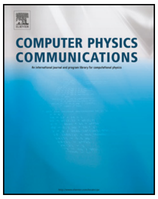
Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc



GenEvaPa: A generic evaporation package for modeling evaporation in molecular dynamics simulations ☆,☆☆

Bradley Harris [a], Gang-yu Liu [b], Roland Faller [a,*]

[a] *Department of Chemical Engineering, University of California, Davis, CA, 95616, United States*
[b] *Department of Chemistry, University of California, Davis, CA, 95616, United States*

ARTICLE INFO

Article history:
Received 25 April 2022
Received in revised form 2 August 2022
Accepted 5 September 2022
Available online 13 September 2022

Keywords:
Evaporation
Python script
Molecular dynamics

ABSTRACT

This work presents a novel general tool for modeling the process of evaporation without the need for modifying existing software using Python. The tool was developed based on the MDAnalysis package, which is used to import a Molecular Dynamics trajectory. The tool then removes solvent molecules and outputs a new structure file to be used for further simulation and analysis. This process is designed to be iterated by using the resulting dynamic simulation trajectory as the input file. The evaporation is designed to randomly delete solvent molecules while preserving solvation shells around solutes. The evaporation rate can be controlled by the length of the MD simulations and the number of particles removed between dynamic simulations. Validity of the tool was tested extensively using the Gromacs suite. Advantages of this tool include its genericness, simplicity and user friendliness, as no significant modification of existing software platform or Gromacs specific tools are needed.

Program summary
Program title: GenEvaPa
CPC Library link to program files: https://doi.org/10.17632/y5c3jnbjvs.1
Developer's repository link: github.com/bradsharris/GenEvaPa
Licensing provisions: GNU General Public License 3
Programming language: Python >3
Nature of problem: Approximate evaporation/drying processes in atomistic and coarse-grained molecular dynamics simulations while maintaining solvation shells around solute of interest. Forced drying in this manner allows for the study of a range of concentrations and self-assembly interactions.
Solution method: A python wrapper for existing molecular dynamics codes that randomly selects solvent for removal relative to a distance criteria around a solute to maintain solvation shells. Removal on final structure files maintains generic applicability for MD source codes and enables incorporation into automated loops to study longer drying.

1. Introduction

Evaporation impacts the outcomes of material assembly processes, chemical reactions, crystallization and mineralization processes, physical phase transitions of matter, as well as engineering design and production. Because of the complexity and variety of evaporation processes, there are numerous ways to model the evaporation of solvents at all scales ranging from continuum to molecular [1–8]. Among them, modeling solvent evaporation and evaporation-driven solute molecular assembly are of fundamental importance towards understanding of the processes and pertaining outcomes, and guiding applications, such as 3D nanoprinting, ink jet printing, surface modification, bio-preservation etc [1,2]. Molecular simulation techniques such as Grand canonical Monte Carlo (GCMC) and Molecular Dynamics (MD) have been utilized to model evaporation and correspondingly molecular assembly [1–4,7]. Current GCMC approaches for assembly during evaporation typically use randomly distributed particles on lattice sites and use various models for the probability of possible particle motion directions at each site. The evaporation is handled by a Monte Carlo par-

☆ The review of this paper was arranged by Prof. N.S. Scott.
☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).
* Corresponding author.
E-mail address: rfaller@ucdavis.edu (R. Faller).

https://doi.org/10.1016/j.cpc.2022.108539
0010-4655/

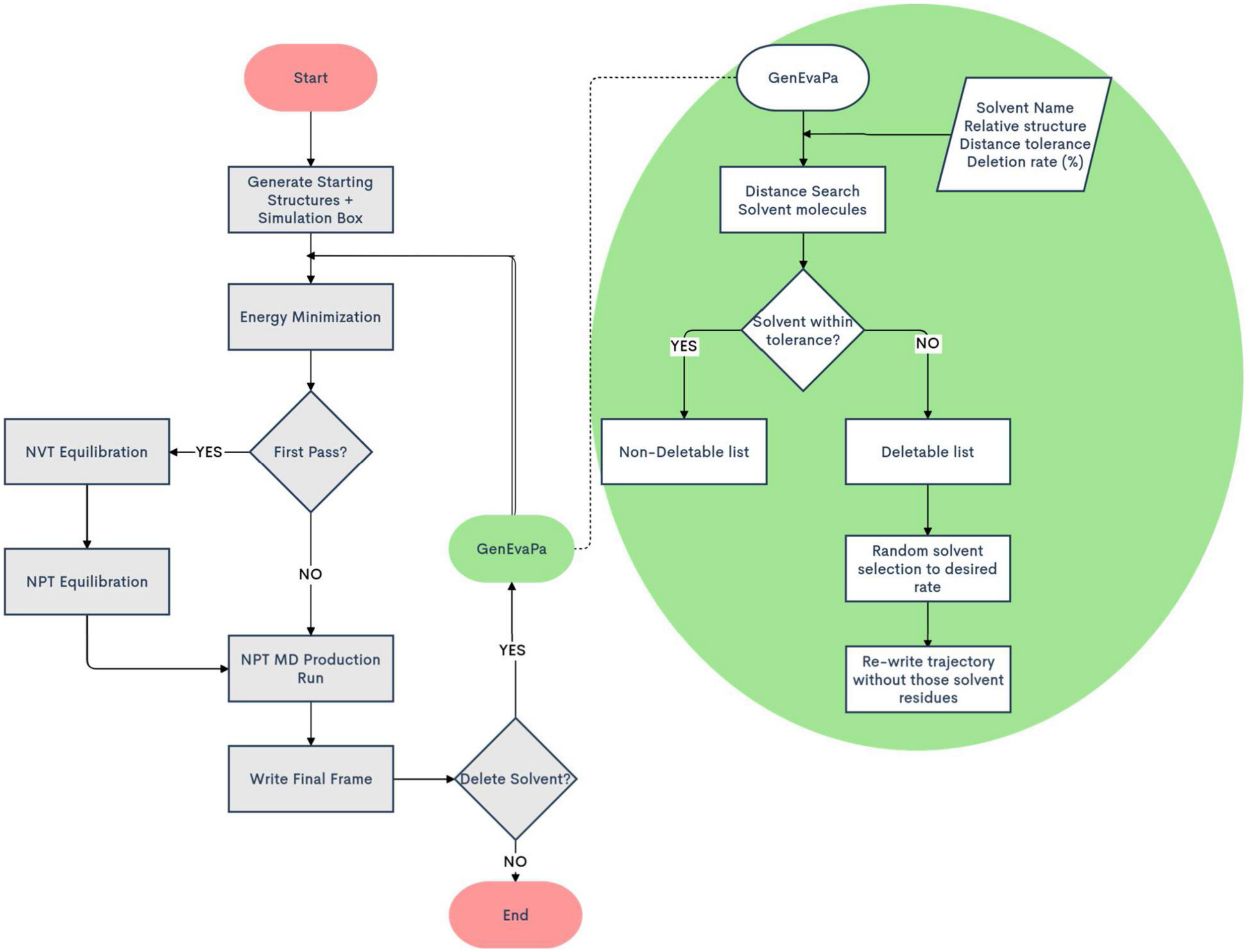


Fig. 1. Schematic of GenEvaPa showing evaporation script details and sample workflow. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

ticle removal move, which has an acceptance criterion based on chemical potential. GCMC techniques are effective for determining equilibrium properties due to their probabilistic nature and being rooted in statistical mechanics, but Monte Carlo techniques do not mimic the real dynamics of a system as the microstates/configurations are not appearing in time order. Molecular Dynamics (MD) is used to retain this dynamics information. Using MD, an existing approach for modeling evaporation processes relies on a diffusion-based model with a removal zone for solvent molecules that diffuse into the zone [2]. This approach is likely the most realistic way to model evaporation, as it replicates the flux driven nature of evaporation. The challenge is in the difficulty to implement generically as it is highly system dependent. Additionally, this approach is inherently limited by the diffusion rates of molecules, potentially limiting the length- and timescales that can be achieved. This same diffusion limitation also affects the ability to simulate late-stage evaporations with very small or no solvent remaining. To avoid the diffusion limitation, another MD approach was developed, i.e., randomly removing solvent molecules, e.g., using a bash script [1]. This approach benefits from being able to achieve a wide range of concentrations including the fully dried state and can be easily automated and sped up for any systems. The downside of this approach is that it does not account for hydration shells, which leads to unequal removal probabilities, and it was written specifically for their system and would require code adaptation.

To address these technical challenges, this work developed a tool to model evaporation with two primary goals: preservation of the hydration shells surrounding solute molecules and generic implementation. This led to implementation as a python wrapper to enable its use in both atomistic and coarse-grained simulations for a variety of existing MD codes, therefore making it system independent. Preserving hydration or solvation shells is scientifically

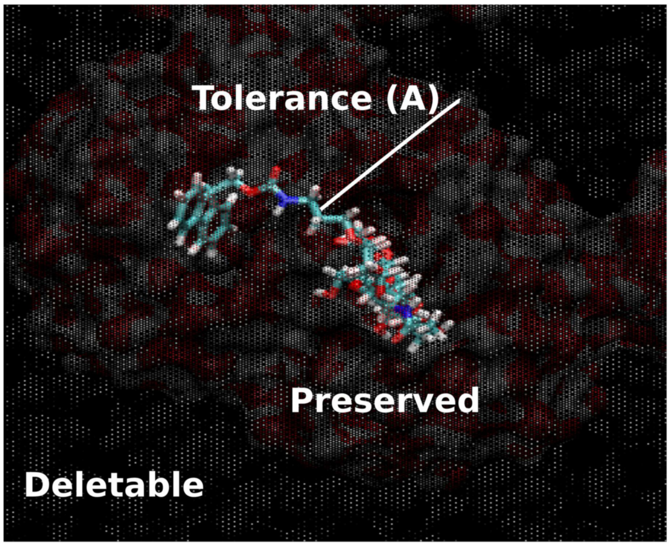


Fig. 2. Visualization of hydration shell preservation during distance search. Tolerance of 10 Angstrom from the evaporation of an aqueous solution of heparosan tetrasaccharide (MW = 1.099 kD) [14].

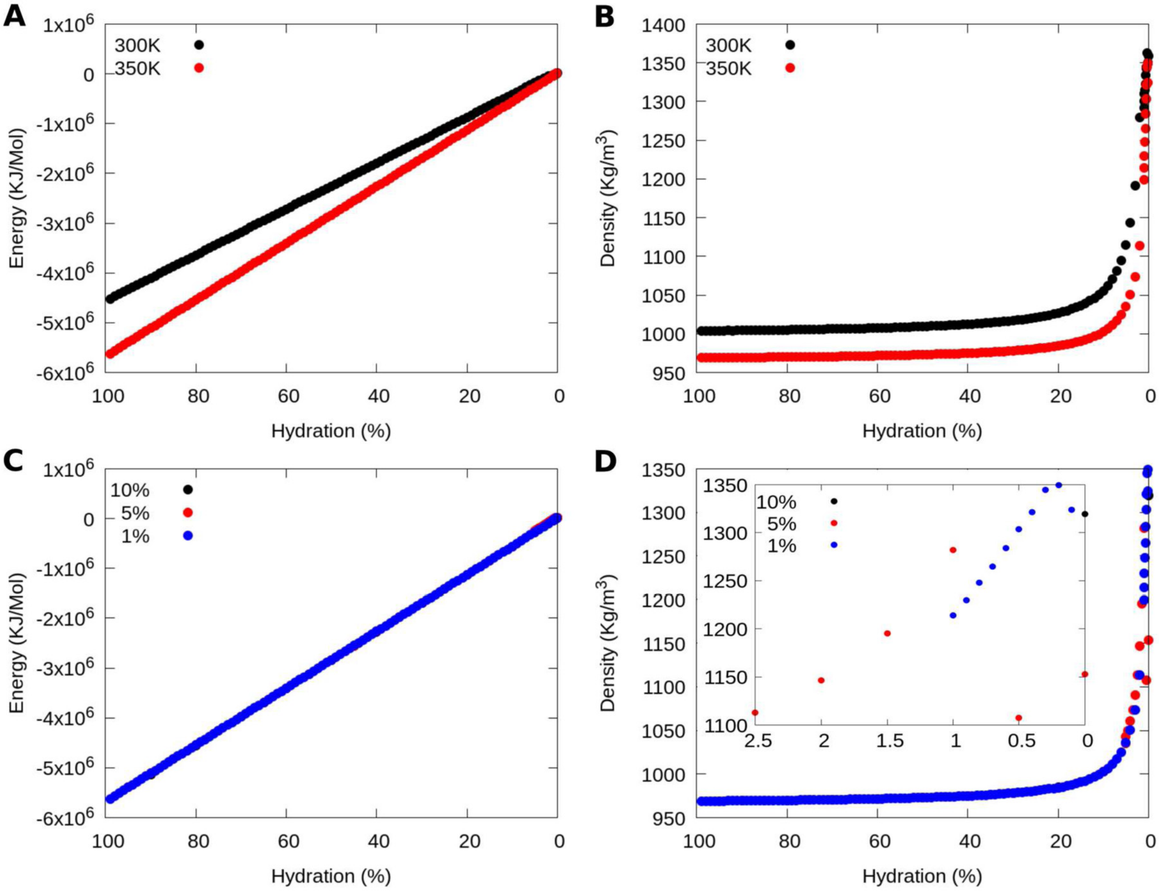


Fig. 3. Simulation performance characteristics: (A) At 1% deletion rate, the total energy is plotted as a function of dehydration at 300 and 350 K. (B) at 1% deletion rate, the density is displayed as a function of dehydration. (C) Total Energy at 350 K as a function of dehydration for different deletion rates. (D) Density against deletion rate as a function of dehydration, inset showing the dehydration of the final 2.5% water. Hydration levels shown as a percentage (%) relative to start.

significant as this more accurately models the physical process and corresponding hydrogen bonding. It also exhibits a technical advantage enabling further simulation to rapidly smooth out voids created by the removal of solvent molecules. This is done by creating those voids in places less involved in a hydrogen bonding network where solvent can diffuse more freely. The script has been tested and validated using Gromacs Versions 5.1.2016 and 2019.1 [9], thus should work for all simulation software packages that are supported by MDAnalysis [10,11].

2. Software description

GenEvaPa is an open-source python script that acts as a wrapper for MD simulation software and enables the deletion of solvent molecules outside of a certain tolerance distance from molecules of interest in order to preserve hydration shells. The script is built using the MDAnalysis python module for handling atom groups, making it compatible with any MD software suite that is compatible with MDAnalysis [11], e.g. Gromacs [9], LAMMPS [12], NAMD [13], CHARMM, etc.

2.1. Architecture

GenEvaPa is contained in a single python3 script that requires MDAnalysis [10,11] as the only external package. A flow chart of the internal workings of the software is shown Fig. 1. This schematic shows how the evaporation script fits into an MD workflow, with the operations of the script emphasized as a zoom in on the green oval. The script is written in an object-oriented manner with the main evaporation process implemented as a class that handles the preservation of hydration shells by calculating and comparing distances between specified solvent molecules and molecules of interest. A representative visualization of the hydration shell preservation for an atomistic system is shown in Fig. 2. It sorts the solvent molecules into deleteable and non-deleteable categories based on distance from a solute and uses the python *random.sample* function to delete the desired number of solvent molecules. The distance calculation is sped up by dynamically binning the simulation box into subsections that are equal to the tolerance of the hydration shell. Periodic boundary conditions are accounted for in this binning process. The sample workflow shown in Fig. 1 demonstrates how the script can be incorporated into a wider workflow. The process begins with initial structure and box generation followed by a series of equilibration steps and followed by a molecular dynamics NPT production simulation. To study the effect of solvent removal the final frame of the production run will be taken as the initial input for GenEvaPa. This file, along with the solvent name, structure of interest, tolerance (in Angstrom), and number of solvents to remove each step are provided, and the deletion is performed. GenEvaPa performs its calculations and outputs the final coordinates without the deleted solvents, which can then be used as the input structure file for the next MD run or other purposes. This process can then be looped and automated through a bash or python script to do complete controlled solvent deletion. An example use case is shown below, and a more detailed description of the classes and methods as well as an example automated workflow and analysis scripts are available at https://github.com/bradsharris/GenEvaPa.

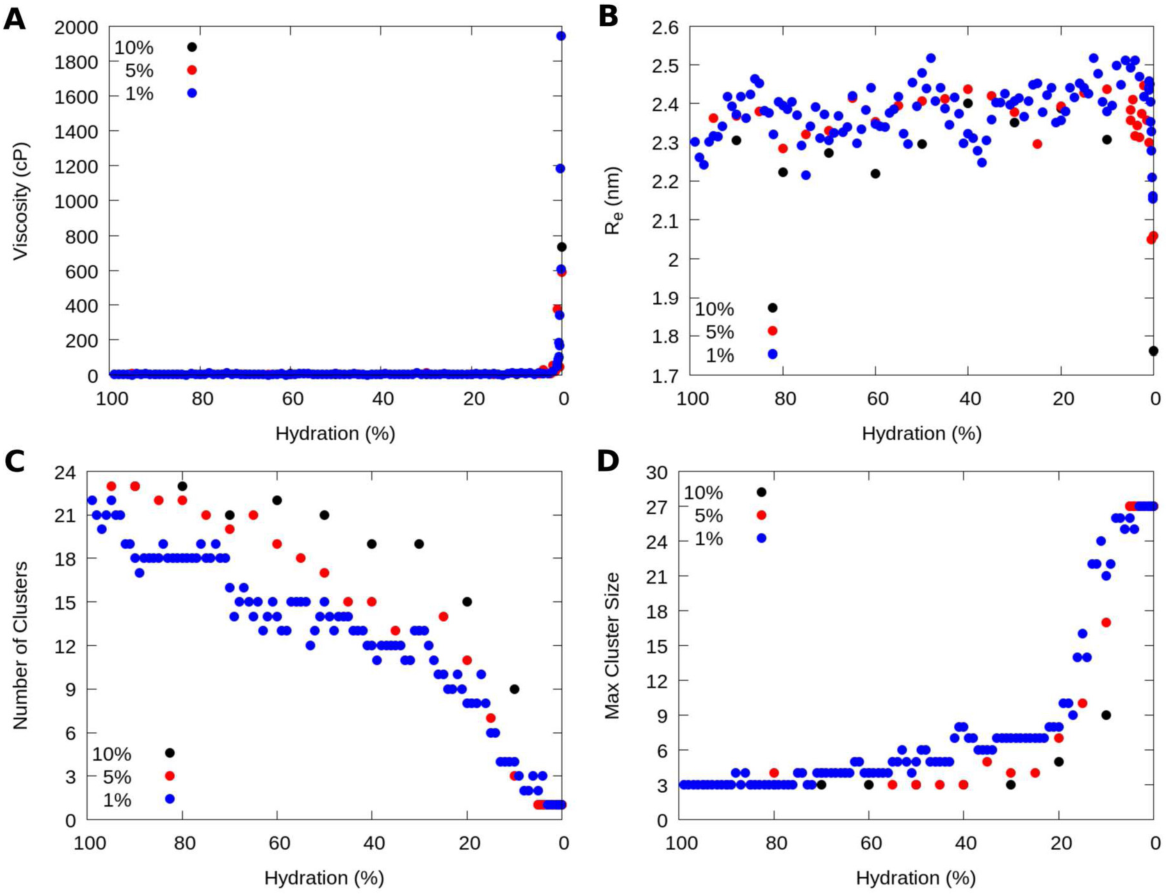


Fig. 4. Properties of the assembled heparosan tetrasaccharide molecules versus evaporation driven dehydration, simulated at all three water-deletion rates. (A) Viscosity vs dehydration. (B) End-to-End distance vs dehydration. (C) Number of clusters vs dehydration. (D) Maximum cluster size vs dehydration. Hydration shown as a percentage (%) relative to start.

3. Illustrative examples

To assess the performance of GenEvaPa several evaporations were performed on an atomistic system using Gromacs 2019.1. These simulations were designed to test the performance of the system at various evaporation rates as well as temperatures. Further uses of the script can be seen in our published work on controlled assembly of small molecules such as heparosan tetrasaccharide [14]. The starting configurations were identical to the pre-deletion equilibrated structures from that paper. Briefly, 27 heparosan tetra-saccharide molecules were added to a simulation box of size 17.19 nm x 17.19 nm x 17.19 nm that contained 162623 water molecules and 54 Na^+ counter ions for a total of 491514 atoms. The system was energy minimized and equilibrated in NVT for 100 ps using a velocity rescale thermostat, and then in NPT for 100 ps using a velocity rescale thermostat [15] and isotropic Parrinello-Rahman barostat [16]. The system was then simulated using NPT at 300 K and 350 K for 20 ns using the same thermostat and barostat to establish a baseline prior to solvent removal. Solvent deletion was performed using the GenEvaPa script following the workflow shown in Fig. 1, and automated to perform deletions of various percentages of initial water in every loop. Deletions performed at 350 K had rates of 1%, 5%, and 10% initial water removed per loop in order to evaluate the stability of the evaporation as a function of deletion rate. The 1% deletion was also performed at 300 K and compared to 350 K to verify consistent performance at different temperatures. Each MD run following deletion during the loop was performed in Gromacs 2019.1 [9], each MD ran for 1 ns NPT simulation time, with the tolerance set to 10 Å for the first 90% of water. The tolerance was then set to 0 for the final 20 ns due to not having enough water molecules that met the original tolerance criteria to be deleted. Chosen simulation performance characteristics of energy and density are shown in Fig. 3. The results show that, for these systems tested, the simulations reach equilibrated and consistent values rapidly for both temperatures and at all three deletion rates. The difference in energy and density profiles for temperature data in Fig. 3A and 3B are consistent with the change in temperature, e.g., decreasing the temperature from 350 to 300 K led to a less dense system, and increases in absolute energy. The data in Fig. 3C and 3D suggest that the system was stable at all three deletion rates tested, as the characteristic system parameters exhibited little rate dependence.

Several properties of the heparosan tetrasaccharide were evaluated to ensure convergence, as illustrated in Fig. 4. These results show that for each deletion rate the properties converge, with minor differences in the amount of interaction time available due to the solute molecules being simulated for varying total simulation times across rates, e.g., 110 ns for 1% and 22 ns for 5%. Overall, the convergence of these properties suggests that this method enables simulation of a wide range of evaporation rate, thus could accommodate various processes involving evaporation.

4. Conclusions

A novel methodology for modeling evaporation in molecular dynamics simulation was developed and implemented as a generic python wrapper. The script has been validated for Gromacs for various temperatures and deletion rates. The key scientific advances include the preservation of hydration shells and the applicability to both atomistic and coarse-grained systems. Technical advantages include its genericness and support for a variety of existing molecular dynamics software packages. This script has been tested for solute systems from small molecules to macromolecules, and in principle, widely supports any molecular dynamics system requiring evaporation or concentration dependence. Ongoing work is being conducted to validate the script for coarse-grained and mixed solvent systems. We anticipate its adoption and applications in many important processes involving evaporation such as chemical reactions, crystallization and mineralization processes, physical phase transitions of matter, as well as engineering design and production.

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.

Data availability

github. https://github.com/bradsharris/GenEvaPa.

Acknowledgements

This work is supported by the National Science Foundation (CHE-1808829 and BSH was partially supported by LLNL's LDRD program, under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory.

References

[1] R. Alessandri, J.J. Uusitalo, A.H. de Vries, R.W.A. Havenith, S.J. Marrink, J. Am. Chem. Soc. 139 (2017) 3697–3705, https://doi.org/10.1021/jacs.6b11717.

[2] S. Cheng, G.S. Grest, J. Chem. Phys. 183 (2013) 064701, https://doi.org/10.1063/1.4789807.

[3] A. Crivoi, F. Duan 4 (2014) 4310, https://doi.org/10.1038/srep04310.

[4] E.S. Landry, S. Mikkilineni, M. Paharia, A.J.H. McGaughey, J. Appl. Phys. 102 (2007) 124301, https://doi.org/10.1063/1.2821753.

[5] R. Ledesma-Aguilar, D. Vella, J.M. Yeomans, Soft Matter 10 (2014) 8267–8275, https://doi.org/10.1039/C4SM01291G.

[6] J. Schlottke, B. Weigand, J. Comput. Phys. 227 (2008) 5215–5237, https://doi.org/10.1016/j.jcp.2008.01.042.

[7] K. Yamashita, K. Kashiwagi, A. Agrawal, H. Daiguji, Mol. Phys. 115 (2017) 328–342, https://doi.org/10.1080/00268976.2016.1262555.

[8] V.V. Zhakhovskii, S.I. Anisimov, J. Exp. Theor. Phys. 84 (1997) 734–745, https://doi.org/10.1134/1.558192.

[9] M.J. Abraham, T. Murtola, R. Schulz, S. Pall, J.C. Smith, B. Hess, E. Lindahl, SoftwareX 1–2 (2015) 19–25, https://doi.org/10.1016/j.softx.2015.06.001.

[10] R.J. Gowers, L. Max, B. Jonathan, T.J.E. Reddy, M.N. Melo, S.L. Seyler, D. Jan, D.L. Dotson, S.K. Buchoux, I.M. Kenney, O. Beckstein, N. p., United States, Web, https://doi.org/10.25080/Majora-629e541a-00e, 2019.

[11] N. Michaud-Agrawal, E.J. Denning, T.B. Woolf, O. Beckstein, J. Comput. Chem. 32 (2011) 2319–2327, https://doi.org/10.1002/jcc.21787.

[12] S. Plimpton, J. Comput. Phys. 117 (1995) 1–19, https://doi.org/10.1006/jcph.1995.1039.

[13] J.C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R.D. Skeel, L. Kale, K. Schulten, J. Chem. Phys. 153 (2020) 044130, https://doi.org/10.1002/jcc.20289.

[14] J. Zhang, H. Yu, B. Harris, Y. Zheng, U. Celik, L. Na, R. Faller, X. Chen, D.R. Haudenschild, G.Y. Liu, J. Phys. Chem. C 124 (2020) 6405–6412, https://doi.org/10.1021/acs.jpcc.9b11377.

[15] G. Bussi, D. Donadio, M. Parrinello, J. Chem. Phys. 126 (2007) 014101, https://doi.org/10.1063/1.2408420.

[16] M. Parrinello, A. Rahman, J. Appl. Phys. 52 (1981) 7182–7190, https://doi.org/10.1063/1.328693.