SoftwareX 11 (2020) 100389

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

RXMD: A scalable reactive molecular dynamics simulator for optimized time-to-solution

Ken-ichi Nomura [a,d,*], Rajiv K. Kalia [a,b,c,d], Aiichiro Nakano [a,b,c,d,e], Pankaj Rajak [f], Priya Vashishta [a,b,c,d]

[a] Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0241, USA
[b] Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0241, USA
[c] Department of Physics & Astronomy, University of Southern California, Los Angeles, CA 90089-0241, USA
[d] Department of Materials Science and Chemical Engineering, University of Southern California, Los Angeles, CA 90089-0241, USA
[e] Department of Biological Science, University of Southern California, Los Angeles, CA 90089-0241, USA
[f] Argonne Leadership Computing Facility, Argonne National Laboratory, Argonne, IL 60439, USA

ARTICLE INFO

Article history:
Received 12 February 2019
Received in revised form 25 October 2019
Accepted 17 December 2019

Keywords:
Reactive molecular dynamics
Parallel computing
Time-to-solution

ABSTRACT

RXMD is a parallel reactive molecular dynamics (RMD) simulator designed to perform large-scale RMD simulations on commodity laptop computers to supercomputer platforms. With extensive Time-to-Solution (TtoS) optimization techniques and scalable algorithms implemented, RXMD enables researchers to explorer materials design space that requires vast spatial-extent and long-time atomistic events with highly accurate chemistry. Extensible and modular software architecture allows to implement innovative optimization techniques. RXMD is one of core simulation engines at Material Genome Innovation for Computational Software (MAGICS) center, and is freely distributed as an open-source software from MAGICS website and Github. RXMD is also used as computational materials courseware to provide basic training of parallel RMD simulations for computational researchers and community.

Software metadata

Current software version	RXMD 3.0.
Permanent link to executables of this version	https://github.com/ElsevierSoftwareX/SOFTX_2019_32
Legal Software License	GPL-3
Computing platform / Operating System	*Linux/Unix, OS X*
Installation requirements & dependencies	Fortran 2008, MPI, OpenMP
If available Link to user manual - if formally published include a reference to the publication in the reference list	https://github.com/USCCACS/rxmd
Support email for questions	cacs@usc.edu

Code metadata

Current code version	RXMD 3.0
Permanent link to code/repository used of this code version	https://github.com/USCCACS/rxmd
Legal Code License	GPL-3
Code versioning system used	Github
Software code languages, tools, and services used	Fortran 2008, MPI, OpenMP
Compilation requirements, operating environments & dependencies	None
If available Link to developer documentation/manual	None
Support email for questions	cacs@usc.edu

* Corresponding author at: Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0241, USA.
E-mail address: knomura@usc.edu (K.-i. Nomura).

https://doi.org/10.1016/j.softx.2019.100389

1. Motivation and significance

Molecular dynamics (MD) is a widely-used materials modeling method, in which atoms are explicitly treated to investigate material of interest. MD method is often categorized by the level of theory to describe the interaction between atoms. For example, quantum MD (QMD) explicitly follows electron dynamics based on quantum mechanics, while classical MD employs simpler functions to approximate the complex electronic dynamics. Though QMD provides deep insights into underlying physics, the rapid increase in the computational complexity with system size prohibits simulating over thousands of atoms. This limitation is especially critical for materials simulations where a typical system size consists of millions of atoms to investigate local heterogeneity, such as dislocations and grain boundary, embedded within bulk of crystalline material.

One approach to circumvent the scalability problem is multiscale modeling where an accurate QM model is used to describe chemically reactive sites while scalable continuum method describes surrounding bulk mechanical property [1–5]. Such multiscale modeling approach has been quite successful for biological applications [6,7], however, it poses theoretical and technical challenges if hand-shake region, i.e. the boundary between different models, is not clearly defined and dynamically evolves over the course of simulation [5].

Another popular approach is to develop an interatomic interaction that is scalable and capable to describe targeted chemical reactions. This type of interatomic interaction is called "reactive" force field that consists of a dozen of energy functions, and requires extensive and sophisticated parameter optimizations. To date, a number of reactive force fields with accurate chemistry along with sound materials property have been proposed. ReaxFF [8,9] is one of reactive force fields that employs the bond-order concept and variable charge scheme, realizing dynamic formation and breaking of covalent-bonds and charge transfer between atoms. Several variable charge models, for example EEM [10], QEq [11], polarizable QEq [12], and ACKS2 (Atom-condensed Kohn–Sham DFT approximated to second order) [13], have been used in conjunction with ReaxFF. Fidelity of chemical reaction relies on the level of accuracy of theory used to train parameter sets. Although ReaxFF achieves highly accurate chemistry with several orders-of-magnitude less computational cost, the multi-core paradigm shift necessitated a novel scalable parallel software in order to fully take advantage of emerging and future computing platforms.

Currently several software packages that support ReaxFF MD are available [9]. LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [14] is an open-source community-based parallel molecular dynamics simulator that has been widely used in research community. The ReaxFF package in LAMMPS, called LAMMPS/User-ReaxC, is based on PuReMD (Purdue Reactive Molecular Dynamics) [15–17]. PuReMD implements highly efficient numerical algorithms to take advantage of state-of-the-art HPC architectures such as many-core and Graphical Processing Unite (GPU).

Amsterdam Modeling Suite (AMS) is a commercial software package that consists of various simulation engines including ReaxFF. AMS comes graphical user interface (GUI), utility tools to construct initial MD system, analyze and visualize MD trajectories, and many online manuals and tutorials that significantly increase users' productivity. In addition, AMS also offers advanced features, such as eReaxFF [18] and ACKS2 to further extend the simulation fidelity and its modeling capability.

GULP (the General Utility Lattice Program) [19] was originally designed to develop interatomic interactions, and later extended the capability to enable parallel MD simulation. GULP supports various classical force fields including ReaxFF. GULP also offers many parameter fitting features incorporating a wide range of material properties such as structural, energetic, mechanical, dielectric, and vibrational properties. These are useful to develop new interatomic potentials.

RXMD has been developed for computational scientists and engineers to perform RMD simulation on desktop to million-core supercomputers. RXMD provides excellent scalabilities as well as sound single-node performance.

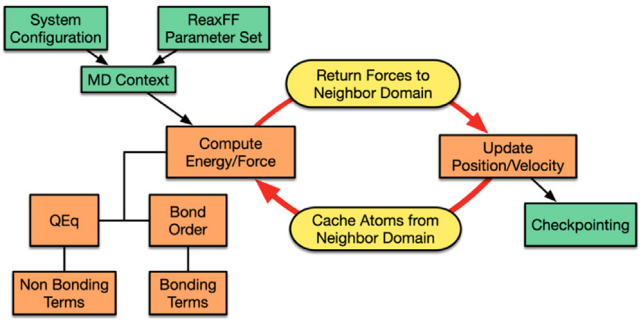

Fig. 1. Schematic of RXMD workflow.

2. Software description

RXMD is based on modern Fortran standard with minimal dependency on third-party library to keep the codebase simple and portable for existing and future computing architectures. RXMD employs an efficient domain decomposition scheme with standard six-way interprocess communication to ensure the sustainable algorithmic scalability. Message Passing Interface (MPI) library is used for interprocess communication across multiple nodes, while multithreading based on Open Multi-Processing (OpenMP) APIs realizes single-node concurrency. Computationally intensive subroutines, such as the bonded and non-bonded energy functions, the neighborlist construction, and the hessian computation in QEq, *i.e.* $\frac{\partial E_{Coulomb}}{\partial q_i \partial q_j}$ where E_{Coulomb} is the Coulomb energy and q_{i} and q_{j} is the charges of *i*- and *j*th atoms, are multithreaded. Shared data among threads are protected by the low-overhead atomic operation. We fuse the energy functions into a single parallel section to enhance the multithreading scalability and minimize overheads incurred by the fork-join thread model. Scalable I/O function implemented with MPI-I/O stores checkpoint file and other information such as atom connectivity. RXMD has been tested on various Fortran compilers and MPI libraries. The latest RXMD source code is freely available from the Github repository [20] as well as the MAGICS website [21].

2.1. Software architecture

Fig. 1 schematically presents a RXMD simulation workflow. The first step is to generate initial configuration file (***rxff.bin***), which contains atom types, positions, and system dimensions, etc. RXMD includes a utility tool called ***geninit*** to generate initial configuration. RXMD solves the Newton's equation of motion (indicated by the red arrows) and produces atomic trajectory for later analysis. One MD step consists of (1) cache atom information from neighbor domains, (2) compute energy and force, (3) return the computed force value to neighbor domains, and (4) update atom position and velocity. The step (1) and (3) are done through interprocess communication using MPI. The energy and force computations consist of several sub-steps, including the bond-order calculation, the determination of atomic charge using QEq, and the force and energy calculation based on bonding and non-bonding functions.

2.1.1. Directory structure

The list below shows several key files and directories in the RXMD repository.

DAT: directory to store checkpoint data and the initial config file (***rxff.bin***). Runtime flag `-o` may be used to overwrite the directory path.

ffield: text file of ReaxFF parameter set. Runtime flag `-f` may be used to overwrite the file path.

rxmd.in: text file contains RXMD input parameters. Runtime flag `-i` overwrites the file path.

init: directory to store utility and create initial configuration using **geninit**.

src: directory to store the RXMD source files

Directory ***src*** contains RXMD source files. In the source files, subroutines and functions are grouped by their functionality indicated by the filename. Most important source files are listed below.

main.F90: MD driver that contains main routine and several utility functions

init.F90: subroutines for system initialization

param.F90: ReaxFF force field parser

pot.F90: subroutines for energy and force computations

qeq.F90: atomic charge computation using QEq scheme

pqeq.F90: core/shell charge computation using PQEq scheme

module.F90: modules for common variables

comm.F90: subroutines for interprocess communication

fileio.F90: parallel I/O routines for checkpointing, atom connectivity, and data visualization

bo.F90: subroutines for bond-order computation

Directory ***init*** contains Makefile and source files to build the ***geninit*** utility tool. To prepare an initial configuration, ***geninit*** requires two input files: (1) a ReaxFF force field filed (***ffied***) designed for the system of interest; and (2) an input file that contains simulation system information such as the lattice parameters of MD box, and element name and coordinate of atoms. Fig. 2 presents the input file format. The first line is the total number of atoms followed by a string to describe the system. The second line stores the six lattice parameters, three edges (a, b, and c) in Å unit and three angles (α, β, γ). From the third line until the end of file stores element name and x, y, z positions in fractional coordinates per line.

2.2. Software functionalities

In this section, we introduce RXMD functionalities such as, the extended Lagrangian method, Shift-Collapse (SC) algorithm and recently added a generalized polarizable charge equilibration scheme, called ReaxPQ+ [22].

The extended Lagrangian method adapted in RXMD was proposed by Nicklasson et al. in the context of quantum molecular dynamics (QMD) [23–25]. It circumvents the computationally-expensive energy minimization step by introducing the fictitious dynamics of wave functions, achieving the excellent energy conservation and the significant reduction in computing time. The extended Lagrangian method should be used with caution due to the known energy drift problem in the XL-BOMD scheme [26]. The energy conservation however can be substantially improved by the inertial extended Lagrangian/self-consistent field (iEL/SCF) scheme recently proposed by Leven et al. [26].

To achieve linear scaling to evaluate range-limited interatomic interactions on a single node, a standard strategy is to organize atomic positions into bins, often called cells, and traverse atom information using linked-list. Based on the translation and reflection invariance of interatomic potential, SC algorithm completely eliminates redundant computations while minimizing data transfers among computing nodes. A generalization of the SC algorithm, called renormalized SC (RSC), has been applied to ReaxPQ+ and achieved 5.0x reduction in the TtoS [22]. RSC is currently in a development branch and not officially supported.

Despite its enormous success, QEq scheme employs atomic charge model thus the electronic contribution in dielectric constant is ignored. To address the issue, a polarized PQEq model, called PQEq scheme [27–29], based on a core–shell model has been proposed. PQEq describes sufficiently accurate dielectric behavior compared to quantum mechanics for various organic and inorganic materials. Further extending the PQEq scheme, RXMD implements a generalized PQEq scheme (ReaxPQ+) to incorporate the effect of electric field (see Fig. 3(a)). To describe the dielectric response of materials, ReaxPQ+ introduces the effect of an external electric field by solving $\mathbf{F}_{\text{inter}} + \mathbf{F}_{\text{external}} = \mathbf{F}_{\text{intra}}$ to determine the shell charge position $\mathbf{r}_{is}$, where, $\mathbf{F}_{\text{intra}}$ and $\mathbf{F}_{\text{inter}}$ are the Coulombic interactions within an atom and between neighbor atoms, and $\mathbf{F}_{\text{external}}$ is the induced force by the external electric field. We have successfully applied the ReaxPQ+ framework to high energy-density polymers, Al_2O_3 crystal, and MoS_2 monolayer.

Fig. 3(b) and (c) present the weak and strong scaling performances of ReaxPQ+ on Intel Xeon Phi (KNL) supercomputer called Theta at Argonne Leadership Computing Facility (ALCF). During the isogranular-scaling (weak scaling), the number of atoms per MPI rank (N/P) is kept constant, i.e., 25,576 MoS_2 atoms per MPI rank. In the isogranular scaling, the speedup on P cores is defined as the ratio of a product of the total number of atoms in the simulation cell and the number of RXMD steps per second with a reference benchmark based on 64 Xeon Phi cores. Here, the number of cores during the benchmark is varied from 64 to 131,072, consisting of up to 3,221,224,472 atoms. We have achieved a parallel efficiency of 0.989, see Fig. 3(b). We have also performed weak-scalability with smaller granularity (RDX system involving 672 atoms per core) and observed no degradation of parallel efficiency up to 16,384 KNL cores. In the strong-scaling performance test (Fig. 3(c)), a fixed problem size of 50,331,648 MoS_2 atoms is used. The numbers of cores used during the benchmark varies between 2048 and 32,768. We have obtained the strong-scaling parallel efficiency of 0.766, which corresponds to 12.26x reduction in the TtoS.

3. Illustrative examples

In this section, we present how to download RXMD, build necessary executables, and perform several example simulations. First step is to clone the RXMD repository by git command.

`git clone` https://github.com/USCCACS/rxmd.git

Fortran compiler, MPI library, and MPI Fortran 90 binding have to be properly installed and configured to build the RXMD executable. **mpif90 –version** tells whether the MPI binding is available and which Fortran compiler is being used.

```
12   "Poly(ethylene) crystal"
7.40000   4.93000   2.53400   90.000   90.000   90.000
C      0.038000000   0.935000000   0.250000000
C      0.962000000   0.065000000   0.750000000
H      0.197000000   0.970000000   0.250000000
```

Fig. 2. Example of geninit input file. Atomic coordinates are in fractional coordinates.

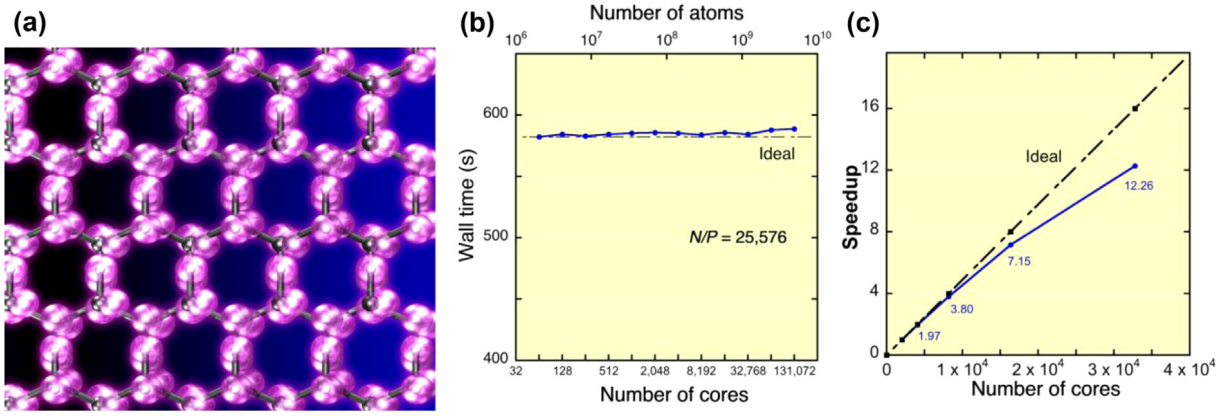

Fig. 3. (a) Distribution of shell charges (purple) of polyethylene crystal subjected to an electric field. (b) Weak-scaling performance of the ReaxPQ+ algorithm using 25,576 MoS_2 atoms per KNL core, in which the number of KNL cores P ranges from 64 to 131,072. The blue line shows the wall-clock time for single MD step in second. (c) Strong-scaling speedup of the ReaxPQ+ algorithm using a constant problem size - 50,331,648-atom MoS_2 system on P cores (P = 2048, …, 32,768) of Theta. The ideal speedup and the measured wall-clock time are shown as black-dotted and blue lines, respectively.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Directory **config/** contains several predefined build settings. Through the examples, we use a standard make file **config/make_example.inc** with GNU Fortran compiler. First step is to copy **config/make_example.inc** as **make.inc**, and type **make all**, which builds the RXMD executable **rxmd** and the utility tool **geninit** under directory **init/** used for all examples.

```
cp config/make_example.inc make.inc
make all
```

Directory **examples/** contains three subdirectories, each of which demonstrates several core functionalities of RXMD.

1-reaxff: RMD simulation with single domain.

2-reaxff-dc: RMD simulation with multiple domains.

3-reaxpq+: ReaxPQ+ simulation with single domain with electric field.

Each directory contains a crystalline polyethylene unit cell (**init.xyz**), ReaxFF parameter set (**ffield**), RXMD input parameters (**rxmd.in**), and **Makefile** to perform the demo simulations.

3.1. RMD with single and multiple domains

The first two examples demonstrate how to run RMD simulation on single domain (**1-reaxff**) and multiple domains based on divide-and-conquer approach (**2-reaxff-dc**). Unless explicitly specified, file and directory paths are relative to **rxmd/examples**.

By changing directory to **1-reaxff/** and typing **make** command, following commands are executed.

```
../../init/geninit -i input.xyz -f ffield -o DAT/ -mc 2 3 5
mpirun -np 1 ../../rxmd
```

The first command creates an initial config using the unit cell structure described in **input.xyz**, replicate the unit cell by 2, 3, and 5 times in x, y, and z directions, respectively. ReaxFF parameter file **ffield** is necessary here to map between element name (string) stored in **init.xyz** and atom type (integer) during RMD simulation. Flag **-o** specifies output directory to place the output file **rxff.bin**, which contains the initial configuration data. After the computation finishes, the atom trajectories are saved under directory **1-reaxff/DAT/** using XYZ format. Any standard molecular visualization can be used to visualize the trajectory.

To achieve the best TtoS for your system of interest, precise control on simulation domain size and efficient use of available computing resources are crucial. Second example shows how to prepare an initial configuration to perform two-domain RMD simulation. Although this is the simplest case of multiple domain simulation, the same procedure can be used for RMD simulation consists of million domains. Directory **2-reaxff-dc/** contains an example of multidomain RMD simulation using two MPI ranks. The **make** command from directory **2-reaxff-dc/** will create an initial configuration and start a short RMD simulation with multi-domain.

3.2. ReaxPQ+ with electric field

Directory **3-reaxpq+/** contains an example of ReaxPQ+ simulation where a polyethylene crystal is subjected to a static electric field of 0.01 Volts/Å in the x direction. Electric field is applied by the runtime flag **-e** (or **–efield**) followed by two numbers; the direction (integer) and the strength of electric field (float) in a unit of Volts/Å, respectively. Use **make** command from **3-reaxpq+/** to perform the example.

```
../../init/geninit -i input.xyz -f ffield -o DAT/ -mc 2 3 5
mpirun -np 1 ../../rxmd
```

XYZ files that contains the positions of core and shell charges are created under **DAT/3-reaxpq+/**, which can be used to compute the polarizability of the system as a function of applied electric field.

4. Impact

The early generation of RXMD software showed excellent scalability up to one billion-atom system on BlueGene/L at ALCF [30]. The early benchmark opened up the possibility for researchers to investigate a wide range of materials properties, in which the

accurate description of vast spatial extent and accurate chemical reaction is inevitable. Thanks to the simple and modular architecture, RXMD has been serving a software-development platform for novel algorithms to improve the TtoS, including the extended-Lagrangian method to bypass the speed-limiting QEq step [31] and renormalized shift-collapse (RSC) method to absolutely eliminate redundant computations [22,32]. RXMD continues to support excellent scalability on single computing node as well as on many high-end supercomputing architectures at national leadership computing facilities.

Large-scale RMD has broad applications across many materials research fields. One example is sulfur-induced embrittlement, where minute amount of impurity makes tough metal exhibit catastrophic brittle failure. Using 48 million-atom RMD simulation, Chen et al. found that the presence of the sulfur impurities in nickel grain boundary resulted in the order-of-magnitude reduction in shear strength, thereby the crack propagation pathways was confined along grains [33]. With RXMD, scientists can design and investigate novel and unique scientific and engineering problems by fully taking advantage of available computing resources. RXMD has been used to investigate stress corrosion cracking [33], shock-induced chemical reactions [34,35], underwater bubble collapse [36], synthesis of nanocarbon [37] and 2D materials [38,39], force-field training [40,41], and a number of novel algorithmic studies [31,32].

In addition, RXMD has been used as part of courseware of the MAGICS materials software workshop. All training modules and courseware materials are freely distributed from the MAGICS website [42] to create an open educational platform for future computational researchers.

5. Conclusions

In summary, RXMD is a scalable and portable parallel RMD simulator specifically designed to perform large-scale RMD simulation using commodity laptop computers to supercomputer platforms. With extensive optimizations on the scalability and the TtoS, researchers are now able to perform quantum-mechanics level accurate RMD simulation at unprecedented spatial and temporal scale. The simple and modular software architecture allows researchers to investigate novel and innovative optimization techniques. RXMD has been shown to be highly scalable from small-to-large granularity problems, which allows to study a wide range of materials-science problems. The software has been developed at the DOE MAGICS center, and the RXMD source code is freely distributed on Github and the MAGICS website. To date, RXMD has been shared with many research groups at universities and national laboratories, and recently used as computational materials training courseware for graduate students and early career researchers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, USA, Basic Energy Sciences, USA, under Award Number DE-SC0014607.

References

[1] Kikuchi H, Kalia R. Collaborative simulation grid: multiscale quantum-mechanical/classical atomistic simulations on distributed PC clusters in the US and japan, Supercomputing, In: ACM/IEEE 2002 conference, vol. 00; 2002. p. 1–8.
[2] Lidorikis E, Bachlechner ME, Kalia RK, Nakano A, Vashishta P, Voyiadjis GZ. Coupling length scales for multiscale atomistics-continuum simulations: Atomistically induced stress distributions in Si/Si3N4nanopixels. Phys Rev Lett 2001;87:86104–1–86104–4.
[3] Ogata S, Lidorikis E, Shimojo F, Nakano A, Vashishta P, Kalia RK. Hybrid finite-element/molecular-dynamics/electronic-density-functional approach to materials simulations on parallel computers. Comput Phys Comm 2001;138:143–54.
[4] Ogata S, Shimojo F, Kalia RK, Nakano A, Vashishta P. Hybrid quantum mechanical/molecular dynamics simulation on parallel computers: Density functional theory on real-space multigrids. Comput Phys Comm 2002;149:30–8.
[5] Takemiya H, et al. Sustainable adaptive grid supercomputing: multiscale simulation of semiconductor processing across the pacific. In: SC '06: Proceedings of the 2006 ACM/IEEE conference on supercomputing; 2006. p. 23.
[6] Senn HM, Thiel W. QM/MM methods for biomolecular systems. Angew Chem Int Edn 2009;48:1198–229.
[7] Warshel A, Levitt M. Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. J Mol Biol 1976;103:227–49.
[8] Van Duin ACT, Dasgupta S, Lorant F, Goddard WA. ReaxFF: A reactive force field for hydrocarbons. J Phys Chem A 2001;105:9396–409.
[9] Senftle TP, et al. The reaxFF reactive force-field: development, applications and future directions. Npj Comput Mater 2016;2:15011, Review Article, 03/04/online.
[10] Mortier WJ, Ghosh SK, Shankar S. Electronegativity equalization method for the calculation of atomic charges in molecules. J Am Chem Soc 1986;108(15):4315–20.
[11] Rappé AK, Goddard WA. Charge equilibration for molecular dynamics simulations. J Phys Chem 1991;95:3358–63.
[12] Naserifar S, Brooks DJ, Goddard WA, Cvicek V. Polarizable charge equilibration model for predicting accurate electrostatic interactions in molecules and solids. J Chem Phys 2017;146(12). 124117.
[13] Verstraelen T, Ayers PW, Van Speybroeck V, Waroquier M. ACKS2: Atom-condensed Kohn–Sham DFT approximated to second order. J Chem Phys 2013;138(7).
[14] Plimpton S. Fast parallel algorithms for short-range molecular dynamics. J Comput Phys 1995;117(1):1–19.
[15] Aktulga HM, Fogarty JC, Pandit SA, Grama AY. Parallel reactive molecular dynamics: Numerical methods and algorithmic techniques. Parallel Comput 2012;38(4–5):245–59.
[16] Kylasa SB, Aktulga HM, Grama AY. PuReMD-GPU: A reactive molecular dynamics simulation package for GPUs. J Comput Phys 2014;272:343–59.
[17] Aktulga HM, Pandit SA, van Duin ACT, Grama AY. Reactive molecular dynamics: Numerical methods and algorithmic techniques. SIAM J Sci Comput 2012;34(1):C1–23.
[18] Islam MM, Kolesov G, Verstraelen T, Kaxiras E, van Duin ACT. eReaxFF: A pseudoclassical treatment of explicit electrons within reactive force field simulations. J Chem Theory Comput 2016;12(8):3463–72.
[19] Gale JD. GULP: A computer program for the symmetry-adapted simulation of solids. J Chem Soc-Faraday Trans 1997;93(4):629–37.
[20] MAGICS. RXMD : Linear-scaling parallel reactive molecular dynamics simulation engine. https://github.com/USCCACS/RXMD.
[21] MAGICS. MAGICS software download site. https://magics.usc.edu/software-downloads/.
[22] Liu K, et al. Shift-collapse acceleration of generalized polarizable reactive molecular dynamics for machine learning-assisted computational synthesis of layered materials. In: 2018 IEEE/ACM 9th workshop on latest advances in scalable algorithms for large-scale systems, scalA; 2018. p. 41.
[23] Niklasson AMN. Extended Born–Oppenheimer molecular dynamics. Phys Rev Lett 2008;100(12).
[24] Cawkwell MJ, Niklasson AMN. Energy conserving, linear scaling Born–Oppenheimer molecular dynamics. J Chem Phys 2012;137(13).
[25] Niklasson AMN, Cawkwell MJ. Generalized extended Lagrangian Born–Oppenheimer molecular dynamics. J Chem Phys 2014;141(16).
[26] Leven I, Head-Gordon T. Inertial extended-Lagrangian scheme for solving charge equilibration models. Phys Chem Chem Phys 2019;21(34):18652–9.
[27] Kwon S, Naserifar S, Mo Lee H, Goddard WA. Polarizable charge equilibration model for transition-metal elements. J Phys Chem A 2018;122:9350–8.
[28] Naserifar S, Brooks DJ, Goddard WA, Cvicek V. Polarizable charge equilibration model for predicting accurate electrostatic interactions in molecules and solids. J Chem Phys 2017;146. 124117.

[29] Oppenheim JJ, Naserifar S, Goddard WA. Extension of the polarizable charge equilibration model to higher oxidation states with applications to Ge, As, Se, Br, Sn, Sb, Te, I, Pb, Bi, Po, and at elements. J Phys Chem A 2018;122:639–45.
[30] Nomura K, Kalia RK, Nakano A, Vashishta P. A scalable parallel algorithm for large-scale reactive force-field molecular dynamics simulations. Comput Phys Comm 2008;178:73–87.
[31] Nomura K, Small PE, Kalia RK, Nakano A, Vashishta P. An extended-Lagrangian scheme for charge equilibration in reactive molecular dynamics simulations. Comput Phys Comm 2015;192:91–6.
[32] Kunaseth M, Hannongbua S, Nakano A. Shift/collapse on neighbor list (SC-NBL): Fast evaluation of dynamic many-body potentials in molecular dynamics simulations. Comput Phys Comm 2019;235:88–94.
[33] Chen HP, et al. Embrittlement of metal by solute segregation-induced amorphization. Phys Rev Lett 2010;104. 155502.
[34] Nomura K, Kalia RK, Nakano A, Vashishta P, van Duin ACT, Goddard III WA. Dynamic transition in the structure of an energetic crystal during chemical reactions at shock front prior to detonation. Phys Rev Lett 2007;99. 148303.
[35] Nomura K, Kalia RK, Nakano A, Vashishta P. Reactive nanojets: Nanostructure-enhanced chemical reactions in a defected energetic crystal. Appl Phys Lett 2007;91:2005–8.
[36] Nomura K, Kalia RK, Nakano A, Vashishta P, van Duin ACT. Mechanochemistry of shock-induced nanobubble collapse near silica in water. Appl Phys Lett 2012;101. 073108.
[37] Nomura K, et al. Nanocarbon synthesis by high-temperature oxidation of nanoparticles. Sci Rep 2016;6. 24109.
[38] Hong S, et al. Computational synthesis of MoS_2 layers by reactive molecular dynamics simulations: Initial sulfidation of MoO_3 surfaces. Nano Lett 2017;17:4866–72.
[39] Hong S, et al. Chemical vapor deposition synthesis of MoS_2 layers from the direct sulfidation of MoO_3 surfaces using reactive molecular dynamics simulations. J Phys Chem C 2018;122:7494–503.
[40] Cheng HC, Rajak P, Sheng C, Kalia RK, Nakano A, Vashishta P. A high-throughput multiobjective genetic-algorithm workflow for in situ training of reactive molecular-dynamics force fields. In: 24th High Performance Computing Symposium. Pasadena, CA: Society for Modeling and Simulation International (SCS); 2016.
[41] Mishra A, et al. Multiobjective genetic training and uncertainty quantification of reactive force fields. Npj Comput Mater 2018;4:42.
[42] MAGICS. MAGICS workshop courseware download site. https://magics.usc.edu/workshop/courseware.