Parallelization, Acceleration, and Advancement of Dissipative Particle Dynamics (DPD) Methods

Timothy I. Mattox*1, James P. Larentzos1, Christopher Stone2, Sean Ziegler1, John K. Brennan3 and Martin Lisal4

1 Engility Corporation at the U.S. Army Research Laboratory, Aberdeen Proving Ground, MD 21005, U.S.A.
2 Computational Science & Engineering, LLC, Chicago, IL 60622, U.S.A.
3 U.S. Army Research Laboratory, The Rodman Materials Research Laboratory, Aberdeen Proving Ground, MD 21005, U.S.A.
4 Laboratory of Aerosols Chemistry and Physics, Institute of Chemical Process Fundamentals of the ASCR and Department of Physics, J.E. Purkyně University, Ústí nad Labem, Czech Republic

Coarse-Graining of RDX
- Bottom-up coarse-graining (CG) from atomistic models is used to develop the meso-scale model.
- Properties which depend on intermolecular interactions, such as the phase diagram, are well reproduced. However, properties which depend on the coarse-grained intramolecular degrees of freedom, such as the heat capacity, are underestimated.
- DPD-E provides a means to recapture these lost degrees of freedom.

Shardlow Splitting Algorithm (SSA)
- Split momenta integration into deterministic dynamics and stochastic dynamics.
- Both can then be integrated using standard numerical integrators.
- Readily extended to other DPD variants (not true for other integrators).
- Larger time steps allowable (~10-100x larger, with comparable energy conservation).

FE Simulation

Parallelization of SSA in LAMMPS
- Domain Decomposition
- Staged Directional Communication

Chemical reaction equations within each DPD mesoreactor can be numerically stiff and costly to solve. The kinetics of each mesoreactor are independent of other particles and can be solved in parallel: an ideal candidate for many-core computing. Thus we accelerated the ODE solver tasks for the mesoreactor on a many-core processor using OpenMP or CUDA C.

While the performance gains for complex reaction scenarios shown above are promising, the SSA consists the vast majority of the runtime for non-reactive DPD scenarios (and for some simpler reactive scenarios). Thus, we are also improving the parallel efficiency of the SSA by restructuring/rewriting the MPI communications and the neighbor list construction for DPD. In addition, we are now using the Kokkos1 C++ library (recently included in LAMMPS) to parallelize the intra-node performance of the SSA as well as the reaction DPD kernels for a diverse set of modern architectures, from GPUs to multi- and many-core CPUs.

* Corresponding author: timothy.mattox@engilitycorp.com

Acknowledgements

This effort is supported by the Institute for Multiscale Reactive Modeling of Invasive Munitions (ISMIM), which is a multi-team effort led by the U.S. Army Research Laboratory and the U.S. Army Armament Research, Development, and Engineering Center. The authors also would like to acknowledge the computational resources and PETT software support from the DoD High Performance Computing Modernization office under Contract No. GS040708C00107.

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

Gain Computational Speed by Removing Atomistic Detail

Constant Energy Dissipative Particle Dynamics (DPD-E)
- DPD-E provides a means to recapture the lost degrees of freedom due to coarse-graining by adding additional forces to account for motions and momentum dissipation. A meso-particle equation of state is coupled to the dynamics which accounts for inner degrees of freedom (i.e., chemical bonding) which were lost during coarse-graining. CG-MD gives poor Hugoniot properties since it is missing the meso-EOS.

Hugoniot Properties

LAMMPS DPD Reaction Kinetics

8000 particles of RDX, 201 reactions, 45 species

While the performance gains for complex reaction scenarios shown above are promising, the SSA consists the vast majority of the runtime for non-reactive DPD scenarios (and for some simpler reactive scenarios). Thus, we are also improving the parallel efficiency of the SSA by restructuring/rewriting the MPI communications and the neighbor list construction for DPD. In addition, we are now using the Kokkos1 C++ library (recently included in LAMMPS) to parallelize the intra-node performance of the SSA as well as the reaction DPD kernels for a diverse set of modern architectures, from GPUs to multi- and many-core CPUs.