QXMD: An open-source program for nonadiabatic quantum molecular dynamics

Shimojo, Fuyuki ; Fukushima, Shogo ; Kumazoe, Hiroyuki ; Misawa, Masaaki ; Ohmura, Satoshi ; Rajak, Pankaj ; Shimamura, Kohei ; Bassman…

(Citation)
SoftwareX,10:100307-100307

(Issue Date)
2019-07

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90006970

SoftwareX 10 (2019) 100307

Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

QXMD: An open-source program for nonadiabatic quantum molecular dynamics

Fuyuki Shimojo [a], Shogo Fukushima [a], Hiroyuki Kumazoe [a], Masaaki Misawa [b], Satoshi Ohmura [c], Pankaj Rajak [d], Kohei Shimamura [e], Lindsay Bassman [f], Subodh Tiwari [f], Rajiv K. Kalia [f], Aiichiro Nakano [f,*], Priya Vashishta [f]

[a] Department of Physics, Kumamoto University, Kumamoto 860-8555, Japan
[b] Faculty of Science and Engineering, Kyushu Sangyo University, Fukuoka 813-8503, Japan
[c] Research Center for Condensed Matter Physics, Hiroshima Institute of Technology, Hiroshima 731-5193, Japan
[d] Leadership Computing Facility, Argonne National Laboratory, Argonne, IL 60439, United States
[e] Graduate School of System Informatics, Kobe University, Kobe 657-8501, Japan
[f] Collaboratory for Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089-0242, United States

ARTICLE INFO

Article history:
Received 25 February 2019
Received in revised form 29 June 2019
Accepted 26 July 2019

Keywords:
Nonadiabatic quantum molecular dynamics
Parallel computing
Hands-on training

ABSTRACT

QXMD is a scalable, parallel program for Quantum Molecular Dynamics simulations with various eXtensions. Its simulation engine is based on (time-dependent) density functional theory using pseudopotentials and a plane-wave basis set, while extensions include nonadiabatic electron–nuclei dynamics and multiscale shock technique. QXMD serves as a community-development platform for new methods and algorithms, a research platform on high-end parallel supercomputers, and an educational platform for hands-on training.

Code metadata

Current code version	*V3.2*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2019_49
Code Ocean compute capsule	https://codeocean.com/capsule/691377f8-8b90-4571-8098-1f0f8887e4a8/tree?ID=558be63ff8ff48d19859829a73a41f74
Legal Code License	*GNU AGPLv3*
Code versioning system used	*git*
Software code languages, tools, and services used	*Fortran2003, MPI, FFT*
Compilation requirements, operating environments & dependencies	
If available Link to developer documentation/manual	https://usccacs.github.io/QXMD/index.html
Support email for questions	cacs@usc.edu

1. Motivation and significance

Quantum molecular dynamics (QMD) is a widely used simulation method to study the dynamic behavior of materials [1–3]. QMD follows the trajectories of each atom in a system, while computing interatomic forces quantum mechanically within the framework of density functional theory (DFT) [4,5]. In particular, nonadiabatic QMD (NAQMD) describes electronic excitations and transitions between excited electronic states assisted by atomic motions, thereby describing excitation dynamics involving electrons and nuclei [6–8].

The ever-increasing computing power of high-end parallel supercomputers [9] is enabling the study of complex material processes encompassing unprecedentedly large spatiotemporal scales, while incorporating higher levels of physical and chemical fidelity [10]. However, it also poses enormous algorithmic and computational challenges for scaling QMD simulations up

* Corresponding author.
E-mail address: anakano@usc.edu (A. Nakano).

https://doi.org/10.1016/j.softx.2019.100307

to emerging computing architectures [10]. QXMD is a software platform for developing new eXtensions and eXperimental features to make QMD simulations metascalable, *i.e.*, "design once, scale on new architectures" [11,12]. The core QXMD simulation engine is based on DFT using pseudopotentials and a plane-wave basis set [2,13]. Extensions implemented thus far include (1) linear-response time-dependent density functional theory (LR-TDDFT) [14] with range-separated exact-exchange correction to describe excitonic binding [15], (2) excited-state forces to describe photo-excited energy landscapes [7], (3) multiscale shock technique (MSST) to describe shock-front dynamics [16], (4) Berry-phase computation of electric polarization [17,18], and (5) divide-and-conquer DFT (DC-DFT) algorithm [19,20] for $O(N)$ QMD simulations [21], in which the computation scales linearly with the number of electrons, N. With these extensions, QXMD has also been used to study a wide range of quantum-dynamical processes in various materials and energy applications on massively parallel supercomputers [22–28].

In addition to serving as both community-development and high-end computational research platforms as described above, elementary features of the QXMD software have been used in our MAGICS (MAterials Genome Innovation for Computational Software) workshops to train researchers on the basics of QMD and NAQMD simulations. In this paper, we outline the architecture of the QXMD software. An earlier version of the software is described in Ref. [29], while subsequent extensions are detailed in Ref. [11]. The present paper focuses on the principal features of the QXMD software and corresponding tutorial material, which can be used in various workshop and classroom settings. Related basics of pseudopotential and plane-wave implementations of DFT are found, *e.g.*, in Ref. [13].

2. Software description

QXMD is a highly scalable, parallel program written in Fortran for performing QMD simulations. The program is parallelized based on hybrid space-band decomposition [11] and uses the message passing interface (MPI) library [30] to perform internode communication. QXMD also requires a fast Fourier transform (FFT) library, such as FFTW3 [31]. Strong-scaling tests show that time-to-solution is reduced by a factor of 2.5 on 64 processor cores compared to 16 cores for a 1,284-atom system. This represents decent scaling for typical QMD applications. While the current QXMD software distribution is for tutorial purposes on small-to-medium scale computing platforms, achieving scalability on high-end supercomputers requires our lean divide-and-conquer density functional theory (LDC-DFT) algorithm. As described in Ref. [12], the LDC-DFT algorithm has achieved 98.4% of the perfect speedup on 786,432 IBM Blue Gene/Q cores for a 50,331,648-atom system. The advanced LDC-DFT feature will be added in a future release of QXMD.

The program is run from the command line, taking as input the initial positions (and optionally velocities) of all atoms in the system, pseudopotentials for each species of atom, and various input parameter settings specific to the simulation being performed. Upon execution, output data (*e.g.*, total energy, trajectories of atoms, Kohn–Sham energies, partial density of states) are written to text files inside an output data directory after each time step of the simulation.

2.1. Software architecture

After downloading QXMD and setting the working directory to QXMD/, the following directories/files will be present:

- Docs/: directory containing user manual for QXMD.
- Examples/: directory containing input files for example simulations.
- Include/: directory containing FFT library.
- LIB/: directory containing pseudopotential files for example problems.
- Makefile: file used to configure and compile QXMD.
- Sources/: directory containing QXMD source code.
- util/: directory containing codes for post-processing output data from QXMD.

The LIB directory contains several pseudopotential files for illustration and tutorial purposes. Additional pseudopotential files can be requested by email as described in the user manual or on the github development page. Source code is contained in the Sources/ directory and is divided into a set of modules that each perform different components of the QMD simulation, including parallelization and internode communication using MPI, matrix multiplication, integration, reading input, writing output, and setting default parameters. Key modules providing the main functionality of the program are the following:

- chgdns: code for calculating, checking, outputting and mixing charge densities.
- eigen: code for solving the Kohn–Sham eigenvalue problem using a preconditioned conjugate gradient method.
- engrad: code for computing gradients of the kinetic and nonlocal pseudopotential energies, as well as the whole system Hamiltonian.
- fermi: code for setting electronic occupation numbers.
- force: code for computing forces and internal stress tensor.
- ftmain: the main program for QXMD.
- nlkbpp: code for nonlocal pseudopotential calculation *via* the Kleinmann and Bylander (KB) method [32].
- pcc: code for computing the partial core correction [33].
- ppkb: code for local pseudopotential calculation *via* the KB method [32].
- pwlda: code to compute exchange–correlation functional based on local density approximation, generalized gradient approximation [34], van der Waals correction (DFT+D) [35], *etc.*
- tddft_fssh: code for implementation of NAQMD based on time-dependent DFT (TDDFT) and surface hopping [36].

2.2. Configuration and compilation

QXMD is configured and compiled using the makefile provided with the distribution. The makefile includes preconfigured setups for various supercomputing architectures. A complete list of supported machines can be obtained by running the command 'make help'. QXMD is configured by running 'make [machine_name]', where the user should specify the name of the particular machine on which QXMD is to be run. After successful completion of the configuration, QXMD is compiled by running 'make qxmd' for a serial executable, or 'make qxmdmpi' for a parallel executable. Note that a parallelized make is enabled by default.

2.3. Software functionalities

QXMD performs QMD simulations under three main paradigms: (i) adiabatic QMD, where the electrons stay in their ground state; (ii) NAQMD based on TDDFT, in which electrons can nonradiatively hop between energy levels; and (iii) MSST to simulate shock-front dynamics. QXMD also supports *in-situ* analyses, such as stress tensor and atomic charge calculations, as well as multiple data dumps such as wave functions, charge density and local potential. To minimize I/O, each data dump is turned off by default as these files are very large, though they can optionally be performed every nth ionic step based on the user's choice.

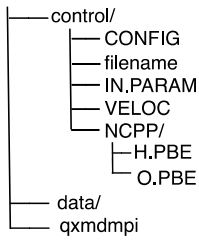

Fig. 1. Tree structure of the directories and files necessary to run a QXMD simulation.

3. Illustrative examples

3.1. Running a generic simulation

In order to run QXMD, there are a few mandatory directories and files that must be present in the correct hierarchy (Fig. 1). The working directory, from which a QXMD job is launched, must contain a control/ and data/ directory. The control/ directory contains setup files/directories as described below:

- CONFIG: configuration file detailing the atomic coordinates of the system in real or normalized coordinates.
- filename: simple text file containing the path to the main input file.
- IN.PARAM: main input file containing various input settings and parameters for the simulation.
- VELOC: optional file containing the initial velocities for each atom in the system.
- NCPP/: directory containing pseudopotential files for atomic species.

The data/ directory stores all output files generated during the simulation. Fig. 1 shows an example directory tree for successfully running a simulation of a water molecule.

There are sample input files for various types of simulations in the Examples/ directory, including optimization of water, adiabatic QMD of a water molecule, NAQMD of monolayer $MoSe_2$, and MSST simulation of SiO_2. The NAQMD and MSST examples are explained in more detail in the sections that follow.

3.2. NAQMD simulation of monolayer $MoSe_2$

In this example, we perform an NAQMD simulation of monolayer $MoSe_2$ [26,37]. In QXMD, this is carried out by promoting a set of electrons from their ground states to excited states, which mimics the dynamics in a system after photo-excitation by a laser. A sample input file can be found in Examples/03_NAQMD/01_MoSe2/control. The key section in the input parameter file, IN.PARAM, for running an NAQMD simulation is the ***TDDFT-MD** section. In this section, the "on/off" variable must first be set to true. The changes in electronic occupation numbers for the relevant energy bands must also be defined with the "occupations" variables. In this example, we promote all four electrons in the two highest occupied orbitals (energy bands 35 and 36) to the two lowest lying unoccupied orbitals (energy bands 37 and 38). Note that if spin polarization is not used, both electrons in one orbital must be moved to a second orbital, otherwise one would need to define whether a single electron being moved is spin-up or spin-down.

Once the simulation completes, there are several utility files available for post-processing of the output data. The main function of the utility files is to transform the raw data output by QXMD into a format that is easily read by existing graphing and visualization software. Here, we explain how to use two of these utility files: 'eig_exocc.f' for creating a plot of the Kohn–Sham eigenenergies and their occupation numbers versus time, and 'gcube.f90' for producing images of the charge densities of various energy bands. 'eig_exocc.f' can be found in Examples/03_NAQMD/01_MoSe2/analysis/eig. The utility program takes the path of the output data directory as a command line argument. After compiling the utility program, the resultant executable can be run with './eig_exocc -d ../../data'. This will produce three files which contain information on the Kohn–Sham energy eigenvalues and their occupation numbers. They can be used to compile all the information into one graph by running the gnuplot script 'plot_eig.gnu', as shown in Fig. 2(a).

'Gcube.f90' can be found in Examples/03_NAQMD/01_MoSe2/analysis/Gcube. This utility program takes multiple command-line arguments, including the path to the output data, as well as options to choose the frequency of times steps and the range of Kohn–Sham eigenstates for which to create Gaussian cube files. After compilation, the executable can be run with './gcube -d ../../data -n 101 -ib 36 -eb 37', which will create Gaussian cube files for every 101st time step for bands 36 through 37. The resultant data files can be used as input for many visualization software packages (*e.g.* VMD) to create an image, shown in Fig. 2(b).

3.3. MSST simulation of SiO_2

In this example, we use QXMD to study shock on SiO_2(α-quartz) using MSST [16]. In MSST, shock is assumed to be planar, and usually a rectangular parallelepiped computation cell is assumed. Here, we employ an omni-directional variant of MSST developed by Shimamura et al. [38,39], which allows us to study shock in any crystallographic direction.

A sample input file can be found in Examples/04_MSST/01_SiO2/control. To enable MSST, the method of dynamics in the ***molecular dynamics** section must be changed to option '*10*'. Shock speed and direction must be provided by setting the "shock wave velocity" input parameter, also in the ***molecular dynamics** section. In this example, we have applied shock with a velocity of 7.2 km/s in the [210] crystallographic direction by setting *shockspeed* to 7,200 m/s and *nshockv*(1:3) to (2,1,0). Particle velocity, pressure and temperature obtained from the Lagrangian after each iteration is written in the 'md_hug.d' file found in the output data directory. Fig. 3, (a) and (b), shows the initial and final configurations after applying shock.

The utility program 'toPDBcell.f' in Examples/04_MSST/01_SiO2/analysis/PDB can be used to create a PDB (protein data bank) formatted file of the atomic trajectories. The program takes the path of the output data directory as a command-line argument. Output PDB files can be visualized in many existing software packages such as VMD or OVITO. After compiling the utility program, the resultant executable can be run with './toPDBcell -d ../../data'. This will produce a PDB file named 'config.pdb', which contains the atomic trajectories of all atoms. This data can be used to create Fig. 3.

4. Impact

Three major impacts of the QXMD software are to: (1) provide an open-source, community code-development platform for advanced QMD and NAQMD simulation methods and algorithms; (2) serve as a research platform on massively parallel supercomputers; and (3) disseminate the extensive hands-on tutorial materials to train a broad scientific community.

With often-disruptive advancements in computer architectures [9], it is necessary to continuously develop new algorithms

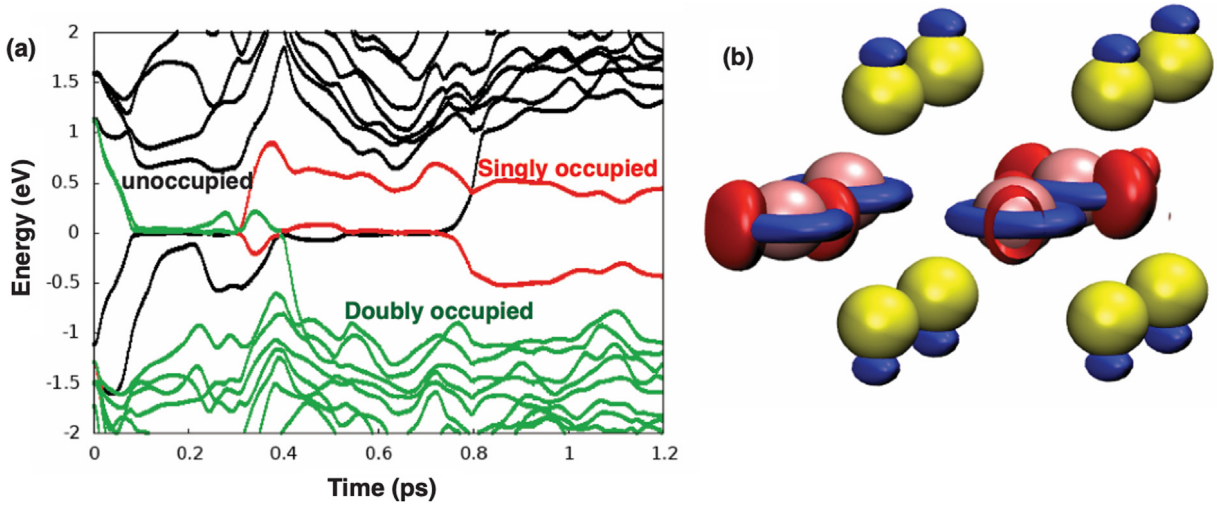

Fig. 2. (a) Kohn–Sham energy eigenvalues that are empty (black), doubly occupied (green) and singly occupied (red) plotted versus time. (b) Charge densities of the 36th (blue) and 37th (red) energy bands juxtaposed atop the $MoSe_2$ monolayer system with Mo and Se atoms shown in pink and yellow, respectively . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

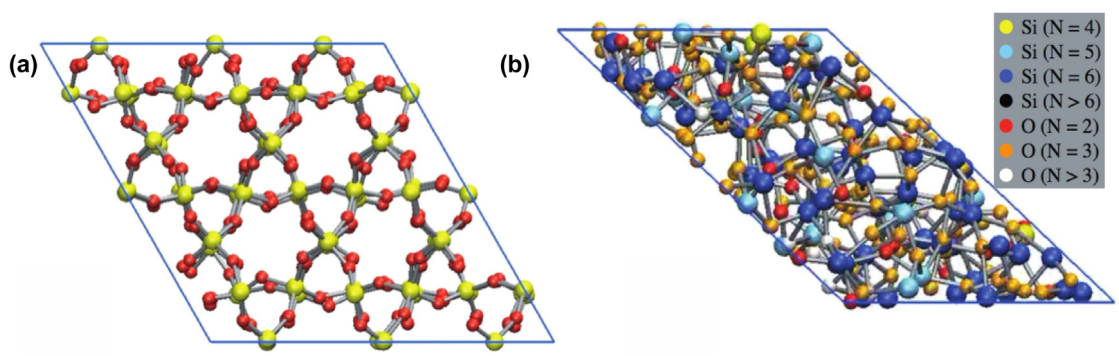

Fig. 3. Shock simulation of α-quartz: (a) initial configuration and (b) final configuration after 4 ps. Color represents the coordination number as shown by the bar in (b) . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and computational techniques for QMD simulations that better adapt to the new architectures [10]. One example is algebraic transformation of nonlocal pseudopotential computations to make them amenable to computationally-efficient level-3 basic linear-algebra software (BLAS3), *i.e.*, *via* BLASification, thereby achieving high floating-point performance [12]. QXMD continues to serve as a platform for such new technological developments.

QXMD has also been used to study a wide range of quantum-dynamical processes on high-end supercomputers, thereby providing new scientific knowledge. Examples include (1) emergence of thermodynamically-stable interphases in ceramic interfaces [22], (2) on-demand hydrogen production from water using metallic nanoparticles [24], (3) emergent magnetism from solely non-magnetic elements in two-dimensional layered materials [27], (4) dynamics of strongly-bound [23] and free [25] electron–hole pairs, and (5) ultrafast optical control of lattice motions [26,28], just to name a few.

Finally, as mentioned in Section 1, QXMD has been used extensively as an educational platform. We have developed hands-on tutorials as part of QXMD, and as of November 2018, 108 users from 55 institutions have been trained using QXMD in the MAGICS software workshops.

5. Conclusions

To date, QXMD has been used for method and algorithmic developments, basic scientific research, and training on first-principles materials simulations, on machines of all sizes, from desktop to high-end parallel computing platforms. Importantly, QXMD also serves as a testbed for hardware–software co-development on emerging computing architectures. For example, to facilitate easy access for development and collaboration, we have ported our code to Code Ocean. QXMD is also one of the ten initial simulation codes that will run on the United States' first exaflop/s computer (which can perform 10^{18} floating-point operations per second) under the US Department of Energy, Aurora early science program [9]. On these platforms, we are developing new techniques to: (1) make QXMD scalable for the unprecedented level of concurrency using globally-scalable and locally-fast (GSLF) solvers [11,12], which combine scalable tree-based algorithms for internode computations and fast BLAS3- and FFT-based solvers for intranode computations; and (2) reduce the time-to-solution using mixed-precision tensor operations by BLASifying key computational kernels. These developments will be reported in future publications.

Declaration of competing interest

The authors confirm that there is no conflict of interest associated with this publication.

Acknowledgments

This work was supported as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy,

Office of Science, Basic Energy Sciences, under Award Number DE-SC0014607.

References

[1] Car R, Parrinello M. Unified approach for molecular dynamics and density-functional theory. Phys Rev Lett 1985;55(22):2471–4.
[2] Payne MC, et al. Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients. Rev Modern Phys 1992;64(4):1045–97.
[3] Botu V, Ramprasad R. Adaptive machine learning framework to accelerate ab initio molecular dynamics. Int J Quantum Chem 2015;115(16):1074–83.
[4] Hohenberg P, Kohn W. Inhomogeneous electron gas. Phys Rev 1964;136(3B):B864–71.
[5] Kohn W, Sham LJ. Self-consistent equations including exchange and correlation effects. Phys Rev 1965;140(4A):A1133–8.
[6] Craig CF, Duncan WR, Prezhdo OV. Trajectory surface hopping in the time-dependent Kohn–Sham approach for electron-nuclear dynamics. Phys Rev Lett 2005;95(16). 163001.
[7] Shimojo F, et al. Large nonadiabatic quantum molecular dynamics simulations on parallel computers. Comput Phys Comm 2013;184(1):1–8.
[8] Meng S, Kaxiras E. Real-time, local basis-set implementation of time-dependent density functional theory for excited state dynamics simulations. J Chem Phys 2008;129(5). 054110.
[9] Service RF. Design for U.S. exascale computer takes shape. Science 2018;359(6376):617–8.
[10] Romero NA, et al. Quantum molecular dynamics in the post-petaflops era. IEEE Comput. 2015;48(11):33–41.
[11] Shimojo F, et al. A divide-conquer-recombine algorithmic paradigm for large spatiotemporal quantum molecular dynamics simulations. J Chem Phys 2014;140(18):18A529.
[12] Nomura K, et al. Metascalable quantum molecular dynamics simulations of hydrogen-on-demand. In: Proceedings of Supercomputing, SC14. IEEE; 2014, p. 661–73.
[13] Martin RM. Electronic Structure: Basic Theory and Practical Methods. Cambridge, UK: Cambridge University Press; 2008.
[14] Casida ME, Huix-Rotllant M. Many-body perturbation theory (MBPT) and time-dependent density-functional theory (TD-DFT): mbpt insights about what is missing in, and corrections to, the TD-DFT adiabatic approximation. Top. Curr. Chem. 2016;368:1–60.
[15] Dreuw A, Weisman JL, Head-Gordon M. Long-range charge-transfer excited states in time-dependent density functional theory require non-local exchange. J Chem Phys 2003;119(6):2943–6.
[16] Reed EJ, Fried LE, Joannopoulos JD. A method for tractable dynamical studies of single and double shock compression. Phys Rev Lett 2003;90(23). 235503.
[17] Umari P, Pasquarello A. Ab initio molecular dynamics in a finite homogeneous electric field. Phys Rev Lett 2002;89(15). 157602.
[18] Souza I, Íñiguez J, Vanderbilt D. First-principles approach to insulators in finite electric fields. Phys Rev Lett 2002;89(11). 117602.
[19] Shimojo F, et al. Embedded divide-and-conquer algorithm on hierarchical real-space grids: parallel molecular dynamics simulation based on linear-scaling density functional theory. Comput Phys Comm 2005;167(3):151–64.
[20] Yang W. Direct calculation of electron density in density-functional theory. Phys Rev Lett 1991;66(11):1438–41.
[21] Hoshi T, et al. Extremely scalable algorithm for 10^8 -atom quantum material simulation on the full system of the K computer. In: Proceedings of Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems, ScalA16. IEEE; 2016, p. 33–40.
[22] Shimamura K, et al. Bonding and structure of ceramic-ceramic interfaces. Phys Rev Lett 2013;111(6). 066103.
[23] Mou W, et al. Nanoscopic mechanisms of singlet fission in amorphous molecular solid. Appl Phys Lett 2013;102(17). 173301.
[24] Shimamura K, et al. Hydrogen-on-demand using metallic alloy nanoparticles in water. Nano Lett 2014;14(7):4090–6.
[25] Hakamata T, et al. The nature of free-carrier transport in organometal halide perovskites. Sci Rep 2016;5:19599.
[26] Lin MF, et al. Ultrafast non-radiative dynamics of atomically thin $MoSe_2$. Nature Commun 2017;8:1745.
[27] Kochat V, et al. Re doping in 2d transition metal dichalcogenides as a new route to tailor structural phases and induced magnetism. Adv Mater 2017;29(43). 1703754.
[28] Tung I, et al. Anisotropic structural dynamics of monolayer crystals revealed by femtosecond surface x-ray scattering. Nat. Photonics 2019. (in press).
[29] Shimojo F, et al. First-principles molecular-dynamics simulation of expanded liquid rubidium. Phys Rev B 1995;52(13):9320–9.
[30] Gropp W, Lusk E, Skjellum A. Using MPI. Third ed.. Cambridge, MA: MIT Press; 2014.
[31] Frigo M, Johnson SG. The design and implementation of FFTW3. Proc. IEEE 2005;93(2):216–31.
[32] Kleinman L, Bylander DM. Efficacious form for model pseudopotentials. Phys Rev Lett 1982;48(20):1425–8.
[33] Louie SG, Froyen S, Cohen ML. Nonlinear ionic pseudopotentials in spin-density-functional calculations. Phys Rev B 1982;26(4):1738–42.
[34] Perdew JP, Burke K, Ernzerhof M. Generalized gradient approximation made simple. Phys Rev Lett 1996;77(18):3865–8.
[35] Grimme S. Accurate description of van der waals complexes by density functional theory including empirical corrections. J Comput Chem 2004;25(12):1463–73.
[36] Tully JC. Molecular dynamics with electronic transitions. J Chem Phys 1990;93(2):1061–71.
[37] Bassman L, et al. Electronic origin of optically-induced sub-picosecond lattice dynamics in $MoSe_2$ monolayer. Nano Lett 2018;18(8):4653–8.
[38] Shimamura K, et al. A crossover in anisotropic nanomechanochemistry of van der Waals crystals. Appl Phys Lett 2015;107(23). 231903.
[39] Shimamura K, et al. Crystalline anisotropy of shock-induced phenomena: Omni-directional multiscale shock technique. Appl Phys Lett 2016;108(7). 071901.