighthill stress tensor Equation 4, which, essentially, is the source term for the linear wave equation in Lighthill's formulation Equation 3. The solution to Equation 3 is generally accomplished using an integral formulation. One of the most successful solutions to date is that of FWH.[27] NavyFOAM used the implementation of FWH embodied in the WOPWOP solver of Brentner and Farassat.[28] The following description of the FWH formulation is taken from Brentner and Farassat.[28] Using the theory of generalized functions, FWH derived the following generalized form of Lighthill's equation for arbitrarily moving sources:

$$\frac{1}{c_o}\frac{\partial^2 p'}{\partial t^2}-\frac{\partial^2 p'}{\partial x_i \partial x_j}= \underbrace{\frac{\partial}{\partial t}\left\{\left[\rho_o v_n+\rho\left(u_n-v_n\right)\right]\delta\left(f\right)\right\}}_{\text{monopole sources}}-\underbrace{\frac{\partial}{\partial x_i}\left\{\left[\Delta P_{ij}n_j+\rho\left(u_n-v_n\right)\right]\delta\left(f\right)\right\}}_{\text{dipole sources}}+\underbrace{\frac{\partial^2}{\partial x_i \partial x_j}\left[T_{ij}H\left(f\right)\right]}_{\text{quadrupole sources}} \tag{7}$$

In Equation 7, $f(x_i, t)=0$ is a data surface that encloses the noise-generating region of the flow, and may include the solid surfaces that are made of the surface imperfection and adjacent wall-surfaces. In general, f can be a moving surface. Also in Equation 7, : n_j is the outward-directed unit-normal vector such that $\Delta f=n_j$; $v_n=-\partial f/\partial t$ is the local normal velocity of the surface; $\Delta P_{ij}=(p-p_o)\delta_{ij}$ and $H(f)$ and δ_{ij} are the Heaviside and Dirac delta functions, respectively. In Equation 7, the first term on the right-hand side represents monopole, or thickness sources caused by displacement of fluid as the surfaces passes. The second term represents dipole, or loading, sources caused by unsteady fluid motion altering the force distribution on the surface. The third term represents volume quadrupole sources. These quadrupole sources arise from turbulent fluctuations in the fluid outside of the data-surface, and are typically negligible for low-Mach number or incompressible, flows. If the data surface is permeable and encloses the acoustic sources, then the first two terms, which are surface-source terms, will account for the acoustic sources outside of the hard surfaces of the body of interest, but inside the permeable data-surface. The solution to Equation 7 is obtained in integral form using the free-space Greens function, $\delta(g)/4\pi r$, where $r=\left|\vec{x}-\vec{y}\right|$ is the distance between the acoustic source and the observer, $g=\tau-t+r/c_o$ and $\tau=t-r/c_o$ is the source time. Integration yields:

$$p'\left(x,t\right)=p'_T\left(\vec{x},t\right)+p'_L\left(\vec{x},t\right) \tag{8}$$

where p'_T and p'_L are the acoustic pressures due to "thickness" and "loading," respectively and

$$4\pi p'_T=\int\int_{f=0}\left[\frac{\rho_o\left(\dot{v}_n+v_{\dot{n}}\right)}{r\left|1-M_r\right|^2}\right]_\tau dS+\int\int_{f=0}\left[\frac{\rho_o v_n\left(r\dot{M}_r+c_o M_r-c_o M_r^2\right)}{r\left|1-M_r\right|^3}\right]_\tau dS \tag{9}$$

and

$$4\pi p'_L= \frac{1}{c_o}\int\int_{f=0}\left[\frac{\dot{l}_r}{r\left|1-M_r^2\right|}\right]_\tau dS+\frac{1}{c_o}\int\int_{f=0}\left[\frac{l_r-l_m}{r^2\left|1-M_r^2\right|}\right]_\tau dS+\int\int_{f=0}\left[\frac{l_r\left(r\dot{M}_r+c_o M_r-c_o M_r^2\right)}{r^2\left|1-M_r\right|^2}\right]_\tau dS \tag{10}$$

In Equations 9 and 10, M is the Mach number of the source, a dot over a variable implies differentiation with respect to the source-time, τ, and all integral kernels in square brackets are evaluated at the source-time. In addition, $l_i=P_{ij}n_j$, and the subscripts n, r, and m refer to projection onto the unit normal vector, the unit vector in the direction of the acoustic radiation, or the surface velocity normalized by the speed of sound, respectively. Equation 9 is referred to as the thickness term, and Equation 10 is referred to as the loading term.

In the current implementation of NavyFOAM, the data-surface f=0 is typically chosen to be a stationary, impermeable surface corresponding to the wall-surfaces in question. Accordingly, the quadrupole volume source terms are usually neglected. Also, since the data surface is impermeable, the normal velocity of the fluid is the same as the normal velocity of the surface, i.e., $u_n=v_n$. Finally, since the data surface is stationary, all Mach number-related terms in Equation 9 and Equation 10 become zero. In the far-field, when the data surface is stationary, the first term in Equation 10 dominates, since it decays as $1/r$, compared to the second term, which decays as $1/r^2$. For evaluation in the far-field, the local force intensity acting on the fluid can be expressed as

$$\dot{l}_r = \frac{\partial}{\partial \tau}\left(P_{ij}n_j\right)\hat{r}_i = \frac{\partial}{\partial \tau}\left[p\delta_{ij} - 2\mu S_{ij}\right]\hat{r}_i \tag{11}$$

where p and S_{ij} are supplied by the incompressible LES.

In general, the free-space Greens function in the FWH formulation is appropriate for calculating the radiated sound when the acoustic source field is characterized by a compressible LES. This is because the compressible flow field naturally contains the acoustic waves, and, thus, satisfies the acoustic boundary conditions at hard-walls for those waves (wave-scattering caused by diffraction and reflection). Scattering can strongly influence both the magnitude and directivity of the radiated sound. The current implementation of the acoustic analogy in NavyFOAM relies on an incompressible LES to characterize the acoustic sources. As a result, no acoustic wave scattering is accounted for when using FWH to calculate the radiated sound. Hence, reliable acoustic radiation calculations are limited to problems where acoustic wave scattering effects are negligible. This restriction is typically acceptable when the acoustic source distribution is acoustically compact. Acoustical compactness is often defined as an acoustic source distribution whose characteristic length is much less than the acoustic wavelength of interest. Fortunately, for many cases of interest at US Naval Surface Warfare Center, Carderock Division (NSWCCD), the working fluid is usually water, which has relatively long wave-lengths for acoustic frequencies up through the tens of kilohertz range. These wave-lengths often far exceed the characteristic lengths of the hard wall-surface features that are responsible for generating the acoustic sources. Consequently, an incompressible LES, using FWH as the acoustic analogy, is often acceptable for radiated sound calculations in water.

B. Synthesis of a Turbulent Inflow Boundary-Layer

Since the *computational* domain of many flow problems is often only a subset of the *actual* space through which the physical flow propagates, the onset turbulent boundary-layer may need to be created within the simulation. This can be done formally by providing enough stream-wise space upstream of the region of interest to allow an initially-laminar boundary-layer to naturally transition to an equilibrium-turbulent boundary-layer. However, providing sufficient upstream space, along with the necessary spatial resolution, can quickly render the simulation too computationally expensive. To generate an onset-turbulent boundary-layer in a more computationally economic fashion, turbulence can be synthesized at the inflow boundary of the computational domain using the Random Flow Generation (RFG) technique of Smirnov, et al.[29] In this formulation, a mean turbulent velocity profile imposed at the inlet is perturbed at each computational time-step by a transient, random, three-dimensional flow field generated from a model velocity spectrum. Since the synthesized flow field is two-dimensional at the inflow boundary, organized turbulent structures will not develop until some distance downstream of the inflow plane. Hence, the domain must still include sufficient stream-wise distance downstream of the inflow plane to allow the perturbed structures to organize into an equilibrium-turbulent boundary-layer. However, this distance is much less than what would be required to simulate a natural transition from laminar to turbulent flow. The mean turbulent velocity profile at the inlet boundary is typically generated for each flow case via a Reynolds-averaged Navier-Stokes simulation of the larger, actual domain—in this case a flat plate boundary-layer at the appropriate free-stream velocity. Velocity profiles at various stream-wise locations along the flat plate are then examined until a profile is found that closely matches the momentum thickness and boundary-layer thickness expected for the actual flow noise problem. This profile is then used as the mean turbulent velocity profile for the inlet boundary condition.

C. Problem Setup

A typical configuration for a forward-facing step is shown in Figure 10(a). A series of experimental investigations into the noise induced by turbulent flow over a forward-facing step have been carried out at the David Taylor Model Basin. These investigations addressed Reynolds numbers, based on step-height, from 2,152 through 33,332, with upstream turbulent boundary-layer thicknesses varying from approximately two- to nine-times the step-height. Data from these experiments has been made available in the form of fluctuating wall-pressure spectra, and far-field acoustic spectra from Chang[c].

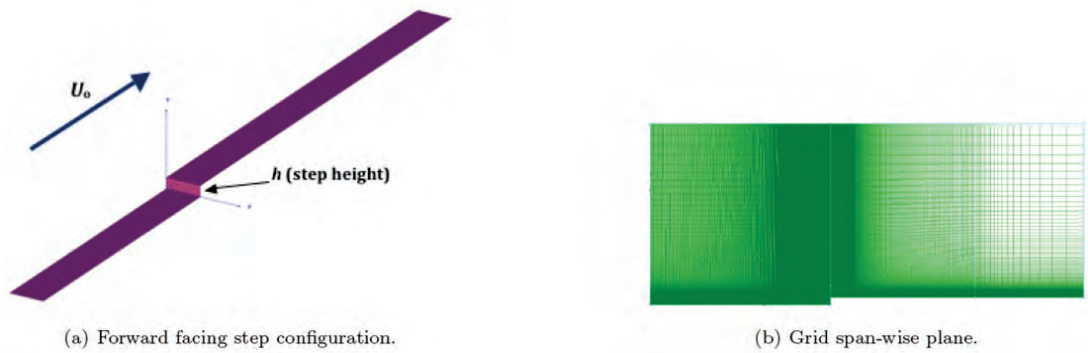


Figure 10. Forward-facing step problem setup

The authors have chosen Re_h=21,904, with an inflow turbulent boundary-layer whose thickness is approximately 2h, and free-stream Mach number of 0.08 as their validation test case. Figure 10(b) shows a portion of the computational domain for the forward-facing step problem. This is a span-wise plane showing the grid resolution in the vicinity of the step. L_x is the length of the domain in the stream-wise (flow) direction, L_y is the length in the wall-normal direction, and L_z is the span-wise length. The grid is non-uniform in the stream-wise and wall-normal directions, and uniform in the span-wise direction. In the present simulation, the grid extended 25h upstream of the step face, 35h step-heights downstream of the step-face, 25h above the step-face, and 4h in the span-wise direction.

In the present simulation, the stream-wise spacing varied from $67\Delta^+$ at the inlet of the domain, $5\Delta^+$ in the vicinity of the step, to $2{,}160\Delta^+$ at the outlet of the domain. The wall-normal spacing varied from $3\Delta^+$ at the wall to approximately $65\Delta^+$ in the outer-layer of the boundary-layer. The span-wise spacing was constant at $29.5\Delta^+$. The resulting grid had 22×10^6 sub-volumes.

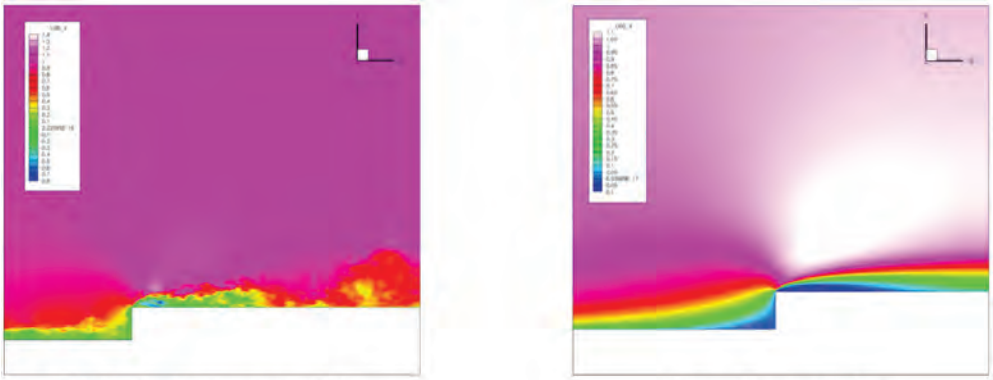
The spectral synthesis method discussed earlier was used to create a turbulent boundary-layer at the inflow boundary. A no-slip boundary condition was imposed at the walls, and a periodic boundary condition was imposed in the span-wise direction. Free-stream velocity was specified at the top of the domain, and a convective boundary condition was imposed at the outlet of the domain.

The simulation was run using a time-step size $\Delta tU/h=5\times10^{-3}$, which was sufficient to converge the numerical solution in both velocity and pressure at each real time-step. On the Army Research Laboratory cluster Harold, a SGI Altix ICE 8200 with 1,344 compute nodes each with two Intel Xeon quad-core Nehalem processors running at a core speed of 2.8 GHz, the simulation was run for 250,000 time-steps in 1,344 hours on 256 CPUs consuming 0.356×10^6 CPU-hours.

D. Results

The LES was run until the solution was statistically stationary. Figure 11 shows some flow characteristics from the LES that are typical of turbulent flows over forward-facing steps. The direction of the flow is from left to right. Figure 11(a) shows an instantaneous realization of the stream-wise velocity, normalized by the magnitude of the free- stream velocity. Figure 11(b), the time-averaged stream-wise velocity component normalized by the free-stream velocity, reveals two recirculation zones that are set up by flow separation and re-attachment. The first is at the base of the step-face, and the second is in an interval about 2−3h long downstream of the step.

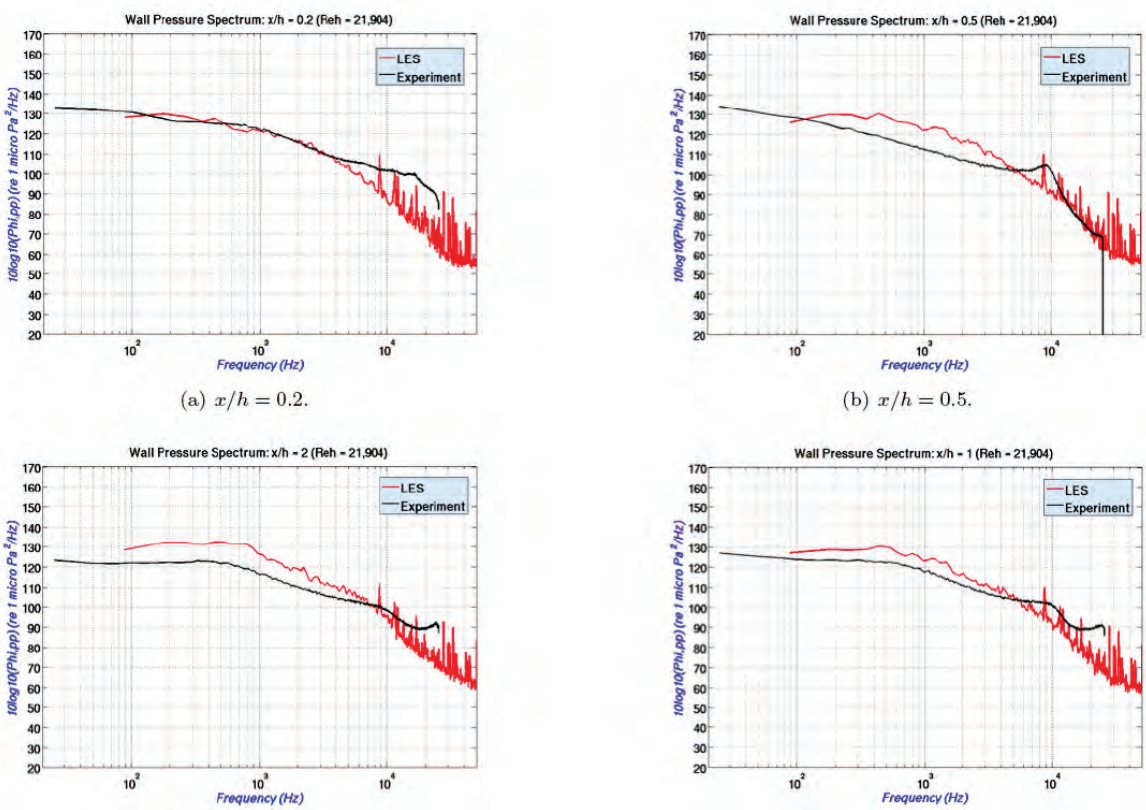
[c] Chang, Natasha, personal communication.



(a) instantaneous stream-wise velocity. (b) mean stream-wise velocity.

Figure 11. Some flow characteristics near the step region from the LES

Figure 12 shows the power spectral density (PSD) of the fluctuating wall-pressure at selected locations along the center-line of the step geometry. These locations correspond to the flow region from 0−2*h* downstream of the step beneath the separated flow region which is shown in Figure 11. This region has been found responsible for forming the strongest acoustic sources.[30]



(a) $x/h = 0.2$. (b) $x/h = 0.5$.

(c) $x/h = 1.0$. (d) $x/h = 2.0$.

Figure 12. Power spectral density of the fluctuating wall-pressure

The experimental PSD are generally broadband. The "peaks" in the experimental PSD at 9×10^3 Hz have been attributed to local cavity-type flow resonances caused by the recessed mounting of the pressure transducers and therefore, should not be in the LES spectra. The very small "peak" in the experimental PSD at x/h=2—the middle of the separated flow zone—at 500 Hz may be due to a weak periodic shedding. In general, the LES has resolved to within about 5% the amplitude of the wall-pressure spectra up to about $4-6\times10^3$ Hz. The "noise" in the LES PSD at $f>9\times10^3$ Hz arises from the spectral perturbations at the inflow boundary, and is not physical.

1. Radiated Sound Spectrum

Using the FWH acoustic analogy identified earlier in this report, the far-field acoustic spectrum was calculated. The span of the step configuration in the DTMB experiments was 29.4*h*, which is greater that the span of 4*h* used in the computational domain. The actual far-field acoustic spectra for the full span (L) of the step configuration must be represented by an appropriate summation of the spectra for L/L_z=7.35 partial span LES domains.

Following Wang and Moin,[8] if Λ_z is the span-wise coherence length for a given frequency, the total acoustic sound spectrum for the full span can be found by one of two limiting conditions:

1. If $L_z \geq \Lambda_z$, the span-wise extent of the computational domain exceeds the coherence length of the acoustical source. Consequently, the acoustical sources in each span-wise domain radiate in a statistically independent manner. Therefore, the total sound spectrum is the linear sum of the contributions from each span-wise domain, resulting in L/L_z independent source regions along the span, and $\Phi_{pa}^{total} \approx (L/L_z)\Phi_{pa}$, where Φ_{pa} is the power spectrum from the acoustic sources in each span-wise computational domain.
2. If Φ_{pa}, the acoustic source is coherent along the entire span of the forward step surface. It is essentially a two-dimensional source, and, ignoring the span-wise variation of the source time $\Phi_{pa}^{total} \approx (L/L_z)^2 \Phi_{pa}$.

In the present study, the span-wise dimension of the computational domain was selected to ensure that the resolved turbulent structures are correlated over a length which is much less that the span-wise dimension of the domain. Since these turbulent structures serve as the basis for the acoustic source terms in Equation 10, $\Phi_{pa}^{total} \approx (L/L_z)\Phi_{pa}$ is expected to be the appropriate summation to use for most of the far-field acoustic spectrum.

In hydro-acoustic simulations, it is important to avoid spurious noise sources which can occur at the domain boundaries. Therefore, the grid adjacent to the outflow boundary of the LES domain was stretched (L_x=2160Δ^+) to provide a buffer region to dissipate the fluid dynamic fluctuations in this area, which would otherwise become spurious acoustic sources via Equation 10. Figure 13 shows the far-field sound spectrum that was calculated for the forward-facing step for an observer position in the mid-span plane of the step configuration and 1m directly above the step-face. Since this problem is not acoustically compact for frequencies near 10kHz and above, the calculated spectrum is only presented for frequencies below about 5kHz. The calculate spectrum is in reasonably good agreement with experimental measurements: displaying the frequency behavior of the experimentally measured spectrum, while over-predicting the amplitude by an average of about 5 to 8 dB.

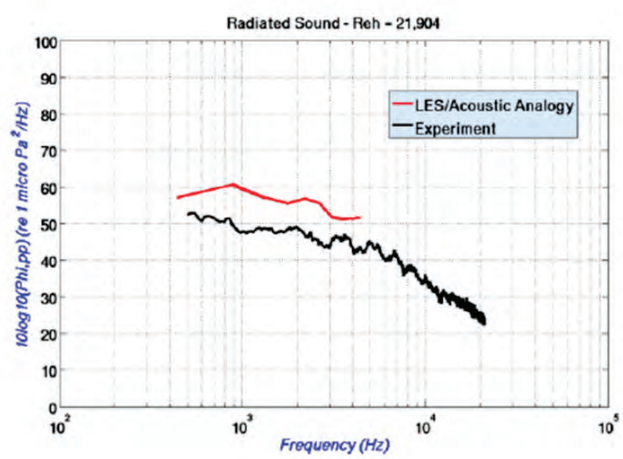


Figure 13. Power spectral density of the radiated sound 1m directly above the step-face

4. Conclusion

In this paper, we show two hydro-acoustics problems that are building-blocks for larger problems. In both cases the hydro-acoustics rely upon the prediction of fluctuating velocities and wall-pressures, which requires using large-eddy simulations (LES). These are very computationally-intensive CFD tools that require large HPC resources.

The objective of the TEVS simulations is to predict the narrow-band far-field radiated noise signature due to vortex shedding from a beveled trailing-edge. The accurate prediction of this signature; however, requires that the kinematically-

and dynamically-correct upstream portion of the hydrofoil flow be simulated. In this paper we show that we can accurately predict the mean flows and fluctuating quantities in the upstream boundary-layers. In addition, we show that we can perform well-resolved simulations in a time frame that is suitable for design evaluations.

For the forward-facing step problem we have implemented a generalized FWH solution to Lighthill's equation into the NavyFOAM solver. In this methodology, the acoustic pressure is a surface integral over the local force intensities which are evaluated using surface pressure and tractions from incompressible LES. We compare the PSD of wall-pressure fluctuations to experimental results with good agreement over a useful frequency band. We also show that the far-field radiated noise is well-predicted to within a constant factor.

Acknowledgements

This work was funded by Office of Naval Research, Dr. Ki-Han Kim, sponsor, Discovery and Invention (D&I) Hydro-acoustics. We would also like to thank Prof. Ken Brentner for generously providing the WOPWOP code; and Dr. Sung-Eun Kim for his assistance with the NavyFOAM LES solver. We appreciate the hydro-acoustics support of Doug Noll, Natasha Chang, and Irek Zawadzki.

The computations were performed under a Grand Challenge grant of high performance computing time (Challenge Project C5K) from the Department of Defense (DoD) Supercomputing Research Center, Army Research Laboratory in Aberdeen Proving Ground, MD; SGI Altix ICE 8200 Harold and the Open Resource System, Cray XE6, Chugach.

References

1. Blake, W.K., *Mechanics of Flow-Induced Sound and Vibration, Vol. 17 of Applied Mathematics and Mechanics*, Academic Press, Inc., Orlando, FL, 1986.

2. Chang, P., Gershfeld, J., and Wang, M., "Towards the Prediction of Vortex Shedding from an Airfoil using LES." *11th AIAA/CEAS Aeroacoustics Conference (20th AIAA Aeroacoustics Conference)*, Monterey, CA, p. 14, 2005.

3. Chang, P., Wang, M., and Gershfeld, J., "Prediction of vortex shedding from a high-Reynolds number airfoil using LES with and without wall model." *2006 ASME Fluids Eng'g Division Summer Meeting*, Miami, FL, 2006.

4. Chang, P.A., Vargas, A., Lummer, D., Jiang, M., and Mahesh, K., "Fully-resolved LES of weakly separated flows." *20th AIAA Computational Fluid Dynamics Conference*, AIAA, Honolulu, HI, 2011.

5. Oberai, A.A., Roknaldin, F., and Hughes, T.J., "Computation of Trailing-Edge Noise Due to Turbulent Flow Over an Airfoil." *AIAA Journal*, Vol. 40, No. 11, pp. 2206–2216, 2002.

6. Wang, M., Moreau, S., Iaccarino, G., and Roger, M., "LES prediction of pressure fluctuations on a low-speed airfoil." *Annual research briefs*, Center for Turbulence Research, 2004.

7. Wang, M. and Moin, P., "Computation of Trailing-Edge Flow and Noise Using Large-Eddy Simulation." *AIAA Journal*, Vol. 38, No. 12, pp. 2201–2209, 2000.

8. Wang, M. and Moin, P., "Dynamic wall-modeling for large-eddy simulation of complex turbulent flows." *Physics of Fluids*, Vol. 14, No. 7, pp. 2043–2051, 2002.

9. Wang, M., "Computation of trailing-edge aero-acoustics with vortex shedding." *Annual research briefs*, Center for Turbulence Research, 2005.

10. Gershfeld, J., Blake, W.K., and Knisely, C.W., "Trailing-edge Flows and Aerodynamic Sound." **AIAA**, Cincinnati, OH, pp. 2133–2140, 1988.

11. Farabee, T. and Casarella, M., "Effects of surface irregularity on turbulent boundary-layer wall-pressure fluctuations." *Journal of Vibration, Acoustics Stress and Reliability in Design*, Vol. 106, pp. 343–350, 1984.

12. Farabee, T. and Casarella, M., "Measurement of fluctuating wall-pressure for separated/reattached boundary-layer flows." *Journal of Vibration, Acoustics Stress and Reliability in Design*, Vol. 108, pp. 301–307, 1986.

13. Farabee, T. and Zoccola, P., "Experimental evaluation of noise due to flow over surface steps." *ASME International Engineering Congress and Exposition*, 1998.

14. Catlett, M. and Devenport, W., "Sound from boundary-layer flows over steps and gaps." *AIAA*, Vol. 2010-3774, 2010.

15. Lighthill, M., "On Sound Generated Aerodynamically: I. General Theory." *Proceedings of the Royal Society of London, Series A: Mathematical and Physical Sciences*, Vol. 211, pp. 564–587, 1952.

16. Moin, P. and Mahesh, K., "Direct Numerical Simulation: a tool in turbulence research." *Annual Review of Fluid Mechanics*, Vol. 30, pp. 539–578, 1998.

17. Harlow, F. and Welch, J., "Numerical calculation of time-dependent viscous incompressible flow of fluid with free-surface." *Physics of Fluids*, Vol. 8, pp. 2182–89, 1965.

18. Mahesh, K., Constantinescu, G.S., and Moin, P., "A numerical method for large-eddy simulation in complex geometries." *Journal of Computational Physics*, Vol. 197, pp. 215–240, 2004.

19. Vysohlid, M. and Mahesh, K., "Large-Eddy Simulation of Crashback in Marine Propellers." *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2006.

20. Vysohlid, M. and Mahesh, K., "Large-Eddy Simulation of Crashback in Marine Propellers." *26th Symposium on Naval Hydrodynamics*, Rome, Italy, 2006.

20a. Chang, Peter A., Ebert, M. Mahesh, K., Shipman, J., "Prediction of High Amplitude Forces During Propeller Crashback." *DoD HPCMP Users Group Conference 2008*, Seattle, WA, July 2008.

21. Henson, V.E. and Yang, U.M., "BoomerAMG: a Parallel Algebraic Multigrid Solver and Preconditioner." *Applied Numerical Mathematics*, Vol. 41, 2002.

22. Jimenez, J., Hoyas, S., Simens, M., and Mizuno, Y., "Turbulent boundary-layers and channels at moderate Reynolds numbers." *Journal of Fluid Mechanics*, Vol. 657, pp. 335–360, 2010.

23. Klebanoff, P., "Characteristics of turbulence in boundary-layer with zero pressure gradient." Tech. rep., NACA, 1954.

24. Yang, Z. and Voke, P.R., "Large-eddy simulation of boundary-layer separation and transition at a change of surface curvature." *Journal of Fluid Mechanics*, Vol. 439, pp. 305–333, 2001.

25. Shan, H., Delanay, K., Kim, S.-E., Rhee, B., Gorski, J.J., and Ebert, M., "Guide to NavyFOAM V1.0." *Tech. Rep. NSWCCD-50-TR-2011/025*, Naval Surface Warfare Center, Carderock Division, Apr 2011.

26. Germano, M., Piomelli, U., Moin, P., and Cabot, W.H., "A dynamic subgrid-scale eddy viscosity model." *Physics of Fluids A-Fluid Dynamics*, Vol. 3, No. 7, pp. 1760–1765, 1991.

27. Ffowcs-Williams, J. and Hawkings, D., "Sound Generation by Turbulence and Surfaces in Arbitrary Motion." *Philisophical Transactions of the Royal Society of London, Series A: Mathematical and Physical Sciences*, Vol. 264, pp. 321–342, 1969.

28. Brentner, K. and Farassat, F., "Modeling aerodynamically-generated sound of helicopter rotors." *Progress in Aerospace Sciences*, Vol. 39, pp. 83–120, 2003.

29. Smirnov, A., Shi, S., and Celik, I., "Random Flow Generation Technique for Large-Eddy Simulations and Particale-Dynamics Modeling." *Journal of Fluids Engineering*, Vol. 123, pp. 359–371, 2001.

30. Slomski, J.F., "Numerical investigation of sound from turbulent flow over a forward-facing step." *17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference)*, Portland, OR, 2011.

6.4 Advanced Component Development and Prototypes

Army

Distributed HPC-enabled LVC S&T Experimentation Environments

Daniel Ku, Rodney Smith, and Young Choi
US Army Communications-Electronics Research, Development and Engineering Center (CERDEC), Aberdeen Proving Ground, MD
{daniel.t.ku, young.s.choi17}.civ@mail.mil, rodney.smith.ctr@mail.mil

Abstract

As the Army's operational paradigm shifts toward Network-Centric Warfare (NCW), the use of high performance computing (HPC) assets to aid experimentation in Live, Virtual, and Constructive (LVC) environments has become increasingly valuable. Live test and evaluation of large, complex Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) System-of-Systems (SoS) is becoming impracticable due to the cost and component availability. The use of a shared HPC-enabled LVC Science and Technology (S&T) environment to aid in stimulating, scaling, and analyzing these technologies is a solution that PD C4ISR & Network Modernization (C4ISR & Net Mod) leveraged heavily during experiments conducted at Fort Dix, NJ for Event 2011.

1. Introduction

The mission of the complex Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) & Network Modernization (C4ISR & Net Mod) Modeling and Simulation (M&S) team is to provide the Army a cost-effective, relevant, and scalable Live, Virtual, and Constructive (LVC) environment, which can be employed to aid the assessment of emerging C4ISR SoS technologies by utilizing simulations of varying fidelity to stimulate live experimentation involving integrated C4ISR Systems-of-Systems (SoS) in a net-centric warfare (NCW) context.

The scales at which these complex SoS need to be examined pose challenges for experiment designers. Availability of emerging C4ISR SoS components, especially hardware, is generally low and the few developmental items in existence are often in high-demand. Further, most of these items are not "ruggedized" and cannot stand up to the rigors of field experimentation. Even if these components were available in large quantities, the cost of conducting live experimentation at the scale necessary would quickly become prohibitive. High performance computing (HPC)-based virtual and constructive simulations are tools that may help overcome these challenges.

2. ARL & SMDC Shared HPC Assets

During E11, the L/V/C team, in conjunction with the Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP) Productivity Enhancement, Technology Transfer and Training program (PETTT), continued a proof-of-concept utilizing two DoD Supercomputing Resource Center (DSRC) HPC resources for HPC-based simulation. The HPC resources are the US Army Research Laboratory (ARL) DSRC located at Aberdeen Proving Ground (APG), MD and the US Army Space & Missile Defense Command (SMDC) Simulation Center located at Huntsville, AL.

The team installed and configured OneSAF on 24 compute-nodes of the MJM HPC at ARL and on 24 compute nodes of the Simulation Machine –Q (SMQ) system at SMDC (see Figure 1).

MJM is a Linux Networx Advanced Technology Cluster. It offers 4400 Intel Woodcrest core 3.0 GHz processors and 8.8TB of total memory with a peak system performance of 52.8 TFLOPS. Each compute-node contains 4 processor cores and 8GB of memory. C4ISR & Network Modernization reserved 24 MJM compute-nodes providing 96 process cores and 192GB of memory. The theoretical system performance is 48 GFLOPS per node for a peak of approximately 1.15 Teraflops.

Simulation Machine-Q (SMQ) is an ATIPA Cluster system. It offers 256 nodes each with 2 quad-core processors totaling 2,048 cores@2.2 GHZ clock-speed and 17.4TB of total memory with a peak system performance of 18 TFLOPS overall. Each compute-node contains 2 AMD Opteron Barcelona quad-core processors and 16GB of memory. C4ISR & Network Modernization reserved 24 SMQ compute-nodes providing 192 processor cores and 384GB of memory. The theoretical system performance is 70.4 GFLOPS per node for a peak of approximately 1.69 Teraflops.

The HPC systems were remotely connected to the E11 SoS through a VPN tunnel operating over the DREN. Both MJM and SMQ are "production systems" that are available to wide-range of general users. Identical OneSAF distributed simulation configurations and scenarios were implemented and loaded on each system. The rationale for running identical scenarios on both systems was to provide a redundant capability. While MJM functioned as the primary BCT-level simulation provider, power outages and scheduled system maintenance activities at APG affected its ability to provide a reliable simulation source during critical experimentation periods. HPCMP PETTT was able to facilitate L/V/C team access to SMQ and the team's prior experience with MJM, and the Harold shared HPC employed in E10, enabled it to configure SMQ rapidly as a back-up.

Figure 1. ARL's MJM and SMDC's SMQ Shared HPCs

3. M&S Tools

OneSAF v3.0 Description

One Semi-Automated Forces (OneSAF) is a government-off-the-shelf (GOTS), Computer-Generated Forces (CGF) simulation suite. OneSAF is a "composable" simulations toolkit that allows multiple Army domains—Advance Concepts and Requirement (ACR), Research Development and Acquisition (RDA), and Training Exercises and Military Operations (TEMO)—to modify, create, and mold the simulation to meet their specific simulation requirements. These types of simulations are often referred to as constructive simulations. A constructive simulation toolkit, such as OneSAF, allows users to represent and control large numbers of individual entities within a given force structure from a user-friendly graphical user interface (GUI). The ability to accurately represent large numbers of Army units in a realistic manner is critical when training Army battle staffs and testing new battle command systems. During E11, OneSAF, running on DSRC HPC, represented the positions and behaviors of approximately 3,300 moving entities (or approximately a brigade combat team (BCT)) that stimulated the live battle command applications under experimentation, including BCS, MCS, AFATDS, DCGS-A, JCR, and FBCB2. OneSAF also was used to provide the position and movements for two virtual platoons that were operating in concert with a live platoon.

4. Use of Shared HPC

The reasons for utilizing a shared HPC resource were two-fold: First, C4ISR & Network Modernization has been without a Dedicated HPC Project Investments (DHPIs) for an extended period of time and the L/V/C team wanted to explore the use of a shared HPC to bridge the gap as an interim HPC solution. Second, the fixed and recurring costs of operating and managing a DHPI can be significant; therefore, the team was interested in examining the viability of a shared HPC as a permanent replacement environment versus obtaining a new DHPI.

In past events, C4ISR & Network Modernization had employed the HALLE HPC to host OneSAF and provide simulated mission command information to the live experimental network. As a DHPI, HALLE had been custom configured to fill this role, a non-traditional use of cluster HPC, utilizing the computing power of each node and the high-speed interconnect to run distributed OneSAF simulations.

As was the case in E10, C4ISR & Network Modernization engaged the HPCMP PETTT team to help locate a suitable shared supercomputing resource that operated in an unclassified environment. This led to identification of the MJM and SMQ systems. The processes used during E10 establish user accounts and make system reservations were reused to set up accounts and reservations during E11. Once again, custom virtual private network (VPN) connections were set up between the ARL DSRC, the SMDC Sim Center, and the Center for L/V/C at APG to support real-time simulation traffic, originating from BCT-sized OneSAF v3.0 simulations running on the HPC systems to the E11 live tactical network at Ft. Dix, NJ. Figure 2 depicts the network connections and traffic flows between the HPCs, APG, and Fort Dix, NJ.

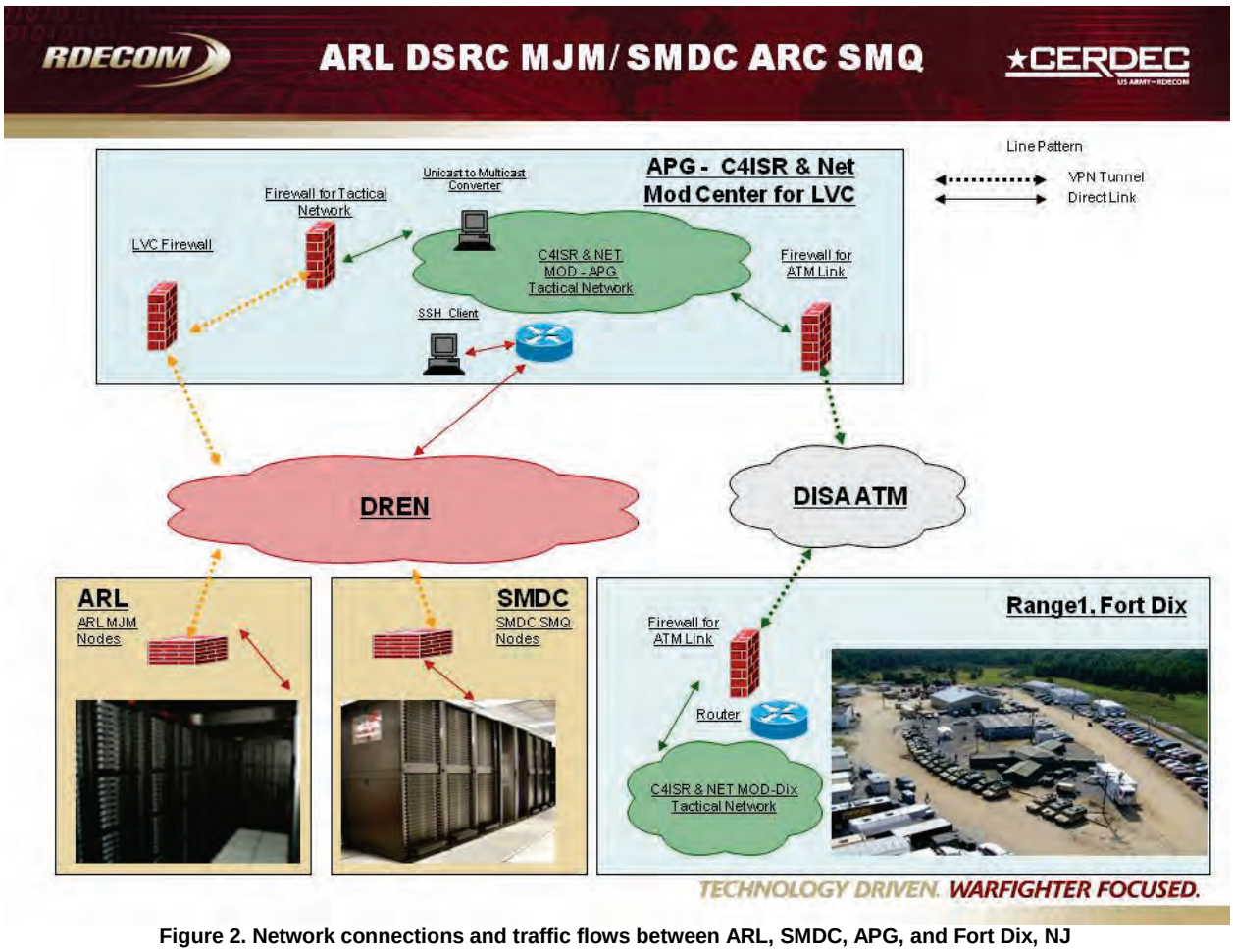


Figure 2. Network connections and traffic flows between ARL, SMDC, APG, and Fort Dix, NJ

OneSAF v3.0 was the baseline installation on both HPC systems. Multiple instances of OneSAF were installed in the Lustre file system, which was mounted on and shared among all the HPC nodes through the Infiniband interconnect, together with a script folder that was used to automate the centrally-controlled startup and shutdown of the OneSAF distributed-exercise across all compute nodes. Based on a OneSAF performance analysis that was performed on multi-core systems, the L/V/C team determined that OneSAF v3.0 performed adequately in multi-threading its child processes and balancing CPU load among all available CPU cores. Figure 3 shows the detailed OneSAF configuration on 10 compute nodes of the MJM system.

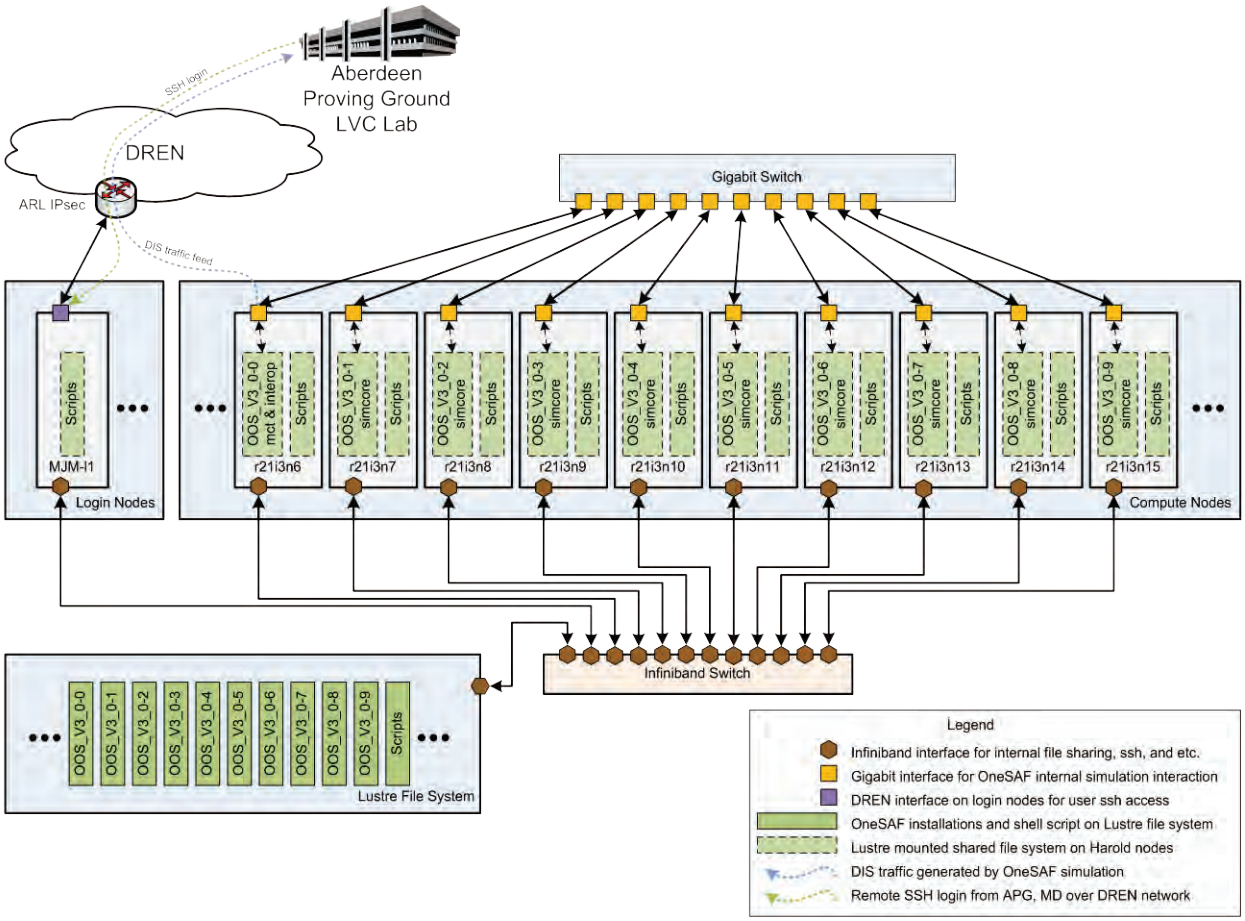


Figure 3. MJM OneSAF Configuration

As in E10, the technical challenges associated with operating a distributed constructive simulation on a shared HPC are small compared to the Information Assurance (IA) and other business process challenges that surface when the simulation must be "long-lined" from a shared resource available to general users to a tactical network classified as a "restricted enclave." Both systems performed well and provided a stable BCT-level scenario that effectively stimulated the live mission command systems. There were no apparent differences in performance between the systems.

5. Lessons-Learned from E11

Shared HPC Employment in a Tactical Experimentation Environment

From a purely technical standpoint, the use of a shared HPC proved fairly simple. There were very few technical challenges to overcome. OneSAF native installation on an Intel-based shared HPC is straightforward. The ability to install software natively reduces the effort required to stay current with OneSAF releases considerably, which allows the L/V/C team to leverage new features and patches in a timely fashion.

The challenges with shared HPC use stem from the security concerns and reliability issues encountered during E11. The L/V/C team was fortunate in that the SMQ system was available as a backup, and required little unique configuration and access work to set up.

Recently, C4ISR & Network Modernization received three small-form factors Cray CX1 DHPI as a result of a joint proposal award with Mobile Network Modeling Institute (MNMI) from HPCMP. It is anticipated that these machines will become the primary means of providing BCT-level simulations during future integrated capabilities events; however, the L/V/C team plans to maintain its working relationship with the HPCMP PETTT to preserve the capability to use shared HPCs as backup or surge assets employing high-fidelity effects models.

6. Significance to DoD

As the Armed Forces of the United States continue their transformation to more agile, responsive and interoperable forces, the ability to conduct SoS experimentation, testing and evaluation at large-scales and high-fidelity becomes more and more critical. The cost in time and other resources to produce and integrate emerging C4ISR SoS components in the quantities required for live experimentation is prohibitive; increasingly, major acquisition decisions will have to be made based on results gained from smaller scale live evaluations augmented by large-scale simulations and emulations, which are, in turn, based on high-fidelity engineering models. C4ISR & Network Modernization has taken the first steps in achieving this end by developing and employing an immersive, integrated LVC environment, based on the non-traditional use of HPC, with the ultimate goal of reducing risk in developing, fielding and deploying new C4ISR technologies in support of Warfighters.

References

Ku, D., Smith, R., Choi, Y., and Sypniewski, J., "C4ISR & Network Modernization E11: Modeling and simulation Final Report."

Early Experiences with Energy-Aware Scheduling

Kathleen A. Smith and Thomas M. Kendall
US Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
{kathleen.a.smith120.civ, thomas.m.kendall4.civ}@mail.mil

William DeSalvo
Instrumental,
bdesalvo@instrumental.com

Michael P. Knowles, Phillip G. Matthews, and James C. Ianni
Lockheed Martin, Aberdeen Proving Ground, MD
{michael.p.knowles8, phillip.g.matthews2, james.c.ianni}.ctr@mail.mil

Chris Sauerwald
Altair, Troy, MI
csauerwa@altair.com

David Cronk
Lockheed Martin, Knoxville, TN
david.c.cronk@lmco.com

Abstract

This paper documents the early experiences and recent progress with employing the Energy-Aware Scheduler (EAS) at the Department of Defense (DoD) Supercomputing Resource Centers (DSRC). The US Army Research Laboratory (ARL) has partnered with Lockheed Martin, Altair, and Instrumental to assess feasibility on current DSR) high performance computing (HPC) systems. Developmental work was completed on the ARL DSRC Test and Development systems and ported to the production systems at the ARL DSRC. The (EAS) is written in Python and works with the current program-wide scheduler, Altair PBS Professional that is deployed across the DSRCs. EAS reduces power and cooling costs by intelligently powering-off compute nodes that are not actively being used by the currently running or reserved for near future jobs. It has been estimated that the Energy-Aware Scheduler could potentially save millions of Kilowatt-hours each year throughout the program. We will describe the extent of our work to date at the DSRC centers and our plans to complete our work by September 30, 2012.

1. Introduction

The Department of Defense (DoD) High Performance Computing Modernization Program's (HPCMP's) Energy Aware Scheduler (EAS) project was initiated as a one-year project under the HPCMP's Green HPC Initiative. The goal of the project is to evaluate the feasibility of reducing power and cooling costs, and the associated environmental impacts, that result from the energy used by idle compute nodes. IDC reports that the power and cooling costs for data centers surpassed the spending on new servers in 2007 and continues to grow as a fraction of data center spending[1]. Clearly, if the HPCMP is to continue to successfully meet the majority of requirements in the Department of Defense, efforts focused on reducing the energy costs need to be sustained.

Through the HPCMP, the EAS was initiated to investigate and develop the ability to save energy by controlling power to the nodes of the various high performance computing (HPC) assets across the DoD Supercomputing Resource Centers (DSRC). This project is a collaboration between the US Army Research Laboratory (ARL), Lockheed Martin, Altair, and Instrumental. The HPC systems across all the centers run on an average 80–90% busy, and it is attractive to reduce the power requirements on the portion of the systems that are idle. This capability is architecture-specific and has been deployed on several systems throughout the DSRCs and has proven to be reliable and effective in reducing power

consumption. The team tested the resiliency and feasibility of EAS on SGI Altix ICE, Cray XE6, and Appro Xtreme (Utility Server architecture) by running benchmarks on these systems with the results published later in this paper. EAS works with the current program-wide scheduler, Altair PBS Professional™ Scheduler, by powering-off resources (nodes) that are idle, to save energy. Typically, idle nodes consume about 50% as much energy as active nodes and it is attractive to power-off resources that are idle to save energy. It is projected that across the entire HPCMP program there is a potential to save millions of kilowatt hours per year. Annual estimated HPCMP kWh savings as a function of the idle percentage and number of hours per day that nodes could be made available for powering off are shown in Table 1.

Table 1. Annual kWh saved

Architecture	10% 12h/D	15% 12h/D	10% 24h/D	15% 24h/D
Cray XE6	224,012	336,018	448,024	672,036
SGI Altix ICE	263,725	395,588	527,450	791,175
Subtotal	487,737	731,606	975,474	1,463,211
Site (1.8 PUE)	390,190	585,284	780,379	1,170,569
Total	***877,927***	***1,316,890***	***1,755,853***	***2,633,780***

We have integrated the Energy-Aware Scheduler (EAS) with Altair PBS Professional™ to control and minimize the power of resources that are not in use. Altair's PBS Professional™ Scheduler already incorporated a capability that was oriented toward power monitoring and control. After performance testing on ARL DSRC Test & Development System (TDS) the original Perl version of the main tool showed that it took at least 20 minutes to complete a scheduler cycle for a 1,000 node system. The script spent a lot of time trying to predict what the scheduler would do, and this consumed considerable time. An additional problem was that the predictions were based on the scheduler's most basic behavior and did not do sorting formula, backfill, strict ordering, node sorting, placement sets and many things that we can expect schedulers in the program to use. Beyond being slow, it may well have been inaccurate for the complexity of the DSRCs. Altair suggested that a Python implementation would be better at delivering on the requirements of EAS. The new version takes the approach of focusing on what the PBS Professional scheduler thinks will happen in the future, rather than trying to make the determination itself. This leaves all the scheduling logic with the scheduler and will adapt as configuration changes are made to the schedule. The only exception is the very basic mechanism for node sorting added for the 'keep minimum nodes online' feature at the tail-end of the project to better support interactive and debugging requirements, especially on the Utility Servers.

2. Results

ARL DSRC SGI Systems

After the initial Python implementation was completed by Altair, the ARL DSRC team began testing the EAS code on a 96 core SGI Altix ICE Test & Development System (TDS) named icecube and saw significant improvements. While analyzing EAS on icecube, several additional features were developed to enhance the code, to include "live" power lock file support that prevents race conditions and a minimum delay between scans to prevent it from running too often. Scripts specific to the SGI Altix ICE were constructed to power up/down nodes with calls initiated through EAS to manage the resources. EAS has 3 modes of operation, passive simulation, active simulation and live power. "Passive simulation" mode executes scripts to make changes to node resources to simulate powering nodes on/off, but do not effect workload scheduling. "Active simulation" mode executes scripts to make changes to node resources to simulate powering nodes on/off and effect's workload scheduling to give a realistic picture. "Live power" mode executes scripts to make all normal resource changes from the other nodes and issues the real commands to power nodes on and off. An additional command utility was built named "pstat" to display a snapshot of the EAS power state of nodes currently being managed by the power management scripts:

In early December 2011, we implemented "live power" mode on the TDS SGI Altix ICE, icecube, and tested extensively before employing on the production system. In mid-December, EAS was configured on Harold, a 10,752 core SGI Altix ICE 8200 system, and began testing in "passive simulation" mode. Following 2 weeks of significant testing on the cluster, we turned "live power" mode on, one rack at a time, starting with the racks dedicated for reservation jobs only. We soon discovered problems on Harold due to a bug where Altair code was dereferencing a pointer inside the python binding that should not have been de-referenced. This eventually led to the crash of the PBS servers. This issue returned

often and manifested when the PBS server daemon restarted its internal Python interpreter. This problem was fixed in 11.x era and went away when 11.2 was installed on Harold. Also, in 10.4.7+ there is a bug with the Python 2.5.1 distribution in the pwd module. Any hook using the pwd module could have a similar de-reference issue as the first bug, but it occurred less frequently. The pwd module was used inside the old SLM/PBS integration hooks only, so removing them fixed the issue. (New versions of these hooks don't use pwd any more). To mitigate these PBS crashes, EAS was configured to only run 18:00 and 08:00 daily.

Table 2. pstat

Name	Power State	Marked idle/Powered off at	PBS state
R1i3n8	Powered-off	OFF: Fri Apr 6 16:35:24 2012	down
R1i0n0	Powered-off	OFF: Fri Apr 6 16:35:24 2012	down
R1i0n1	Powered-off	OFF: Fri Apr 6 16:35:24 2012	down
R1i1n1	Powered-off	IDLE: Fri Apr 6 16:15:16 2012	free
R1i0n8	Powered-off	IDLE: Fri Apr 6 16:20:53 2012	free
R1i0b9	Powered-off	IDLE: Fri Apr 6 16:20:53 2012	free

In early February 2012, Harold was upgraded from PBSPro 10.4.7 to 11.2 and EAS was configured for a second time in "live power" mode 24×7. Perl scripts were developed to generate accounting report details on power-cycled nodes and total number of kilowatt hours saved by EAS. In addition, benchmarks were run on the system before and after the implementation of EAS and these demonstrated no significant performance degradation in job execution times. The results of these benchmarks are published later in this paper. Testing by SGI determined the SGI Altix ICE draws 147 watts from each idle node. From January to February 2012, we saved a total of 6,910 kWh on Harold by powering down idle nodes. Since December 1, 2011, there have been 4 node failures (3 BMC, 1 HCA) as a result of EAS. Perl scripts were developed to generate reports on the number of nodes power-cycled and the total number of kilowatt hours saved by EAS. Savings recognized since February on Harold are in the table below. As Harold is in the middle of its expected lifecycle, the available idle time is not as high as is typically experienced on HPCMP systems in the early and late stages of their lifecycles. It is expected that the savings will significantly increase as Harold approaches and enters year four of its lifecycle.

Table 3. Harold kWh saved

Dates	EAS hours saved	EAS Hours saved × 147w
02/06– 02/12	3971	583 kWh
02/13–02/19	4282	629 kWh
02/20–02/26	5240	770 kWh
02/27–03/04	2597	381 kWh
03/05–03/11	9084	1335 kWh
03/12–03/18	2331	342 kWh
03/19–03/25	6370	936 kWh
03/26–04/01	1116	164 kWh
04/02–04/08	1595	234 kWh
04/09–04/15	1360	199 kWh
04/16–04/22	3734	548 kWh
04/23–04/29	1780	261 kWh
TOTAL		***6388 kWh***

During weekends, Harold typically has additional nodes available for power-down, as most of the eligible "backfill" type jobs have completed. The backfill jobs have wall-clock time of less than 24 hours, and can also run on the reservation nodes of the cluster. When 'large' jobs are submitted, the scheduler will try to keep nodes idle in order to prevent backfill jobs from pushing the top job start time out. While these nodes are idle, they are subject to being shut down by power management to reduce wasted energy. Typically jobs on Harold are 64 nodes or less. Large jobs are typically 256 nodes or higher.

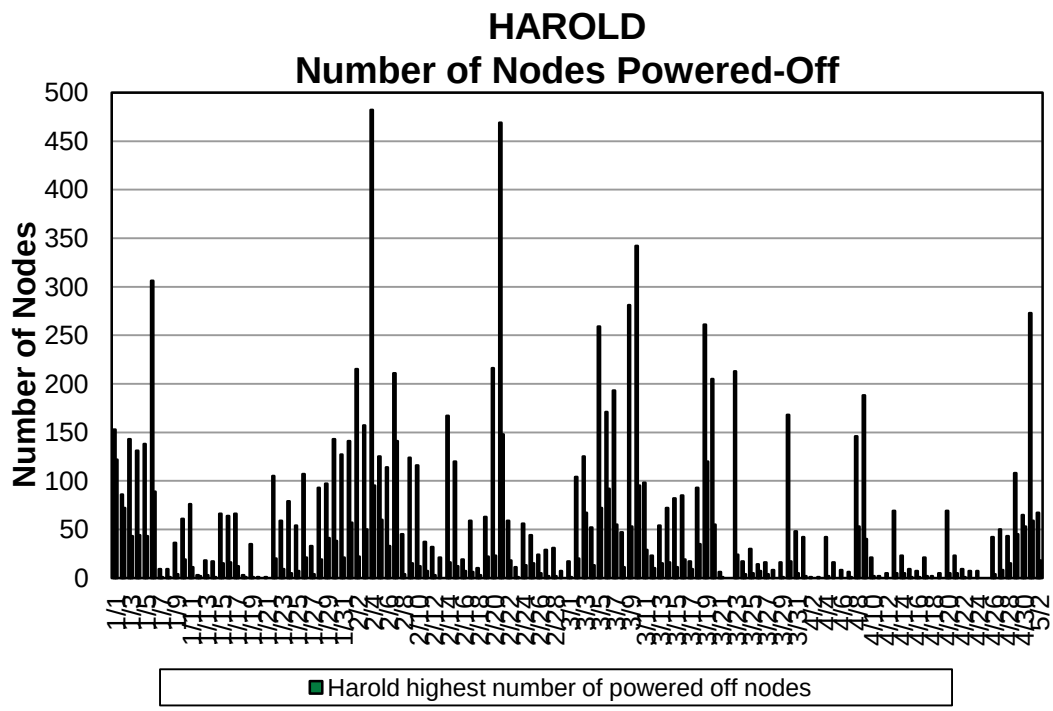


Figure 1. Harold

In early March, EAS was configured on a second SGI Altix ICE system, TOW, a 6,656 core SGI Altix ICE 8200 system. TOW was first configured in "passive simulation" mode and then slowly transitioned over to "live power" mode, one rack at a time. On this system, we saved 9,893 kWh in the month of March alone by powering-down idle nodes. TOW runs a significantly different job mix than Harold and after a month of running "live power" mode we started to notice a trend in jobs randomly dying on start-up. EAS was shutdown to rule out the power monitoring process and jobs returned to running as expected, without dying at start-up. The problem was diagnosed to be associated with a Lustre issue when starting nodes. SGI is investigating the problem, and has implemented a Lustre patch to improve the reliability of node start-up. Energy savings with EAS on TOW are in the Table 4.

Table 4. TOW kWh saved

Dates	EAS hours saved	EAS Hours saved × 147w
03/03–03/04	46	6 kWh
03/05–03/11	3614	531 kWh
03/12–03/18	6424	944 kWh
03/19–03/25	11810	1736 kWh
03/26–04/01	2983	483 kWh
04/02–04/08	3214	472 kWh
04/09–04/15	0	0 kWh
04/16–04/22	0	0 kWh
04/23–04/29	0	0 kWh
TOTAL		***4172 kWh***

On TOW, there is a significant opportunity for potential power savings, mainly due to the large-core-count jobs that run on this system. Since the large jobs typically take time to accumulate enough nodes to run, we have a large amount of nodes that can be powered-off. Also, given that new jobs are typically not submitted during this weekend on this system we have additional idle nodes that are candidate for power-down as can be seen in Figure 2.

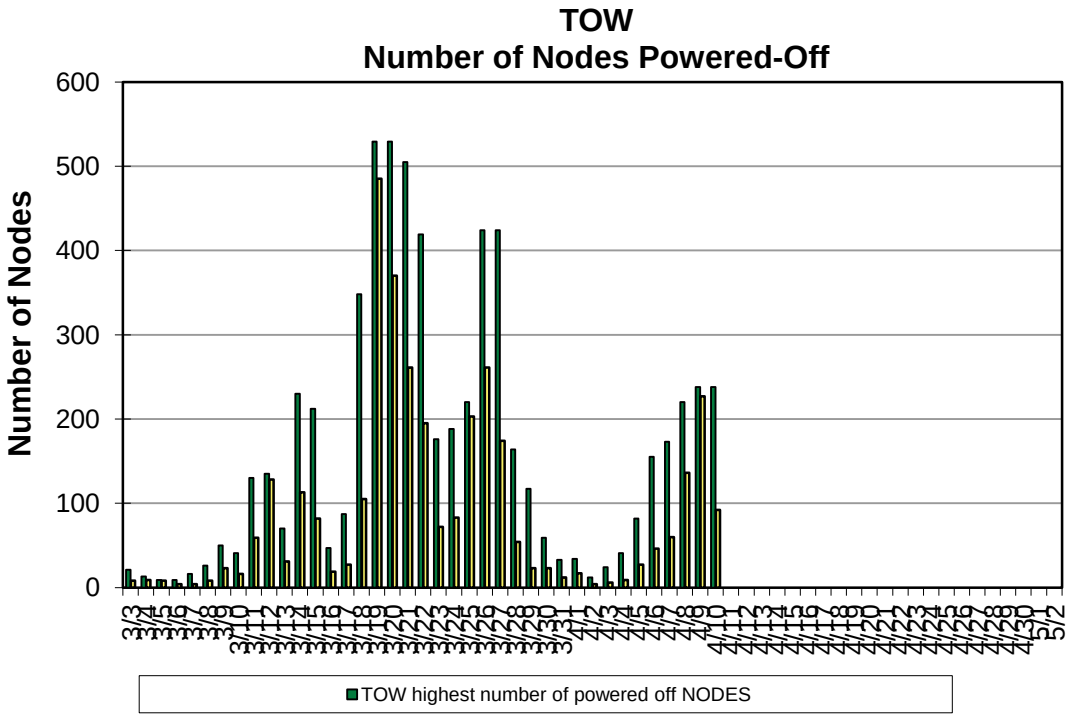


Figure 2. TOW

MHPCC DSRC Dell System

During the week of March 12, 2012, Chris Sauerwald, Altair and Bill DeSalvo, Instrumental, visited MHPCC to configure EAS on both, Mana 9,216 core Dell PowerEdge system and the Utility Server. EAS was configured in "active simulation" mode. Since reservations are handled differently on these systems, Altair built a special PBS version, **11.2.2**, to support Maui's requirement for reservations that allows reservations to cross dedicated time boundaries. Because of the large percentage of nodes in use by reservations, a feature was added to EAS that allows any reservation longer than X seconds to have their nodes powered on ONLY if jobs need them. If reservations are less than X, their nodes will be powered-on in time for the reservation start. From March 13–31, the virtual savings on Mana were 203,711 hours for a total of 30,556 kWh and a dollar savings of approximately $11,000 at $0.37/kWh. From April 1-17 there were 261,207 hours of virtual savings. Mana was enabled in "live power" mode on April 25 and during the first day alone, there were 935 nodes powered-off by EAS, about 80% of the system. The estimated dollar savings are $25,000 for the month of April.

Utility Servers

We have integrated EAS on 3 out of 6 Utility Servers across the DSRCs, 2 at ARL DSRC and 1 at the MHPCC DSRC. The Utility Servers have a requirement for a given number of nodes to always be powered-on and available for user jobs. Subsequently, Altair updated the EAS code to support this requirement. The Utility Servers are fairly new to the program and, as such, not highly utilized; there is an excellent opportunity to power-down idle nodes. The early numbers suggest a savings of about 27,000 hours a month on the unclassified Utility Server. On the unclassified Utility Server there is a good balance between powered-off nodes, and current workload on the system as can be seen in Figure 3.

We have begun to integrate the classified Utility Server with EAS. The graph in Figure 4, displays how we have gradually added nodes, ran for several weeks, and upgraded the system on Apr 11, 2012, then disabled EAS to allow for further application testing of the cluster by the support staff. We plan to enable EAS on only the weekends in order to provide for uninterrupted user access during the weekdays until we can install the 0330 version of EAS.

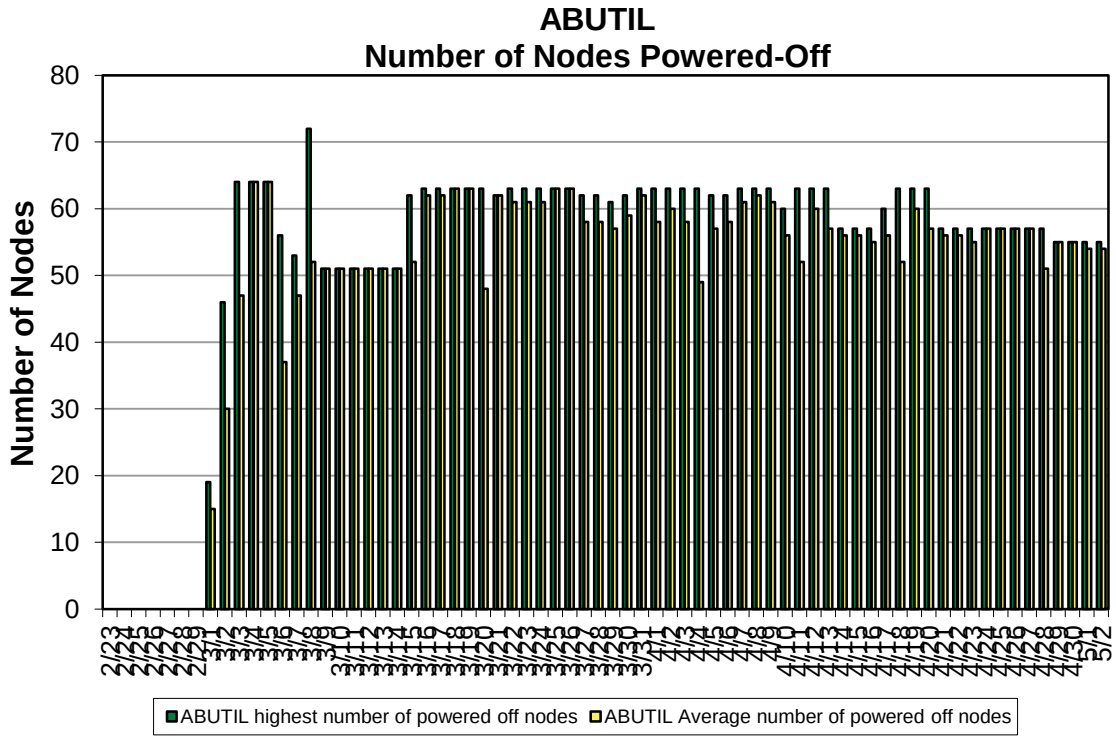


Figure 3. Abutil

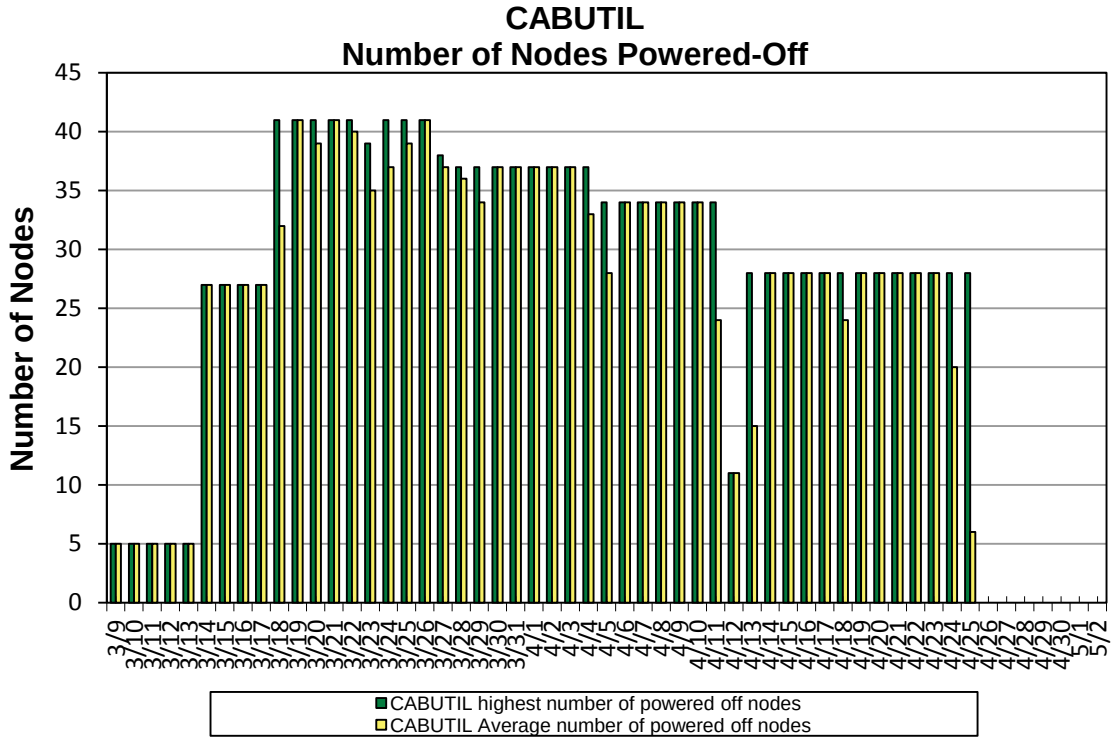


Figure 4. Cabutil

Cray XE6

We began to investigate the potential cost savings using EAS on a Cray XE6. The early testing revealed promising results. On a 16 compute node Cray XE6, Tana, at the Arctic Region Supercomputing Center (ARSC), Dr. James C. Ianni, Lockheed Martin, and Liam Forbes, University of Alaska Fairbanks, tested compute node power utilization and the possible impacts of idle nodes shutting-down and booting during running jobs. The first test was to shut-down all idle compute nodes while an 8 node GAMESS job was running. The job was configured to use half the cores and all memory on each node. The test was performed 3 times. The second and third execution completed successfully; however, on the first execution, the running job ended after running only 7 minutes 50 seconds with an exit status of 0. The cause of the job failure was unable to be determined. The next test case was to boot 8 compute nodes while the same GAMMESS job was running. This test was also executed three times with all tests completing successfully. The job runtimes during all tests did not vary substantially, except for the one case. On Tana, we don't believe we are testing at a scale to demonstrate possible impacts on job runtimes, and are working to test larger cases and/or different applications on one of the larger Cray XE6s in the HPCMP to investigate possible impacts. Other similar tests were performed to get baseline measurements and to try powering-down larger portions of the system (for example a blade). Based on those tests, we concluded that if EAS is implemented on a Cray XE6, the focus should be on manipulating individual compute nodes. Trying to manipulate blades, cages, or cabinets "breaks" the system interconnect, either killing running jobs or crashing the system. Liam Forbes collected power measurements during all tests and calculated that a Cray XE6 compute node draws approximately 115 watts at idle. There are minor hardware differences between the XE6 compute nodes at the different DSRCs, so some minor verification tests should be run on each system, but we think this is a reasonable estimation for calculating possible cost savings based on average numbers of idle nodes on each system. Discussions with Cray, Inc. support and engineering regarding impacts and possible mechanisms to implement EAS are ongoing.

Benchmarks

Benchmarking can be a measure of any possible degradation of the systems network during a computational node disruption through power cycling. The DSRC has a dedicated suite of software (the Sustained Systems Performance suite or SSP) to provide an adequate benchmarking test on large super-clusters. The Energy-Aware Scheduling (EAS) system was installed on a test system. Due to the small size of the test systems, the full SSP system was not utilized. Due to this constraint an alternative to the SSP benchmarking suite was utilized.

Those alternatives were SKaMPI[1] and a custom input to the GAMESS computational chemistry system[2]. SKaMPI is a benchmark for testing the MPI communications system and can do measurements of point-to-point communication, collective communication MPI operations and many others. Most of the MPI calls with various sized send/receive MPI buffers are called. The total time for the entire SKaMPI suite run is recorded as one benchmarking time. The GAMESS run for this benchmark utilizes the second-order energy correction of Moller-Plosset perturbation theory (MP2) with the molecule benzoquinone. A geometry optimization calculation was performed on the benzoquinone. The basis set employed is Pople's double-zeta basis with an additional d-polarization function on the heavy atoms (6-31G*).

The SKaMPI Benchmarks were initially run on Harold before and after the installation of EAS on a subset of Harold. The runs were performed on 64 cores (8 nodes, 2 GB/core) under OpenMPI-1.4.1. As shown below, there was a positive significant change in the timings of the SKaMPI runs. This is probably not due to the installation of EAS, but the reconfiguring of Lustre to use MPI-I/O locking between the December 14th and January 10th time period.

The SKaMPI Benchmarks were also run on the ARL Utility Server before and after the installation of EAS. The sam benchmarks were also executed on a small subset of Harold. The runs were performed on 64 cores (8 nodes, 2 GB/core) under OpenMPI-1.4.1. As shown in Table 5, there were no significant changes in the timings of the SKaMPI runs.

Table 5.

Machine	Run Date	Start	End	Elapsed Time
Harold				
	Wed Dec 14 18:48:11 EST 2011	6:48:11	8:03:23	1:15:12
	Tue Jan 10 12:57:11 EST 2012	12:57:11	13:40:58	0:43:47
	Thu Jan 12 12:35:59 EST 2012	12:35:59	13:25:43	0:49:44
ARL Utility Server				
	Wed Dec 14 15:56:39 CST 2011	15:56:39	17:17:17	1:20:38
	Tue Jan 10 13:06:17 CST 2012	13:06:17	14:32:56	1:26:39

A decision was made to switch to a more realistic test case for the ERDC machines. In this case, the above-mentioned GAMESS system was executed. Although the EAS system was not installed on the ERDC test system for Chugach, named Tana, benchmarks were run before and after nodes were power-cycled to simulate what the EAS system would perform. Those runs are shown in Table 6. The first two runs are the reference runs, where none of the nodes in the entire Tana test system were power-cycled. Runs 3 to 16 represent a series of runs where Tana nodes were power-cycled. With the exception of run 11, most runs did not deviate too far from the 1,059 second reference runs. The average and standard deviation of those runs are 1,062 +/- 4 seconds indicating almost consistent timings. The outlier at run 11 could be attributed to a transient network error in the Tana test system when that job ran. The output had a "CqWaitEvent failed in Wait; err 11" in the output. This network error comes from Cray's GNI (General Network Interface) CqWaitEvent() function and not from MPI or from the GAMESS code. This error was not repeatable.

Table 6.

Run #	GAMESS Run *(sec)*
1	*1059*
2	*1059*
3	1064
4	1061
5	1059
6	1070
7	1058
8	1065
9	1057
10	1068
11	466
12	1063
13	1059
14	1062
15	1057
16	1064

3. Significance to DoD

The Energy-Aware Scheduler has proven to be feasible and reliable on the different architectures that we have tested, and measured powered savings are encouraging.

4. Conclusion

In this paper, we gave a brief overview of the efforts to deploy the Energy-Aware Scheduler across the HPCMP. Implementing the Energy-Aware Scheduler on the ARL DSRC systems has thus far saved 1,000's of kWh and also has the potential to save in reducing cooling requirements, with very little fallout from hardware failures. We are in the process of working with the other DSRCs and Original Equipment Manufacturer vendors to further expand this capability. We have heard many concerns while implementing EAS that we are working to address These concerns range from the number of times a blade can be powered up/down, system stability; increase in System Administrator workload due to troubleshooting increased on the nodes that are powered-off. With this enhanced power-efficiency we can lower costs at DoD centers. Early results have proven that there is the potential to save the projected millions of kWh throughout the HPCM Program. This project is scheduled to be completed in September 2012.

Acknowledgements

This effort was supported in part by the HPCMP's Green Initiative. The authors would like to thank Liam Forbes for all his research that he did on Tana. We would also like to thank MHPCC for their work in implementing EAS. We look forward hearing your success stories with EAS.

References

1. IDC Report, *The Impact of Power and Cooling on Data Center Infrastructure*, http://www-03.ibm.com/systems/cn/resources/systems_cn_IDC_ImpactofPowerandCooling.pdf.

2. *SKaMPI*, Werner Augustin and Thomas Worsch, April 7, 2008 version, http://liinwww.ira.uka.de/~skampi/.

3. *GAMESS, Version 24*, Mar 2007 (R6), Iowa State University, M.W. Schmidt, K.K. Baldridge, J.A. Boatz, S.T. Elbert, M.S. Gordon, J.H. Jensen, S. Koseki, N. Matsunaga, K.A. Nguyen, S.J. Su, T.L. Windus, M. Dupuis, and J.A. Montgomery, *J. Comput. Chem.*, 14, pp. 1347–1363, 1993.

OneSAF as a Simulation Service using High Performance Computing

Christina Bouwens
Science Applications International Corporation, Orlando, FL
christina.bouwens@saic.com

Amy Henninger
Center for Army Analysis (CAA), Arlington, VA
amy.henninger@us.army.mil

Gloria Flowers and Alesya Paschal
US Space and Missile Defense Command (SMDC), Huntsville, AL
gloria.flowers@smdc.army.mil, alesya.paschal@us.army.mil

Abstract

In recent years, the US Army Modeling and Simulation (M&S) community has increased its use of high performance computers (HPCs) to enable high-fidelity forces modeling support to a number of research, development, test and evaluation (RDT&E) initiatives. Because of OneSAF's flexibility and portability, it has been hosted on a number of HPCs over the past 6 years. Each implementation is unique as each HPC is unique in its architecture. Since 2009, the OneSAF community of HPC users has been collaborating to introduce improvements and additional capabilities to OneSAF to allow it to make better use of HPC resources and architecture, while providing an easy to use interface for users to access the simulation on HPCs. This paper provides an overview of current efforts to improve OneSAF's use of HPCs and to provide OneSAF simulation as a service via the HPC portal environment being developed at the DoD High Performance Computing Modernization Program's Maui HPC Center.

Keywords: OneSAF, Computer Generated Forces, High Performance Computing, Parallel Computing, Software as a Service

1. Background

1.1 Forces Modeling and Simulation (FMS) with High Performance Computing (HPC)

Forces modeling and simulation (FMS) systems are simulations that represent interacting groups of entities. These entities are primarily associated with military-type forces, but they may also represent non-military entities that may be present within an area of interest. FMS may be represented by live, virtual or constructive simulation systems.

The FMS systems have been developed using commercial-off-the-shelf (COTS) products such as personal computers, which are attractive because of their affordability. However, the need for higher fidelity, more complex representations with increased numbers of entities has made the use of highly-parallel high performance computer (HPC) systems attractive.

Today's HPCs, sometimes referred to as supercomputers, are massively-parallel systems using thousands of ordinary computational processing units (CPUs). These CPUs are connected together by a high-speed network with several types of shared memory options. The operating system of choice is usually Linux® (Linus Torvalds) or some form of Unix® (X/Open Company). These systems are usually configured as shared systems, processing batch jobs submitted by the users.

1.2 Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP) Background

"The High Performance Computing Modernization Program (HPCMP) was initiated in 1992 in response to congressional direction to modernize the Department of Defense (DoD) laboratories' high performance computing (HPC) capabilities. The HPCMP was assembled out of a collection of small HPC departments, each with a rich history of supercomputing experience

that had independently evolved within the Army, Air Force, and Navy laboratories and test centers."[1]

The DoD HPCMP sponsors the User Productivity Enhancement, Technology Transfer and Training (PETTT) program that has the mission to:

- Enhance DoD HPC user productivity
- Provide world-class technical support and training
- Focus HPC technologies on key DoD strategic areas
- Foster outreach and collaboration with military service, agency and national HPC communities

Over the past eleven years, the PETTT program has helped to bridge the gap between the FMS and HPC communities by providing expertise and funding to explore the key challenges of FMS implementation in the HPC context.

1.3 Challenges for the FMS Community

The efforts of the HPCMPO that provides HPC infrastructure (Defense Research and Engineering Network (DREN) and DoD Supercomputing Resource Centers (DSRC)) and support PETTT has made HPCs more accessible to the FMS community, but many challenges remain before the community can take advantage of the full benefits that HPCs offer.

1.3.1 Hardware Architecture Issues

Most FMS applications have been developed for execution on uni-processor systems in a network of workstations mode. The highly-interactive simulations are not well-suited to take advantage of the highly-parallel nature of HPC systems that almost always operate in a batch mode setting. Additionally, many of the HPC systems tend not to have multiple network connections necessary to support a large number of concurrent user interface sessions required for these interactive simulations. Constructive simulations are better suited for HPC systems, but they typically run as a single threaded process, or are limited by the platform operating systems (OS) they support. Many of these simulations operate within the WinTel (Microsoft Corporation's Windows®, Intel Corporation's Intel®) platform, which utilizes standardized hardware architecture. In contrast, there is no universal architecture for HPCs. Software that is developed for, or installed and configured to run on, one HPC system will not necessarily run on another system.

1.3.2 Application Architecture Issues

As the FMS community grew, it sought to expand the capability and scope of FMS applications, particularly human-interactive simulations, through the use of distributed technologies such as distributed interactive simulation (DIS) and the high-level architecture (HLA). The use of these technologies introduced the requirement to maintain simulation-state consistency across distributed systems and added network communications overhead and system complexity. Delays and bottlenecks were experienced from network transport. Even moving this paradigm to the HPC environment simply moves the bottleneck of communications from the network transport to inter-memory communication between processors. Similar limitations exist, even for HPC implementations. Probably a more significant concern is that this approach reinforces a high-resolution, entity-centric view of the world. The association of real-world vehicles and platforms (tank, plane, automobile, etc.) to simulation entities that operate on a single processor, while minimizing the inter-processor communication, requires that whole entities are simulated on a single processor.

This focus on platforms or physical entities does not take into account organizational or cognitive structures that have a significant impact on properly modeling the engagement environments found in the world today. To get around this limitation, the FMS community has resorted to constructive simulations with lower resolution than entity-centric models, but with greater representation of operational command-and-control behaviors. However, these applications have not been architected to support multi-threading within an HPC environment. In recent years, user requirements for increased scope and depth of representation have driven even these models to such detail that time required from scenario development to completed post-processing is a challenge. Such modeling requires much more computational power and needs to be performed within an architectural construct that is more adapted to the parallel environment offered by HPCs.

1.3.3 FMS Adoption of HPC is Slow

Given the hardware and application issues, the FMS community has been reluctant to move to HPC systems. The effort required to get things working is more than the community has been willing to invest. Some of the issues include the fact that:

- HPC user interfaces are not intuitive for the FMS community (still command line/terminal focused).
- There are many HPC systems with little commonality between architectures – lack of portability between systems.
- A large number of FMS models exist:
  - Models built by model developers and analysts who are good programmers, but unfamiliar with creating code that can implement parallelism
  - Many models have been built in the Windows domain and not easily portable to current DSRC HPC systems (which are primarily Unix-like)
  - There is a lack of available off-the-shelf tools, best-practices and documentation for the FMS community

1.3.4 FMS HPC Experience is Growing

FMS simulation has made progress in its implementation of HPC for a variety of applications to support test and evaluation, analysis, and training. OneSAF® (Department of the Army) has been used in several instances of FMS HPC applications. The rest of this paper will focus on OneSAF's implementation on HPC, and the current effort to make it more accessible through a "Simulation-as-a-Service" construct.

1.4 OneSAF Background

1.4.1 OneSAF Modeling and Simulation (M&S) Support

OneSAF is the Army's next-generation entity-level simulation that provides a composable, distributed and scalable simulation of real-world battlefield situations using validated physical models and doctrinally-correct behavior models. It can support analysis, acquisition, planning, testing, training, and experimentation. OneSAF allows users to compose a wide-range of complete simulation systems from a set of component-based tools, develop new or extend existing tools, as well as compose new single or multi-resolution entities, units, and associated behaviors from existing physical and behavioral software components.

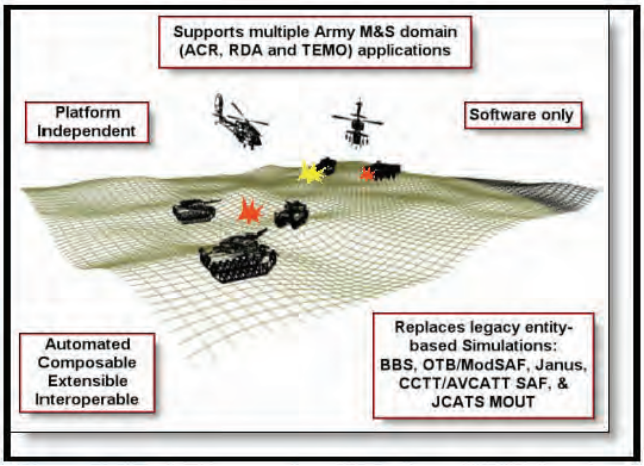


Figure 1. OneSAF supports multiple Army M&S domains

OneSAF also accurately and effectively represents activities within the Army warfighting functions to include:

- Intelligence
- Movement-and-maneuver
- Fire support
- Protection
- Sustainment
- Command-and-control

1.4.2 OneSAF Scalability Dimensions

OneSAF hardware/system performance requirements are impacted by a number of distributed simulation-related factors. For this paper, we explore two dimensions of scalability: fidelity and entity count. Increases in either dimension require additional hardware and communications resources to properly support the forces representation in the OneSAF model.

Fidelity Dimension

OneSAF supports variable levels of fidelity, making it possible to tailor the simulation in order to maximize satisfaction of diverse use cases. Here, fidelity refers to the faithfulness of the model to the real-world object being modeled. Often fidelity is equated to model detail and computational requirements, as they are typically, but not necessarily, related. For most entities, units, behaviors, and even terrain, OneSAF supports three levels of fidelity: low, medium, and high.

For example, low-fidelity dismounted infantries (DIs) move, sense and shoot, but they only move in a simple pattern along routes designated by the simulation operator and will not avoid obstacles. Medium-fidelity OneSAF entities will also move, sense and shoot, but they have more sophisticated mobility, sensor and weapon models than their low-fidelity counterparts. So, for example, medium-resolution DIs will exhibit more realistic movement behaviors along routes and will avoid obstacles. High-fidelity entities have all the capabilities and attributes of medium-fidelity entities; however, they are enhanced in some way. They might have a high-fidelity missile fly-out model, rather than Army Materiel Systems Analysis Activity (AMSAA)-provided probability of hit/probability of kill (Ph/Pk) values. They might have sensor models that have greater detail, or they might have a higher-fidelity mobility model. OneSAF's composable architecture provides for complete fluidity among the various physical and behavioral models that can be instanced as a part of an entity. For instance, the medium-fidelity mobility model can be freely combined with the low-fidelity communication model and a high-fidelity acquisition model to create an entirely new, customized entity. There are no rigid rules that determine the fidelity of any given entity; it is up to the creator to assess and properly define them. For this reason, it is impossible to easily estimate global performance characteristics of entities based on their stated fidelity alone.

Entity Count Dimension

Entity count is tied more specifically to the size of the force being represented. However, entities may be used to represent more than just dismounted infantry or vehicles. When used to also represent communication devices (such as radios or cell phones) and sensors, the entity count dimension can quickly grow with increased fidelity in battle environment representation.

Different applications require different fidelity/entity count requirements. Staff training, course-of-action (COA) development and high-level COA analysis can be supported using low-fidelity, high-entity-count simulations. Medium-fidelity simulations are more suitable to support battlefield functional area (BFA)-specific training, detailed COA analysis, mission rehearsal, mission execution monitoring, and some types of experimentation. They can also support concept definition and tradeoff analyses. High-fidelity, low-entity-count simulations, on the other hand, work well to support detailed analyses appropriate for research, system tradeoff analyses, etc.

OneSAF provides the option to support a mixture of entity fidelities in the same exercise. The user can select medium- or high-fidelity entities in areas where detailed modeling of battlefield conditions is needed, but use low-fidelity entities elsewhere as "wrap-around" to fill in the rest of the battlespace. This allows the computational resources to be focused on modeling items of interest in the simulation scenario. This composable approach allows users to tailor the simulation to meet their specific domain requirements.

High performance computing offers the OneSAF M&S community a chance to increase the simulation representation along both dimensions of scalability, where workstation or network resources currently limit the simulation ability to scale and force the simulation architect to limit one dimension or the other.

2. OneSAF HPC Implementations

Since 2009, the OneSAF community of HPC users has been collaborating to introduce improvements and additional capabilities to OneSAF to allow it to make better use of HPC resources and architecture while providing an easy to use interface for users to access the simulation on HPCs. High performance computing offers the OneSAF community a chance to increase the simulation representation along both dimensions of scalability, where workstation or network resources currently limit the simulation ability to scale and force the simulation architect to limit one dimension or the other.

OneSAF's composable software framework makes it a prime candidate for implementation in the HPC environment. Although OneSAF out-of-the-box is not able

to take full advantage of the HPC environment, recent work with its implementation has made the application more compatible with use on super computers.

OneSAF Node Requirement by Exercise Size

Small-scale exercises involve less than 20 total nodes:

- up to 5 simulation Processing Nodes (simcores)
- 1–15 Management and Control Tools (MCTs)
- 1 C4I Adapter,
- 1 After Action Review Tool (AAR)s, and
- 1 System Configuration and Asset Management Tool **(**SCAMT**)**

Medium-scale exercises involve between 20–30 total nodes:

- 5–10 simulation Processing Nodes (simcores)
- 20–25 Management and Control Tools (MCTs)
- 1 C4I Adapter,
- 1 After Action Review Tool (AAR)s, and
- 1 System Configuration and Asset Management Tool **(**SCAMT**)**

Large-scale exercises involve up to 75 total nodes:

- 20 simulation Processing Nodes (simcores)
- 40–50 Management and Control Tools (MCTs)
- 1 C4I Adapter,
- 1–2 After Action[4]

Several implementations of OneSAF on an HPC exist. These successful implementations affirm the suitability of OneSAF on an HPC, requiring minimal staffing. It also highlights the ability to use the HPC platform to conduct distributed operations. And, it allows for graceful scaling of performance with distribution of increased loads over multiple processors. It also introduces a framework that can be extended to incorporate any number of "hard" (e.g., high-fidelity physics) and "soft" factors (e.g., hunger, attention, etc.).

2.1 Army Constructive Training Federation (ACTF)

OneSAF was implemented on a Linux cluster, "Eagle," at the AFRL DSRC. Eagle is an SGI International Corporation, Inc.'s Altix® 3700 consisting of four 512-processors nodes of Intel Corporation's Itanium® two (2) (1.6 GHz) processor[1].

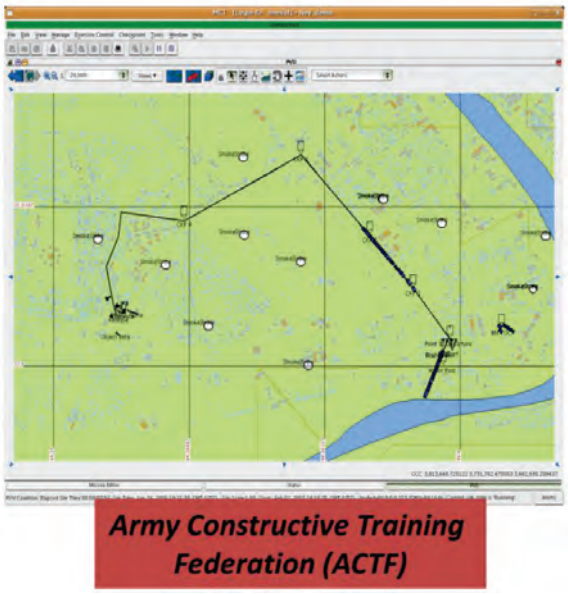


Figure 2. Early OneSAF HPC implementation for ACTF

[1] This HPC system has since been retired from service and is no longer in use at AFRL.

One issue encountered was that the OneSAF application would “hang” at different locations on the Eagle compute node. Both implementations were with early versions of OneSAF and executed in an HPC environment which was early in its support for interactive applications.

2.2 Physics-based Environment for Urban Operations (PEUO)

This effort used the OneSAF baseline, and previous work porting OneSAF to Linux-based HPCs to support an experimental simulation environment that combined physical, logical and behavioral models. This environment enabled the team to understand war and its consequences in context (e.g., improvised explosive devices (IEDs), urban combat, smoke, loss-of-signal), at high-resolution and fidelity. The effort focused on creating an operationally-relevant urban combat test bed that integrated high-fidelity emulation of physical and electromagnetic environments with behavioral, organizational, and cultural simulation along with real-world sensors and events. Physics models included command, control, communications, computers, intelligence, surveillance and reconnaissance (C4ISR) and three-dimensional plume dispersion, where OneSAF provided operational context. OneSAF entities were required to react to the parameters of the models by utilizing sensors and reporting detections based on sensing. Plume effects were also added. The environment allowed for the simulation of entity reaction to smoke and nuclear, biological and chemical (NBC) clouds, going from mission-oriented protective posture (MOPP) level 0 to 1 with the smoke and from level 0 to 4 with the NBC cloud. Wind-shift effects on the smoke plumes were also represented with the high-fidelity physics model.

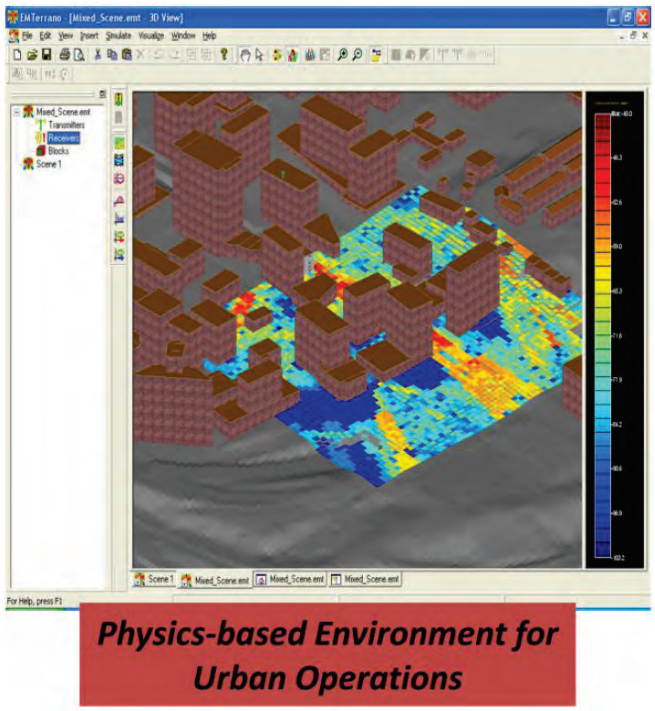


Figure 3. Integration of high-end physics simulation within an urban environment

The resulting integrated environment demonstrated that OneSAF could be integrated with higher fidelity physics models in order to provide an enhanced environment for testing and training.

2.3 C4ISR On-the-Move (OTM)/C4ISR & Network Modernization

CERDEC C4ISR On-the-Move (OTM) Experiment 2008–2011 (E08, E09, and E10) and C4ISR & Network Modernization 2011 (E11) also used OneSAF on an HPC to support their experimentation efforts. This effort provides a relevant environment/venue to assess emerging capabilities in a C4ISR system-of-systems (SoS) configuration to enable a network-centric environment in order to reduce and mitigate risk for Future Force concepts and capabilities, accelerate technology insertion into the current force, and support Army Brigade Combat Team (BCT) Modernization and the Future Force.

This technology demonstration creates virtual world support using OneSAF on the HPC to support live-range assets located at Fort Dix, New Jersey. OneSAF is used as a modeling and simulation driver for the experiments – providing a real-time, brigade-sized stimulation of live assets.

Figure 4. C4ISR OTM Experiment M&S Workstation

Originally, the 2008 and 2009 C4ISR OTM exercises used a standalone HPC called HALLE-HPC Army Laboratory for live/virtual/constructive (LVC) Experimentation to support the exercise. In 2010, the exercise migrated to the Army Research Lab's (ARL) DSRC Harold. Harold is a SGI Altix ICE 8200 Linux cluster. It offers 10,752 Intel Xeon 5500 series processor cores and higher memory bandwidth than other DoD HPCMP HPC, enabling more of the peak processing capability to be delivered with less code optimization. The use of ARL DSRC-Harold proved that HPCs could support distributed exercises. The July 2011 exercise used another ARL DSRC HPC asset-the MJM machine, a 4,400-core Linux Network Advanced technology Cluster. With MJM's September 2011 decommission, last year's exercise also relied on the SMDC/ARSTRAT Simulation Center (SimCtr) HPC asset-SMQ, a Linux Cluster as a backup to the experiment.

For the experiment, these OneSAF simulated entities are inserted as battle command messages and sensor information at multiple echelon-specific "injection" points. Additionally, the simulated entities represented two Brigade Combat Teams (BCTs), and provided scale that could not be achieved with live assets alone.

2.4 OneSAF HPC Support for Space

The US Army Space and Missile Defense Command /Army Forces Strategic Command (USASMDC/ARSTRAT) conducts space and missile defense operations and provides planning, integration, control and coordination of Army forces and capabilities in support of US Strategic Command missions (strategic deterrence, integrated missile defense, and space operations); serves as the Army force modernization proponent for space, high-altitude and global missile defense; serves as the Army operational integrator for global missile defense; and conducts mission-related research and development in support of Army Title 10 responsibilities.

Space M&S representation requires large computation power to represent:

- Friendly-force tracking through a GPS constellation
- Space communication, extending the terrestrial network through space
- Intelligence, Surveillance and Reconnaissance (ISR) that space enables
- Visual images that space provides

OneSAF on an HPC provides the scale of entities required for space representation. USASMDC/ARSTRAT current HPC SimCtr already provides missile defense (MD) research design and analysis of complex MD systems with state-of-the-art computational and simulation resources by providing shared government-furnished property high performance computational (HPC) assets. This facility is the Army's premier center for air, space, and missile defense high-performance computational support. It works to provide the tools, technologies, and expertise needed to realize both the USASMDC/ARSTRAT and the Missile Defense Agency vision for delivering air, space, and missile defense. Using these assets, the USASMDC/ARSTRAT's has been exploring the implementation of OneSAF on HPCs for a number of years. Initially, OneSAF v1.5 was installed on an Altix 3700. OneSAF is currently hosted on one of USASMDC/ARSTRAT's HPCs. This machine has 12 Core AMD (Advanced Micro Devices, Inc.) Istanbul 2.2 GHZ processors, and 16GBs of memory per node.

USASMDC/ARSTRAT's experience highlights one of the challenges with hosting FMS software on shared HPCs – finding support for specific software requirements (in this case Java), while continuing to provide support to other HPC users. This effort also highlighted the expansion of HPC from strictly supporting scientific code to also hosting Java-based code.

2.5 Operationally-Relevant Environment for Testing Ad Hoc Wireless Sensor Networks

The US Army Redstone Test Center (RTC) Distributed Test Control Center (DTCC)/HPC supports T&E and experimentation activities in the facilities. Full system life-cycle support is given to the RTC customers. The DTCC/HPC facilities provide a complete live, virtual, constructive (LVC) environment to immerse test items so they perceive and respond to the stimuli just as they would in tactical environments. The DTCC/HPC facility has computing resources, simulations, network emulations, data acquisition systems, displays, virtual warfighter machine interfaces, and test control assets that operate in Unclassified and Secret configurations depending on the need. The capabilities are connected to high-performance nationally/internationally distributed test networks operating from Unclassified to Top Secret/Sensitive Compartmented Information (TS/SCI).

The DTCC/HPC uses OneSAF on its HPC to provide an operationally-relevant environment for realistic testing of wireless tactical networks. These include networked sensors and sensor fusion systems, which require the network environment to stimulate the system. Providing real-time network emulation with interfaces to real-systems forces perception and response as it would in the real-world.

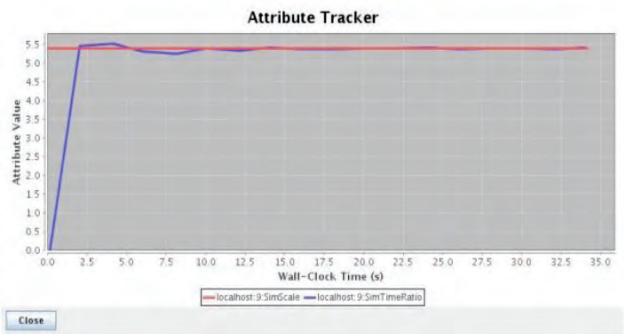


Figure 5. The Remote Distributed Performance Monitor

2.6 OneSAF Performance Monitoring

The US Army Research Laboratory Simulation Training Technology Center (STTC) in partnership with the US Army Program Executive Office for Simulation, Training and Instrumentation (PEO STRI) and the USASMDC/ARSTRAT supported a research effort to explore the performance characteristics of OneSAF in massively-parallel, distributed HPC environments to assess FMS support for training as a service. There were two primary technical components to this effort. The first component was to enhance OneSAF's original benchmarking performance tool to collect more relevant performance parameters from OneSAF. The second component was to make modifications to OneSAF to allow key performance parameters to be accessible to the benchmarking tool (BMT). The result was the development of the Remote Distributed Performance Monitor. This tool provides an enhanced capability to the OneSAF community to measure simulation performance, while providing the DoD community with access to a well-established tool for training and analysis activities.

The OneSAF Remote Distributed Performance Monitor (RDPM) is a lightweight tool designed to help users, developers, and co-developers monitor and analyze performance of multiple instances of OneSAF simulation engines running on a local network. The tool interacts with OneSAF remotely via JMX, and is explicitly designed for very minimal interference with simulation operation. It can be run alongside any OneSAF simulation in almost any configuration. It can be configured to track a wide variety of information regarding the OneSAF simulation engine, and can report such information directly to a file log for post-processing and analysis, or displayed online as the simulations are executing.

2.7 OneSAF V5.1 Features Developed Specifically for HPC Use

A number of recent OneSAF improvements have been introduced into the baseline to make the application more HPC-friendly. The V5.1 improvements include:

- The ability to run multiple instances of OneSAF from a single installation. Previous HPC installations of OneSAF required the software to be separately installed for each node on which it was executing. Although not a big deal for networked workstations, this is very impractical for HPCs. This new feature allowed OneSAF to be installed once and then instanced on multiple nodes.

- The ability to execute multiple OneSAF applications on a single node. When OneSAF reduces the simscale to address network loading, it is able to reset the scale higher once network loading is reduced.
- OneSAF running in as-fast-as-possible mode when running in a distributed cluster.[2]

3. Current Efforts

As part of an effort to expand OneSAF use in the community, the Army Modeling and Simulation Office (AMSO) has partnered with the US Army PEO STRI to extend OneSAF's HPC capability while exploring the delivery of OneSAF simulation services through a "cloud"-like delivery model.

Two projects have been initiated to address HPC application of OneSAF:

- Improvement of Simulation Software to More Effectively Utilize High Performance Computing – Parallelizing OneSAF
- Simulation Software-as-a-Service for Forces Modeling and Simulation

3.1 Parallelizing OneSAF Functionality

Although OneSAF has been successfully implemented on several HPC systems, the software has not been optimized to take full-advantage of HPC resources. There are a number of capabilities that can help OneSAF run more efficiently within the HPC environment. Several of these are currently under development. These include:

- Improved Communications - OneSAF was developed to communicate within the constraints of a TCP/IP Ethernet network. OneSAF is being updated to leverage the faster network communications mechanisms (Infiniband) present in HPCs by removing the tcp wrapper from OneSAF messages.
- Off-load Computationally-Intensive Functions to Other Processors – Calculations like Line-of-Sight (LOS) and earth skin state are computationally-intensive, and may be more efficiently executed by running them on separate CPUs or on GPUs. Several options are under consideration for "off-load" to determine if improvements can be made for improving computational speed and efficiency. This would also introduce the possibility in the future for other models to provide these same functions to the OneSAF model.
- Improve OneSAF Batch-Run Capability – Analysis use of OneSAF in batch mode is a classic example of an embarrassingly parallel problem; being able to execute many replications of OneSAF model runs at the same time, thus providing a shorter time-to-solution. OneSAF's replication mechanism is being updated to farm-out replication iterations to multiple OneSAF clients in the HPC, parallelizing what is now a sequential task.

The project that is implementing these capabilities is just getting underway and should have some results later this year.

3.2 OneSAF Simulation-as-a-Service

The Army is exploring the concept of Simulation as a Service to more flexibly deliver OneSAF simulation capabilities to a broader user audience by reducing the cost of entry, while providing high-quality and high-performance simulation support. Delivery of Simulation-as-a-Service using HPC systems provides the ability to expand and contract resources needed for simulation execution as needed by the specific user. It also provides the software developer the ability to rapidly deploy new software versions and capabilities to users who will no longer need to wait for media to be created, and local installation to take place.

This capability will allow the Army to explore simulation deployment options for the future that will conserve resources and make efficient use of shared resources. Simulation service issues will be addressed through the following activities:

- Development of a web-based "portal" in conjunction with the development of the Maui Portal development project for accessing the OneSAF simulation – providing an FMS option on the portal.
- Allow users, through the portal, to select simulation mode (real-time, faster than real-time or batch - analysis) along with preloaded scenarios and configurations for execution on a selected HPC system.
- Develop a methodology for remote visualization of OneSAF Controller and Battlemaster views that support human-in-the-loop interaction to support exercises run on the HPC.
- Develop the scripts to automatically configure, initialize and kickoff OneSAF scenarios selected through the portal based, on user input through the portal.
- Provide a user's manual for using OneSAF through the HPC portal.

Preliminary progress on this project includes the development of a web-based front-end that allows the user to configure the OneSAF execution, as well as load-in scenarios from a local repository. Several options for remote visualization are also under review for implementation.

5. Summary/Conclusion

HPC provides a key capability to allow the DoD community to develop critical technology concepts quickly and in a more robust, high-fidelity environment. OneSAF provides an operationally-relevant context to support training, test and evaluation, and analysis activities. Together they provide a robust synthetic environment.

As the use of OneSAF on HPC grows, HPC improvements continue to be made to OneSAF, allowing OneSAF's performance capabilities to grow as computational hardware advances toward more parallel, cluster-type systems. It also provides DoD engineers and researchers a valuable tool for scaling the simulation environment to meet a more demanding, higher-fidelity application of OneSAF.

Remaining challenges exist in determining true simulation performance, exploring distributed configurations, and interoperability performance in new environments – DIS vs. HLA.

6. Future Efforts

As the use of OneSAF on HPC's grows and the capability of OneSAF to use HPC resources improves, there is great potential to utilize the DoD supercomputing resource centers to offer OneSAF simulation-as-a-service to DoD users. Future efforts will seek to continue to extend cloud access and parallelizing efforts to develop a more robust simulation environment.

Acknowledgements

The work of the PETTT program is sponsored by the High Performance Computing Modernization Program Office under contract to High Performance Technologies, Inc./DRC.

References

1. DoD High Performance Computing Modernization Program website, http://www.hpcmo.hpc.mil/cms2/index.php, January 18, 2012.

2. *OneSAF Users Manual, Version 5.0*, Document Control No. OneSAF A&I-06000173-092410-8.0, September 24, 2010.

3. Bouwens, C. and D. Pratt, "The Application of High Performance Computing for Forces Modeling and Simulation." Fall 2010 Simulation Interoperability Workshop, September 2010.

4. *OneSAF HPC Session*, Fall 2010 Simulation Interoperability Workshop, September 2010.

5. Wittman, R., D. Franceschini, and J. Surdu, "OneSAF Objective System: A Simulation Toolbox." *Proceedings of the Spring Simulation Multiconference*, San Diego, CA, pp. 201–206, April 2005

Progress Report on the Microsoft HPC Portal

Rob Fisher
BAE Systems, Vienna, VA
jrfisher@hpcmo.hpc.mil

Pat Collins
US Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
james.p.collins106.civ@mail.mil

Jim Waterman
Lockheed Martin, Aberdeen Proving Ground, MD
james.l.waterman.ctr@mail.mil

Robert Ames and Ian Shields
US Naval Surface Warfare Center, Carderock Division (NSWCCD), West Bethesda, MD
{robert.ames, ian.shields}@navy.mil

Manuel Vindiola
Dynamics Research Corporation, Aberdeen Proving Ground, MD
manuel.m.vindiola.ctr@mail.mil

Abstract

The Engineering Tools Portal (ETP) is a prototype web portal that delivers Windows high performance computing (HPC) applications to users' desktops. It is built on Microsoft's HPC Server 2008 R2 and RemoteApp technologies. The ETP web site displays application icons based on the user's permissions and, when clicked, the application runs on the portal and is displayed on, and integrated with, the user's desktop. HPC jobs are submitted to the back-end cluster through the remote application's interface.

ETP became fully operational as a prototype at the Army Research Laboratory's Distributed Shared Resource Center in January, 2012. This paper focuses on the progress since then and the user experiences with this system. To date, MATLAB and the Computational Research and Engineering Acquisition Tools and Environments (CREATE) Ships Program applications have been the primary focus. Progress on these applications is discussed in some detail.

1. Introduction

The Engineering Tools Portal started as an US Army Research Laboratory (ARL) project to evaluate Microsoft's HPC Server (Microsoft, 2010) around the same time the High Performance Computing Modernization Program (HPCMP) was getting interested in web portals. ARL's involvement in both efforts naturally led to a merger of the two projects. An access method was needed for the high performance computing (HPC) server and portal access looked promising. With help from Microsoft, architecture was developed that used many of their proven, off-the-shelf technologies for building web access to remote applications. The design point was a system that delivers HPC capable applications to the user's desktop with WAN access restricted to the https protocol.

The architecture that was developed is shown in Figure 1. It has two main components: the Portal that provides the access mechanism and the Microsoft HPC Server that provides the computing resources. Entry into the system is through the SharePoint web server. The Portal home page presents a set of application icons to users based on their access rights. When an icon is clicked, an RDP over HTTPS connection is established between the user's desktop and the Remote Desktop (RD) Gateway. This connection goes through the Microsoft Forefront Threat Management Gateway (TMG) firewall where Common Access Card (CAC) authentication is enforced. The embedded RDP connection is then completed with one of the RD Session Host servers. The application is started on the Session Host server and its interface is presented to, and integrated with, the user's desktop via Microsoft's RemoteApp (Microsoft, 2010) technology. Because portal

applications are integrated with the HPC Server job scheduler, users have access to HPC resources from the familiar application interface. Engineering Tools Portal (ETP) was designed and built around this architecture.

Other portal components that support this connection mechanism include: the RD Broker that provides load balancing of the Session Host servers, an RD License server, Active Directory, DNS, and SQL servers.

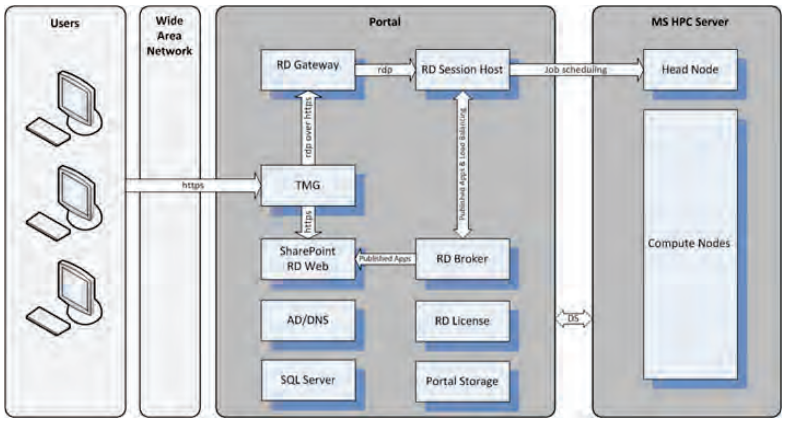


Figure 1. ETP architecture

MATLAB was chosen as the canonical application for evaluating ETP because of its widespread use within the Department of Defense (DoD), and because it is capable of utilizing HPC resources. According to MathWorks, there are more than 13,000 MATLAB users in the DoD. Many of these users could take advantage of the HPC computing resources.

There are, however, many other computationally-intensive Windows applications used in DoD that could also benefit from such a system. These include the suite of applications being developed by CREATE-Ships program and numerous forces modeling and simulation (FMS) codes. During the prototype phase, the goal is to evaluate as many applications as possible to assess the feasibility of building an expanded system in a production environment. Since the system became operational, the focus has been on MATLAB and the Computational Research and Engineering Acquisition Tools and Environments (CREATE) Ships applications.

Preliminary results with ETP have been very encouraging. A number of MATLAB users from across all branches of the Services are using the system and are very pleased with it. The CREATE-Ships Program is seriously considering expanding the system to provide their user base with a production quality system.

This paper is a progress report on the system since it became operational. This is done in two parts. Section 2 presents the system status. The user base will be described and system utilization statistics will be presented. Section 3 provides some details and experiences with MATLAB and the CREATE-Ships applications. For more detailed information about the ETP system design, please refer to (Collins, et al., 2011).

2. Status

ETP received an Interim Authority to Operate (IATO) at the ARL DoD Supercomputing Resource Center (DSRC) on 9/30/2011. The system was made available to a small group of select users for shakedown testing for the three months following the IATO. During that period some issues were discovered and resolved. Also during that period, a collaboration began with the CREATE Ships program. Installation and testing of the CREATE-Ships codes began in December. At the end of December, it was determined that ETP was stable and ready for pioneer MATLAB users. ETP was advertised to the MATLAB community in January.

Figure 2 shows the growth in the ETP user base since the IATO and Figure 3 shows the distribution of these users across the services and applications. The slope of the growth of MATLAB users curve clearly shows where advertising began. The difference between total users and MATLAB users is mostly the CREATE-Ships development team; with the exceptions being a few test users, a few technology evaluators and one Forces Modeling and Simulation user.

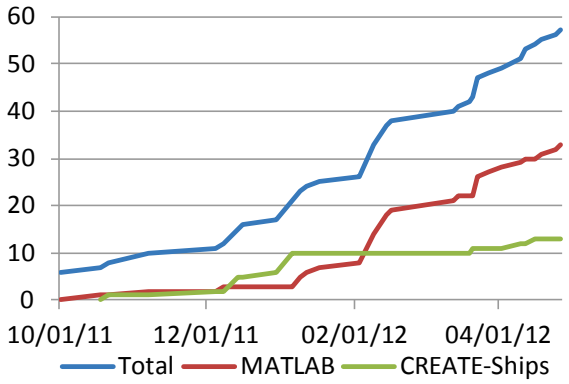


Figure 2. ETP user growth

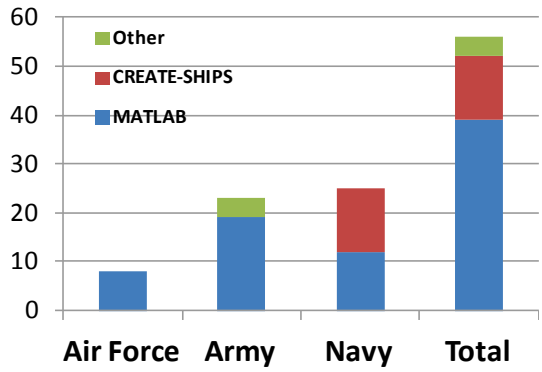


Figure 3. ETP user distribution

Because there is a learning curve associated with using the HPC server, utilization of it started off low, but has steadily increased. Figure 4 shows utilization for April. The two curves show percent utilization based on "All cores" and on "Available cores". The "All cores" percentage is based on the system maximum of 264 cores. The "Available cores" takes into account nodes that might be down or offline for some reason. Because the CREATE-Ships applications are not yet tied to the HPC scheduler, any nodes utilized by their codes must first be taken offline. For the past 4 months, the standard configuration has been 4 nodes (48 cores) dedicated to CREATE-Ships development with the 216 cores available for MATLAB.

The graph shows sporadic use, but the peaks are getting high. For this small system, a small number of MATLAB users can easily fill up the system.

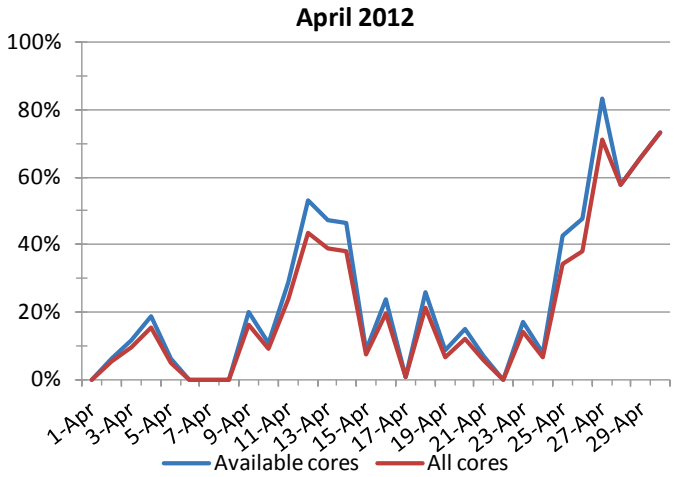


Figure 4. HPC server utilization for April 2012

A new addition to ETP is the ability to check system status and HPC node utilization. At the bottom of the ETP home page is a matrix showing the status of systems and services. It also shows which nodes are dedicated to CREATE-Ships (IHDE Compute Nodes) and which are available for MATLAB (HPC Compute Nodes), and their status. At the very bottom is a heat map showing HPC server CPU utilization.

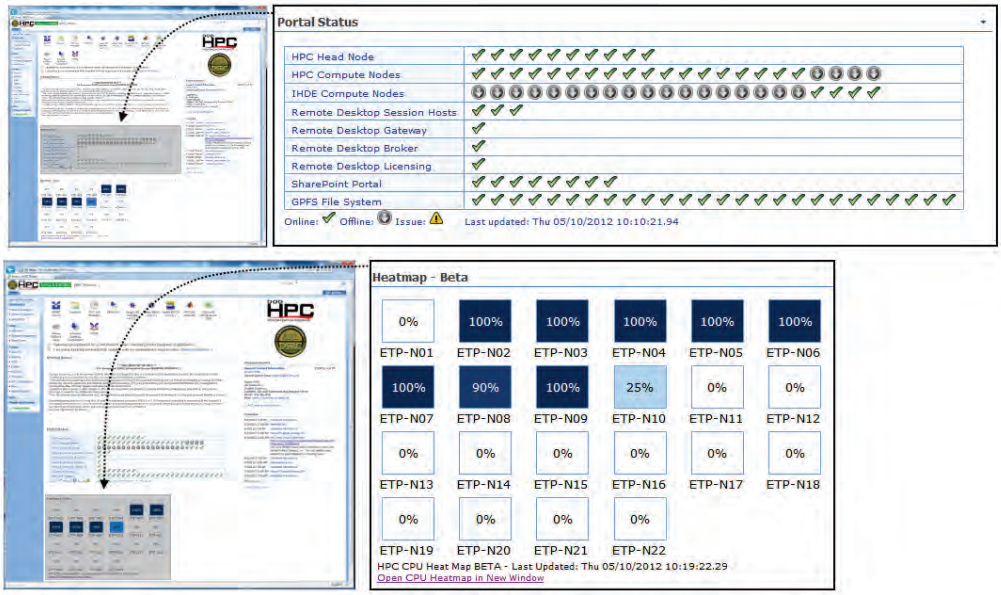


Figure 5. Portal status and HPC utilization

3. Applications

MATLAB and the CREATE-Ships applications have been the focus of ETP since it became operational. The next two sections discuss these applications relative to ETP.

3.1 MATLAB

To take advantage of the HPC server, MATLAB users must learn and use the Parallel Computing Toolbox (PCT) (MathWorks, Inc, 2011). Although it is relatively easy for a seasoned MATLAB user to become proficient at this, there is still a learning curve associated with it. The ETP team has been working with some of the users to get their codes running on the HPC server. A simple example of how to run a parametric study using PCT is included in the ETP user guide (ETP Development Team, 2012).

A survey of the users was taken in April. One of the questions asked was if they used the Parallel Computing Toolbox. Out of the 19 respondents, 17 responded that they have actually used ETP and 9 of those reported that they did use PCT. The goal is to get all users running their computationally-intensive calculations on the HPC server. The ETP team continues to work with the users to make this happen.

As of this writing there are 33 MATLAB users using the system. ETP has sufficient licenses to support 16 of them concurrently, and sufficient MDCS licenses to utilize all compute cores. They develop and run a variety of codes in support of DoD projects. A sampling of the technical areas that ETP users support is:

- Counter-Improvised Explosive Device,
- Testing receivers' ability to acquire and demodulate signals,
- Ocean Survey Program,
- Aircraft performance data modeling,
- Neural Mass Modeling,
- Flapping-Wing System for Micro Air Vehicle,
- Climate change vulnerability assessment modeling,
- Modeling of Brillouin scattering in optical fiber amplifiers.

The Neural Mass Modeling project will be used to illustrate the benefits of the Engineering Tools Portal to MATLAB users. This work is currently being done by co-author Manuel Vindiola. Mr. Vindiola is one of the first ETP users and was able to quickly take advantage of the HPC server.

The neural mass modeling MATLAB code uses a dynamical system to generate simulated brain imaging signals. The simulations examine how underlying brain network topology influences the Electroencephalogram (EEG) and local field potential (LFP) signals generated by the dynamical system. From these simulated signals, statistical measures for identifying time-locked amplitude or phase correlations between signals are computed to reconstruct the effective topologies of the network. Figure 6 shows the functional connectivity of a particular topology from a simulation done on ETP. Both the results and figures were computed on ETP. This code takes approximately 25 minutes to simulate 11,000 time-points for a total of 11 seconds worth of simulated data. The statistical measures require repeated measurements, and ETP can run many independent trials simultaneously using the PCT distributed computing functions. Results from this work aim to contribute to improvements in Soldier-specific training and tasking, and the development of brain-computer interaction technologies.

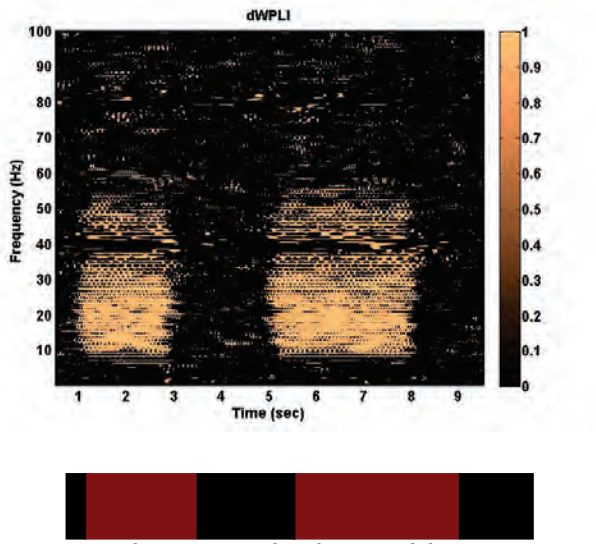


Figure 6. Functional connectivity

A snippet of the PCT code that runs the neural mass modeling code on the cluster is given below.

```
sched = findResource();
job = createJob(sched,'Name', ['NMM Simulation #' num2str(version)])
p = pathlist(genpath('\\ETP-H1\GroupShares$\MATLAB\Manuel M Vindiola\NMM'));
set(job, 'PathDependencies', p);
for itr = 1:iterations
        createTask(job,@spm_int_ode, 1, {M(itr).pE,M(itr),M(itr).pE.U});
end
submit(job);
disp(['Submitted Job ' job.name]);
wait(job);
jobresults = getAllOutputArguments(job);
```

The PCT constructs used here are: findResource(), createJob(), createTask(), submit(), wait() and getAllOutputArguments(). The findResource() function grabs the HPC scheduler. The system has been configured so that this always returns the ETP Windows HPC server scheduler. A single job is created with createJob(), and many independent tasks are added with createTask(). The submit(job) command sends the job to the head node where it is scheduled to run. The command getAllOutputArguments(job) gets the results. This works because a wait(job) command has been issued prior to it. Waiting is not necessary and getAllOutputArguments() could be run at a later time after the HPC job has completed.

Mr. Vindiola has not done a formal comparison between running jobs on the desktop and running on the cluster. He has reported; however, a significant increase in productivity and throughput. Jobs that would have taken at least 20 hours on his desktop can now be run in 30 minutes on the ETP, and now many more can be run.

The particular advantages of ETP that Mr. Vindiola enjoys are:

- switching between offices and having ETP available in both places,
- the flexibility of checking on jobs or starting jobs from home,
- the MATLAB GUI is just as quick and responsive when at work or at home.

The biggest concern with ETP that Mr. Vindiola expresses is the system size. As few as 3 users can, and have, filled up the system. This hasn't happened much, but as HPC server utilization picks up, job turnaround-times will increase.

3.2 CREATE-Ships

The CREATE program's mission is to bring HPC resources to DoD acquisition programs. CREATE-Ships is the part of that program which focuses on HPC tools for ship acquisition. Most Navy acquisition program engineers and analysts use Windows computers and applications, and consequently many of the tools being developed, including some HPC-capable tools, are being written for the Windows platform. Windows applications and HPC resources don't typically go hand-in-hand within the traditional HPC user community; however, ETP has the potential to change that. It provides a unique opportunity to bring HPC resources to a community of users that typically do not use HPC computers, and it does so in a way that is familiar and easy. Because of this, the ARL ETP team started collaborating with the CREATE-Ships team in December, 2011.

CREATE-Ships have two projects targeting Windows-based customers: Hydrodynamics and Rapid Design Integration. In support of this, the following software has been installed on ETP and is currently being tested:

- Integrated Hydrodynamics Design Environment (IHDE),
- Rapid Ship Design Environment (RSDE),
- Parent Hull-form ,
- Leading-Edge Architecture for Prototyping Systems (LEAPS) Database and Utilities,
- LEAPS SHCP,
- Advanced Surface Ship Evaluation Tool (ASSET).

Figure 7 shows the ETP home page for the CREATE-Ships developers. It contains icons for all the software listed above, plus supporting software such as Excel, an ETP Explorer window, and the HPC job manager. Figure 8 shows the IHDE graphical user interface (GUI) as it would appear on a user desktop. The GUI shows a particular hull-form that was loaded from the LEAPS database in preparation for a hull-resistance analysis.

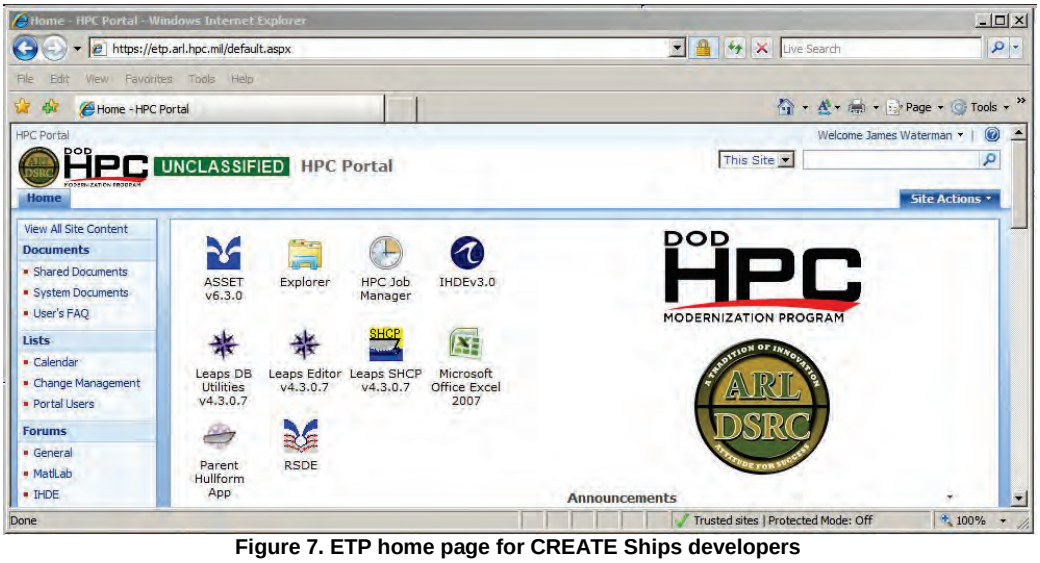


Figure 7. ETP home page for CREATE Ships developers

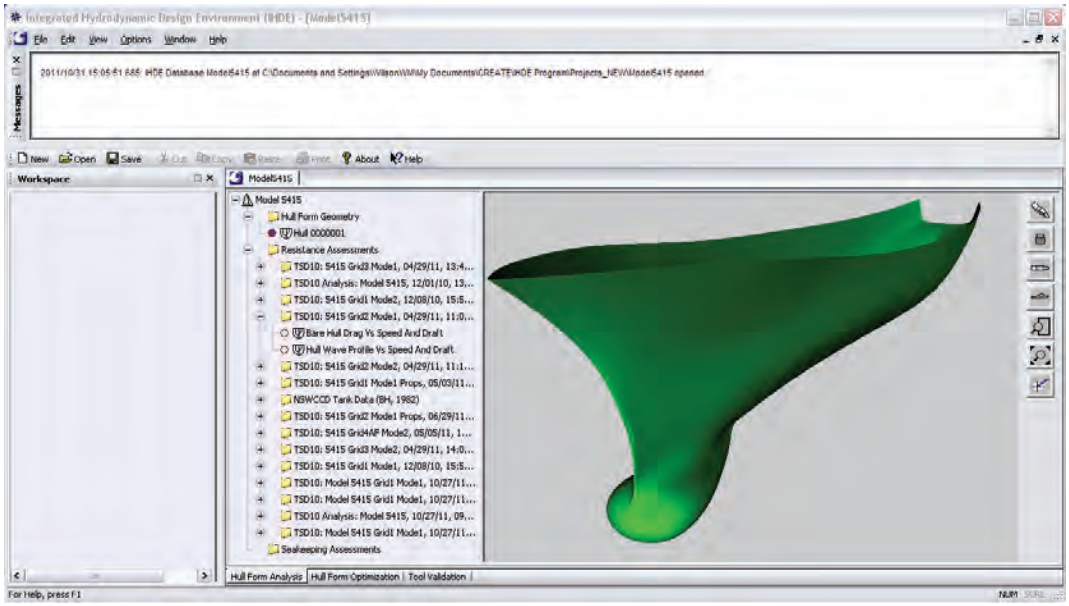


Figure 8. The IHDE GUI showing a hull-form from the LEAPS database that will be used in a resistance analysis

After a few months of testing, the CREATE-Ships development team has found ETP to be very promising and is interested in moving towards a production system. The advantages they found and would like to duplicate on a production system are:

- Easy and familiar access:
  Ship acquisition programs typically take place at "Design Sites". These can be either government or industry locations; typically contractors "just outside the gate". Scientists and engineers supporting these design sites can be located at the Navy Yard, labs such as Carderock, Dahlgren, NRL, and other contractor sites or shipyards. ETP solves this problem by providing easy and familiar access from almost anywhere.
- Solves many security issues:
  IT Security and the Navy Marine Corps Intranet (NMCI) significantly hamper what can be achieved at customer's site, even with Windows-based applications. Using ETP does not require any software to be installed on the desktop. This overcomes the need to get IT security's approval to install software. Also, the ETP prototype has met all the security requirements necessary to operate in the ARL DSRC environment. Any subsequent production system would do the same.
- It provides an interactive environment with almost the same experience as running on the local desktop.

There is still much work to be done. The CREATE-Ships Remote Execution System (RES) shows promise, but thorough testing has not yet been completed. Integration with the HPC scheduler is planned, but has not yet been done. There is also concern about long-term Windows support within the HPCMP program, scalability and Remote Execution System interoperability with and between other DSRC machines.

4. Summary and Future Direction

The CREATE-Ships Program has had very good success with running their applications on ETP. These successes are sufficient to consider moving from a prototype to a production system with capabilities to meet the computational requirements. ARL, the HPCMP, and the CREATE-Ships Team are looking into ways to make this happen.

MATLAB has been a big success on ETP. Users that take advantage of the HPC server on a regular basis have significantly increased their productivity. Easy and familiar access to the system has allowed them to work from almost anywhere. An important point is, to take advantage of the HPC server the user interacts only with the MATLAB interface. With just a few PCT constructs, a user can take advantage of the HPC resources.

The ETP team would like to continue supporting MATLAB on ETP in FY13. License costs, however, prohibit scaling the system to support a larger number of MATLAB users at this time.

Acknowledgements

The authors would like to thank Tom Kendall (ARL) and Dave Morton (MHPCC) for their continued support for the project; Andy Greenwell, John Kreatsoulas, Chris Crall, Tom Quinn, and James Baker from Microsoft; and the HPCMP for their financial support. A special thanks goes to Dr. Aram Kevorkian (SPAWAR) for his help in advertising ETP to the MATLAB community.

References

Collins, P., Kendall, T., Waterman, J., Knowles, M., Fisher, R., and Greenwell, A., et al., "A Microsoft HPC Portal." *HPCMP User Group Conference 2011*, Portland, Oregon, IEEE Xplorer, 2011.

ETP Development Team,. *Engineering Tools Portal (ETP) User Guide*. Retrieved May 2012, 2012, US Army Research Laboratory, http://www.arl.hpc.mil/special/ETPUserGuide.html, Jan 3, 2012.

MathWorks, Inc., *Parallel Computing Toolbox*, Retrieved May 13, 2011, http://www.mathworks.com/products/parallel-computing/, 2011.

Microsoft, *Overview of RemoteApp*, Retrieved May 11, 2012, Overview of RemoteApp: http://technet.microsoft.com/en-us/library/cc755055.aspx, 2012.

Microsoft, *Windows HPC Server 2008 R2*, Retrieved May 11, 2012, Windows HPC Server 2008 R2 Suite: http://www.microsoft.com/hpc/en/us/default.aspx, 2010.

Author Index

Adamec, Stephen 541
Addison, Jason 268
Allen, Jeffrey B. 113
Almalkie, S. 147
Ames, Robert 605
Aram, Shawn 98
Arbic, Brian K. 467
Au, Henry 438
Barnes, Caleb J. 52
Batyrev, Iskander 107
Bellaiche, L. 201
Bernholc, J. 160
Bhamidipati, Keerti K. 10
Bique, Stephen 409
Boles, John A. 20
Borelli, Kathy 268
Boris, J.P. 428
Bouwens, Christina 595
Brès, Guillaume A. 168
Brinckman, K. 330
Brown, Michael S. 249
Brucker, Kyle A. 455
Burt, Jonathan 315
Byrd, Edward F.C. 354
Cabello, Alex 308
Cain, Alan B. 484
Calahan, Josh 308
Caldwell, Steve 308
Cannon, Randal 551
Carlen, Frank 551
Chang III, Peter A. 563
Chaves, Juan Carlos 387, 507
Chen, Chiung-Chu 122
Chen, Sue 444
Choi, Young 581
Cialone, Mary A. 325
Clarke, Jerry 382
Collins, Pat 605
Cornwell, Charles F. 113
Cummings, Russell M. 238
Dash, S.M. 330
Datta, Anubhav 342
Davis, Roger R. 525
de Bruyn Kops, S.M. 147
Decker, Robert 64
DeSalvo, William 586
DeSpirito, James 390
Devine, Bryce D. 113
Dinavahi, Surya P.G. 98
Dogaru, Traian 363
Dommermuth, Douglas G. 455
Doyle, James 444
Du, Yongle 484
Duncan, R. Bruce 268
Ebrahimi, Houshang 276
Elton, Bracy 507
Erwin, James P. 180
Farnsworth, Brad 268
Fasel, Hermann F. 40, 73, 192
Feldman, G. 330
Fisher, Rob 605
Flowers, Gloria 595
Freericks, J.K. 219
Gaitonde, Datta V. 249, 276
Ghoreyshi, Mehdi 238
Gordnier, Raymond E. 52
Gottiparthi, Kalyana C. 209
Grant, Luke 87
Grinberg, Ilya 137
Gross, A. 73, 192
Guézennec, Nicolas 257
Gutmark, E. 428
Hagenmaier, M.A. 20
Ham, Frank E. 168
Harrington, Aaron 98
Heavey, Karen R. 390
Heeb, N. 428
Hench, David L. 31
Henninger, Amy 595
Henz, Brian J. 375, 401, 537
Hirata, Jason 308
Hodak, M. 160
Hodur, Richard 444
Hosseinverdi, S. 73
Howard, Rebecca M. 290
Ianni, James C. 586
Jacobi, R. 192
Jayaraman, Buvana 342
Jiang, Lai 137
Jin, Hao 444

Author Index

Jin, Jian-Ming 497
Jin, Yi 444
Josyula, Eswar 315
Kailasanath, K. 428
Kendall, Thomas 586
Kennedy, K.D. 330
Kenzakowski, D.C. 330
Knowles, Michael P. 586
Kolpak, Alexie M. 137
Krishnamurthy, H.R. 219
Kroninger, Christopher 98
Ku, Daniel 581
Lariviere, Brian 257
Lê, Hung 168
Lechniak, Jason A. 10
Leiter, Kenneth W. 282
Lele, Sanjiva K. 168
Levchenko, Sergey V. 157
Liao, DaHan 363
Lill, James 5
Linford, John C. 541
Liou, Chi-Sann 444
List, Michael 290, 299
Liu, J. 428
Liu, Yuming 476
Lu, W. 160
Lum, Gregory 438
Lynch, Edward D. 257
Macon, Charles 497
Mahesh, Krishan 563
Mallo, Adam 268
Malo-Molina, Faure J. 276
Malony, Allen D. 541
Masquelet, Matthieu 257
Matthews, Phillip G. 586
Mattson, Michael D. 563
McGhee, Sean M. 387
McLay, Justin 444
McQuaid, Michael J. 122
Meng, Huan-Ting 497
Menon, Suresh 209, 257
Metzger, E. Joseph 467
Mikelsons, K. 219
Monday, D. 428
Morris, Philip J. 484
Morton, David 512
Moskaitis, Jon 444
Mullenix, Nathan J. 249
Munipalli, Ramakanth 257
Nardelli, M. Buongirno 160
Nelson, Chris C. 484
Nichols, Joseph W. 168
Nie, Baolin 497
Nikhil, Rishiyur 409
Nusca, Michael J. 122
O'Shea, Thomas T. 455
Park, Song J. 375, 401
Paschal, Alesya 595
Pasiliao, Crystal L. 10
Polanco, Miguel Angel Méndez 137
Polsky, Susan A. 420
Porter, Christopher 64
Post, Martiqua L. 238
Potsdam, Mark 342
Raeth, Peter G. 387, 507
Ramseyer, George 268
Ranawake, Udaya 551
Ranjan, V. 160
Rappe, Andrew M. 137
Renard, Kenneth 382, 537
Reynolds, Carolyn 444
Richie, David A. 375, 401
Richman, James G. 467
Ridout, James 444
Risha, Daniel 276
Rodebaugh, Gregory P. 180
Rokos, Kyle 551
Ross, James A. 375, 401
Ross, Virginia W. 31
Ruloff, Jordan J. 375, 401
Sabol, Chris 268
Sahu, Jubaraj 390
Sanders, Darius 299
Sauerwald, Chris 586
Seidel, Jürgen 64
Shan, Hua 563
Shende, Sameer 541
Shields, Ian 605
Shires, Dale R. 375, 401
Shriver, Jay F. 467

Author Index

Singh, Rajneesh 98
Sinha, Neeraj 180
Sivasubramanian, Jayahar 40
Slomski, Joseph 563
Smedstad, Ole Martin 467
Smith, Kathleen A. 586
Smith, Rodney 581
Spetka, Scott 268
Spyropoulos, John 168
Stone, Christopher P. 525
Strang, William Z. 231
Sullivan, Anders 363
Talcott, Denise 551
Talley, Douglas G. 257
Tan, B. 160
Taylor, DeCarlos E. 107
Titov, Evgeny 315
Tomaro, Robert F. 231
Tuckey, Todd 308
Unpingco , Jose 507
Viloria, Ron 268
Vindiola, Manuel 605
Visbal, Miguel R. 52
Wallcraft, Alan J. 567
Walters, Richard I. 325
Waterman, Jim 605
Weingarten, Neil S. 354
Welch, Charles R. 113
Wissink, Andrew M. 541
Wong, Steve 497
Woodward, Christopher 5
Wurtzler, Kenneth E. 231
Xiao, Wenting 476
Yang, Yurong 201
Yentsch, Robert 249, 276
Yue, Dick K.P. 476
Zambon, A.C. 330

High Performance Computing Modernization Program

www.hpc.mil