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

Timothy I. Mattox, James P. Larentzos, Sean Ziegeler

Engility Corporation
APG, MD 21005, U.S.A.
+1 410-278-9767
timothy.mattox@engilitycorp.com
james.larentzos@engilitycorp.com
sean.ziegeler@engilitycorp.com

Christopher P. Stone

Computational Science and Engineering, LLC
Chicago, IL 60622, U.S.A.
+1 773-796-4108
chris.stone@computational-science.com

John K. Brennan

U.S. Army Research Laboratory
The Rodman Materials Research Laboratory
Aberdeen Proving Ground, MD 21015, U.S.A.
+1 410-306-0678
john.k.brennan.civ@mail.mil

Martin Lísal

Laboratory of Aerosols Chemistry and Physics
Institute of Chemical Process Fundamentals of the ASCR and
Department of Physics
J.E. Purkinje University
Ústí nad Labem, Czech Republic
+420 220 390 301
lisal@icpf.cas.cz

Keywords—Performance acceleration, DPD Method

I. Extended Abstract

In recent years, our group has developed mesoscale modeling capabilities necessary to represent salient physical and chemical features of material microstructure. We built novel simulation tools that incorporate coarse-grain (CG) models upscaled from quantum-based models, which are then coupled with continuum level approaches. We further developed a suite of discrete-particle modeling tools, based upon the Dissipative Particle Dynamics (DPD) method [1][2], for simulation of materials at isothermal, isobaric, isoenergetic, and isoenthalpic conditions. While DPD was already available in numerous scalable commercial and open-source software packages, it has been limited to simulations at isothermal conditions, but more significantly, these codes used integration schemes that require time steps too small for simulations of DoD-relevant materials such as RDX. To broaden the applicability and usability of our DPD methods, we have recently integrated them into LAMMPS [3], one of the premier scalable and open-source molecular dynamics codes for computational chemistry and materials science applications. We call this combined software package DPD-LAMMPS which includes our novel suite of DPD variants. Our poster presents the parallelization, acceleration, and advancement of DPD methods for DPD-LAMMPS, as briefly described below.

In addition to the ability to simulate at various conditions, DPD-LAMMPS has a particularly unique capability to model chemical reactivity within the particles. Such simulations routinely consist of $O(10^{7-8})$ CG particles that act as mesoreactors. Each mesoreactor has a corresponding set of reaction kinetic equations that must be repeatedly solved efficiently and accurately over millions of DPD timesteps. Considering a multi-step reaction mechanism, the resulting chemical kinetics equations are expressed as a system of ordinary differential equations (ODEs), which can be solved by application of numerical integration schemes to update the composition of the CG particle as a function of time. The reaction DPD kernels include interfaces to an internal 4th order Runge-Kutta solver as well as external ODE solver libraries, such as the GNU Scientific Library (GSL) [4] and CVODE [5]. Each particle solves its own set of reaction kinetic equations, which can be performed independently of the other particles and provides an opportunity for parallelization and acceleration.

Another critical accomplishment from our efforts was the design and implementation of the first-ever parallel version of the Shardlow-splitting algorithm (SSA), which overcomes the time step restrictions of other integration techniques [6]. While the DoD has greatly benefited from these DPD developments, our current work focuses on performance enhancement to more efficiently utilize HPC resources and exploit emerging/heterogeneous HPC architectures (e.g., GPU, Xeon Phi), while enabling simulations at previously inaccessible scales. Our goal is to develop performance portable and maintainable software to leverage the compute resources that become available over many years. Therefore, the poster also presents results of our efforts to increase the performance of DPD-LAMMPS, utilizing accelerators, general code optimization, as well as algorithmic improvements. In particular, we are leveraging the Kokkos C++ library [7][8] (recently included in LAMMPS) to portably accelerate 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. We have demonstrated that the reaction DPD kernels can be greatly accelerated with CUDA C [9], and expect to maintain a large fraction of those performance gains when converted to Kokkos. We are also improving the parallel efficiency of the SSA by restructuring/refactoring the MPI communications and the neighbor list construction for DPD-LAMMPS.

Nearly all materials that are critical to DoD technologies require specific material properties that are intimately linked to the material's microstructure. Therefore, the concepts of multiscale-modeling in support of next-generation materials design impacts many classes of microstructure-dependent materials, including biomolecules, polymers, energy-storage materials, energetic materials, and armor materials. The work presented in this poster enables the utilization of HPC resources for unprecedented simulations of phenomena not previously possible, guiding the development of emerging technologies and innovations.

REFERENCES

[1] M. Lísal, J.K. Brennan, and J.B. Avalos, J. Chem. Phys., 135, 204105 (2011).

[2] J.K. Brennan, M. Lísal, J.D. Moore, S. Izvekov, I.V. Schweigert, and J.P. Larentzos, J. Phys. Chem. Lett., 5, 2144 (2014).

[3] S. Plimpton, J. Comp. Phys., 117, 1 (1995).

[4] M. Galassi et al, GNU Scientific Library Reference Manual (3rd Ed.), ISBN 0954612078.

[5] A. C. Hindmarsh, et al. , "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," ACM Transactions on Mathematical Software, 31(3), pp. 363-396, 2005.

[6] J.P. Larentzos, J.K. Brennan, J.D. Moore, M. Lísal, and W.D. Mattson, Comput. Phys. Commun., 185, 1987 (2014).

[7] H.C. Edwards, C.R. Trott, "Kokkos: Enabling Performance Portability Across Manycore Architectures", Extreme Scaling Workshop (XSW), pp. 18-24, 2013.

[8] http://trilinos.org/packages/kokkos/

[9] C. Stone and R. Davis, “Techniques for solving stiff chemical kinetics on graphical processing units”, AIAA J. Propulsion and Power, v. 29, n. 4, 2013.