SoftwareX 9 (2019) 20–27

Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

GOMC: GPU Optimized Monte Carlo for the simulation of phase equilibria and physical properties of complex fluids

Younes Nejahi [b], Mohammad Soroush Barhaghi [a], Jason Mick [a], Brock Jackman [b], Kamel Rushaidat [b], Yuanzhe Li [b], Loren Schwiebert [b], Jeffrey Potoff [a,*]

[a] Department of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI 48202, USA
[b] Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

ARTICLE INFO

Article history:
Received 21 June 2018
Received in revised form 15 November 2018
Accepted 21 November 2018

Keywords:
Molecular simulation
Monte Carlo
Gibbs ensemble
Adsorption
Phase equilibrium
GPU

ABSTRACT

GPU Optimized Monte Carlo (GOMC) is open-source software for simulating molecular systems using the Metropolis Monte Carlo algorithm. It supports simulations in a variety of ensembles, which include canonical, isothermal–isobaric, grand canonical, and Gibbs ensemble. GOMC can be used to study vapor–liquid equilibria, adsorption in porous materials, surfactant self-assembly, and condensed phase structure for complex molecules. GOMC supports a variety of all-atom, united atom, and coarse grained force fields such as OPLS, TraPPE, Mie, and Martini. The software has been written in object oriented C++, and uses OpenMP and NVIDIA CUDA to allow for execution on multi-core CPU and GPU architectures.

Code metadata

Current code version	*V2.31*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_93
Legal Code License	APGL 3.0
Code versioning system used	Git
Software code languages, tools, and services used	C, C++, OpenMP, CUDA, cmake
Compilation requirements, operating environments & dependencies	ANSI C89 and C++11, Unix, Linux, Windows
If available Link to developer documentation/manual	http://gomc.eng.wayne.edu/manual/
Support email for questions	gomc@eng.wayne.edu

1. Motivation and significance

Given a molecular structure, and an accurate description of interactions between atoms, or groups of atoms, computer simulations may be used to calculate nearly any physical property associated with a specified molecule. Using simulations, it is possible to determine relationships between atomic-level interactions, nanoscale structure, and macroscopic properties. Hence, computer simulations are able to provide insight that may otherwise be impossible to obtain from experiments. Notable examples include self-assembly of nanoscale structures [1–4], understanding biological structure-function relationships [5–7], drug design [8–10], and materials design for separation and storage of gases [11–16]. Advances in the use of molecular simulation for materials design have been driven by simultaneous advances in force fields [17–25], algorithms [26–31], computer hardware, and software designed to take advantage of parallel computer architectures, such as multi-core CPUs, and graphics processing units (GPUs) [32–39].

Molecular mechanics simulations may be divided into two categories: molecular dynamics, where the system evolves according to Newton's equations of motion, and Monte Carlo, where the system evolves through trial moves accepted according to probabilities defined by statistical mechanics. Molecular dynamics simulations have been adopted widely by the research community due to their ability simulate nearly any molecule, and the widespread availability of high-quality free, or nearly free, software.

Examples of commonly used molecular dynamics codes include NAMD [33], AMBER [40], CHARMM [39,41], GROMACS [38],

* Corresponding author.
E-mail address: jpotoff@wayne.edu (J. Potoff).

https://doi.org/10.1016/j.softx.2018.11.005

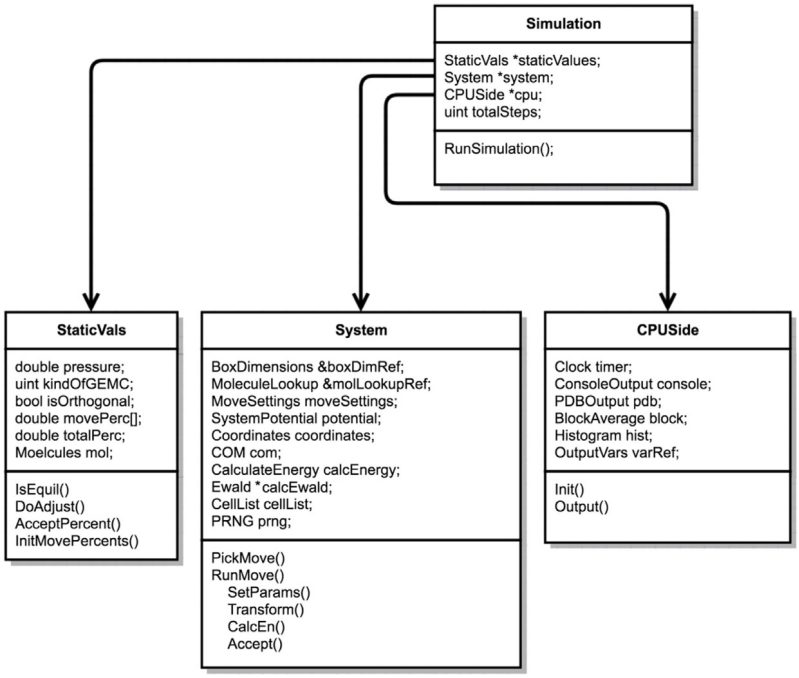

Fig. 1. Schematic of GOMC software architecture.

LAMMPS [34,42] and HOOMD [37,43]. The molecular dynamics algorithm is inherently parallelizable, and extensive efforts have been focused on the optimization of these codes for high performance on large numbers of CPU cores and GPUs.

Unlike molecular dynamics codes, where many have been developed by large consortiums of researchers, Monte Carlo codes have typically been developed on an ad-hoc basis within individual research groups to solve specialized problems. Some of these codes were made available to the research community [44–47], with varying levels of documentation. Others were restricted to use within a research group, or were commercial software [48]. This code development model presents a serious challenge for ensuring data reproducibility [49].

Recognizing the need for a publicly available general-purpose Monte Carlo simulation engine, Marcus Martin published Towhee in 2000, the first true open-source code for the simulation of phase equilibria and condensed phase structure of complex molecular systems [53]. While having a broad feature set, Towhee runs on a single CPU core, and has mediocre computational performance [54]. In recent years, a number of other open-source codes have become available, each with varying features, including HOOMD [55], Etomica [56], FEASST [57], MS2 [47,58,59], RASPA [60], Cassandra [61], and GOMC (this work). RASPA was designed primarily for simulation of adsorption in porous materials, and like Towhee, runs on a single CPU core. Cassandra and GOMC are focused broadly on the simulation of phase equilibria and adsorption, and both are capable of parallel computation on multi-core CPUs. HOOMD supports simulations of hard polyhedrons. GOMC and HOOMD have the ability to use graphics processing units (GPUs) for the calculation of non-bonded interactions, leading to significant performance gains for some systems.

2. Software functionalities

With GOMC, users are able to perform simulations for complex molecules in a variety of ensembles, which include canonical (NVT), Isobaric–Isothermal (NPT), grand canonical (μVT), and NVT and NPT variants of the Gibbs ensemble algorithm [28]. GOMC has support for the following Monte Carlo moves: rigid-body displacement, rigid-body rotation, volume exchange (both isotropic and anisotropic), insertion/deletion within and between boxes, and regrowth. GOMC supports simulations in non-orthorhombic boxes. To enhance the rate of acceptance for molecule swap and sampling molecular configurations, the coupled–decoupled configurational-bias algorithm is used [62]. GOMC supports a variety of force fields, including TraPPE [62–69], Mie [20,70–74], OPLS [75] and Martini [76], which use a Lenard-Jones, or Mie, potentials plus point charges to represent intermolecular interactions, and a combination of harmonic and cosine functions to account for bonded interactions. Support is included for scaling of both 1–4 van der Waals and electrostatic interactions.

When performing Gibbs ensemble Monte Carlo simulations, the code produces the saturated liquid and vapor densities, vapor pressure and heat of vaporization at the temperature of interest. For mixtures, the code also produces the compositions of the corresponding vapor and liquid phases. In grand canonical Monte Carlo simulations, users can output histograms of the states sampled, enabling the use of histogram-reweighting methods to determine coexistence properties [77,78]. Using the FixedAtoms feature, it is possible to identify atoms/molecules that should not participate in various moves (insertion/deletion, displacement, etc.) during the simulation. This feature is especially helpful for simulations of gas adsorption in rigid porous materials. In all ensembles, GOMC outputs atomic coordinates in PDB format, while instantaneous and average values of various properties are written to a plain-text log file. GOMC uses file formats that are compatible with NAMD and VMD [33,79], enabling the reuse of as many existing software components as possible.

3. Software architecture

GOMC is licensed under AGPL 3.0, which allows for commercial, private, and patent use. The code is publicly available for download through our GitHub repository [80] and the GOMC website [81]. Users can register on the website to receive notifications about new releases and bug fixes. GOMC development follows the Git-

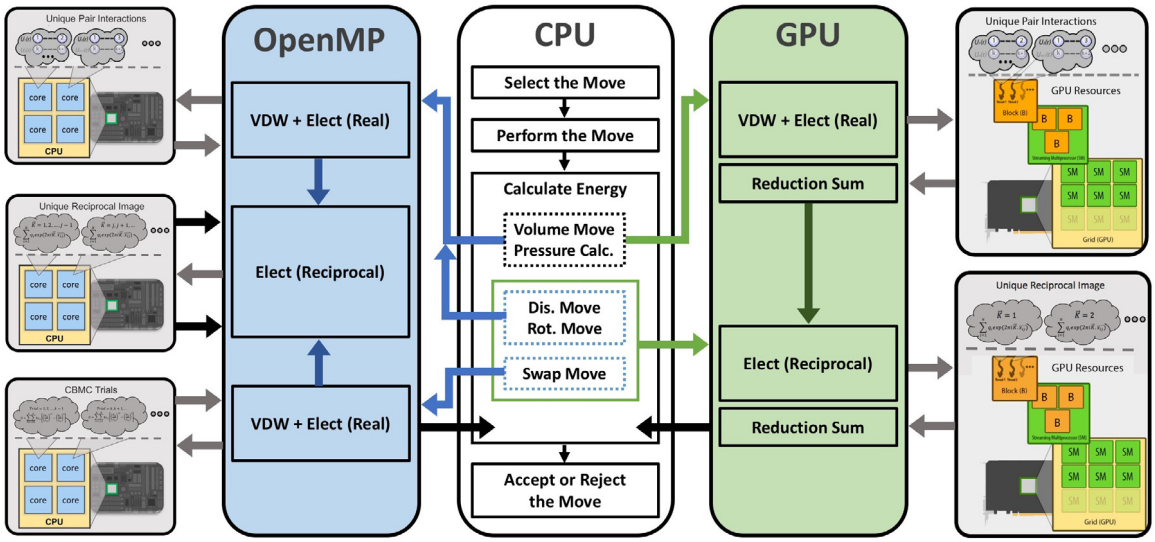

Fig. 2. Parallelization scheme used by GOMC.

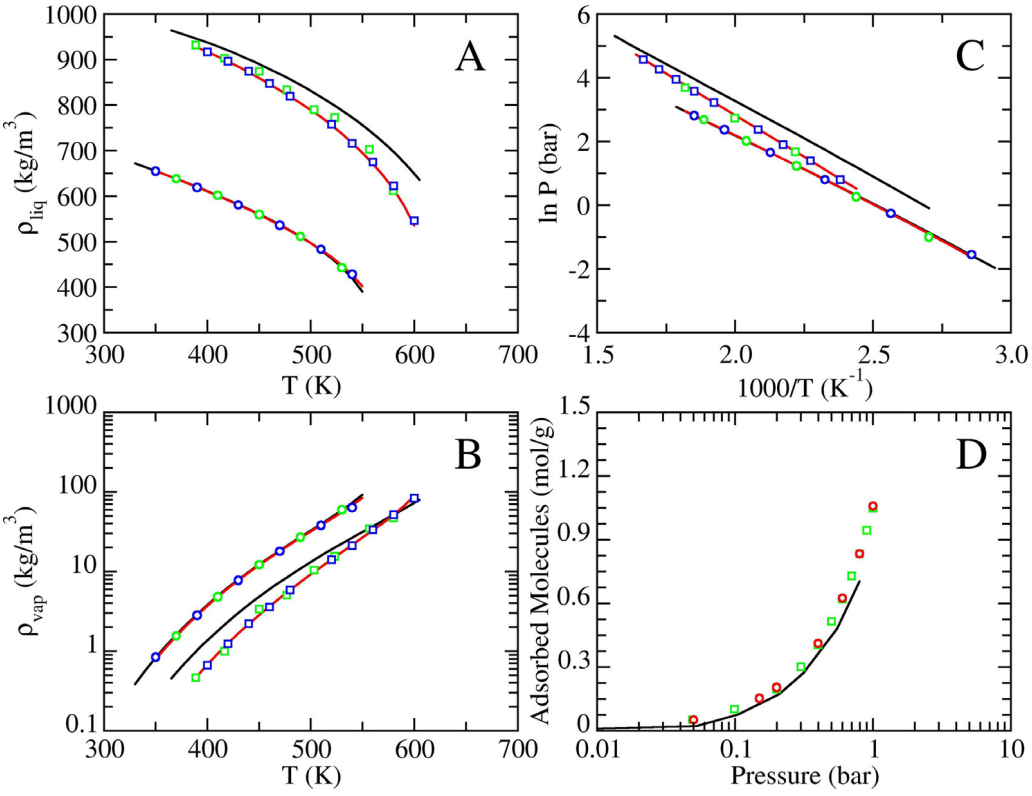

Fig. 3. (A, B) Saturated liquid and vapor densities for n-octane and SPC/E water, (C) Clausius–Clapeyron plots for n-octane and SPC/E water and (D) adsorption isotherm of CO_2 in IRMOF-1 predicted by GOMC. Experimental data (black lines), GOMC predictions (red and blue symbols), previous work (green symbols). (A, B, C): NIST Chemistry WebBook [50] (black lines), GCMC+histogram reweighting calculations (red lines), GEMC predictions for n-octane (blue circles), GEMC prediction for SPC/E water (blue squares), values obtained by Potoff et al. [20] (green circles), and values obtained by Boulougouris et al. [51] (green squares). (D): Experiment [52] (black line), predictions of GOMC (red circles), and calculations of Parkes et al. [52] (green squares). . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Flow branching model, uses CircleCI to test build each commit, and automated shell scripts are used to increase the reliability of regression testing. A wide range of test cases have been generated for this purpose [82], and a number of step by step tutorials are available [83].

3.1. Code architecture

GOMC has been written in object oriented C++, which offers a good balance between code development time, interoperability with existing software elements, and code performance. The software may be compiled as a single-threaded application, a

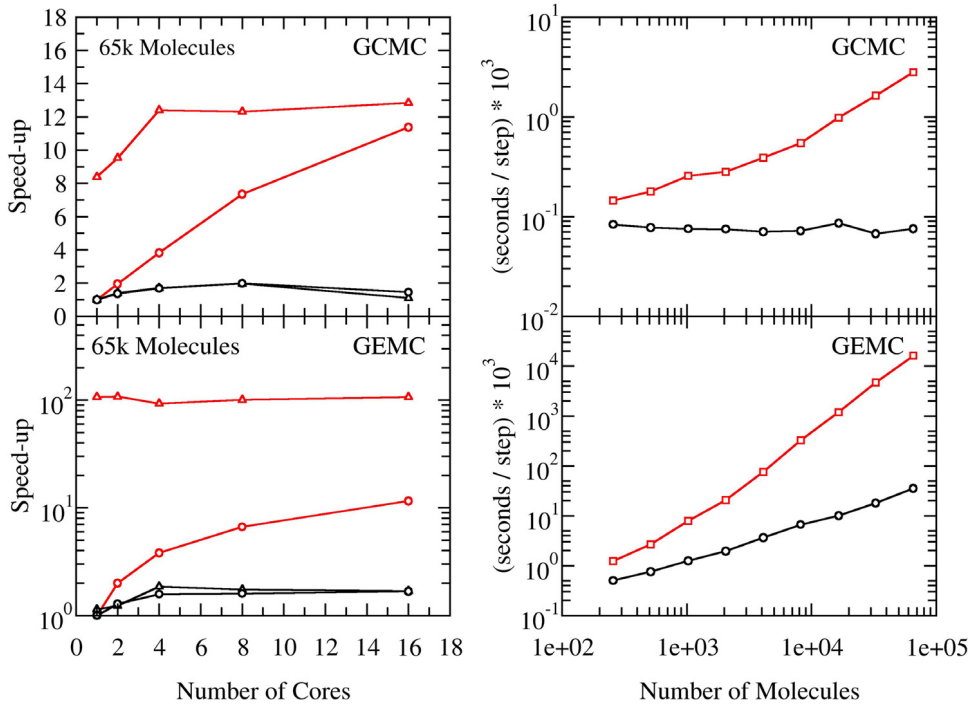

Fig. 4. (Top left) Speed-up for GCMC simulations in GOMC, relative to calculations performed on a single CPU core, for n-octane with OpenMP (black circle), n-octane with GPU+OpenMP (black triangle), water with OpenMP (red circle), and water with GPU+OpenMP (red triangles). (Bottom left) Speed-up for Gibbs ensemble Monte Carlo (GEMC) simulations in GOMC, relative to calculations performed on a single CPU core, for n-octane with OpenMP (black circle), n-octane with GPU+OpenMP (black triangle), water with OpenMP (red circle), and water with GPU+OpenMP (red triangle). (Top right) Execution time for single core GCMC simulations in GOMC for n-octane (black) and water (red). (Bottom right) Execution time for single core GEMC simulations in GOMC for n-octane (black) and water (red). . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

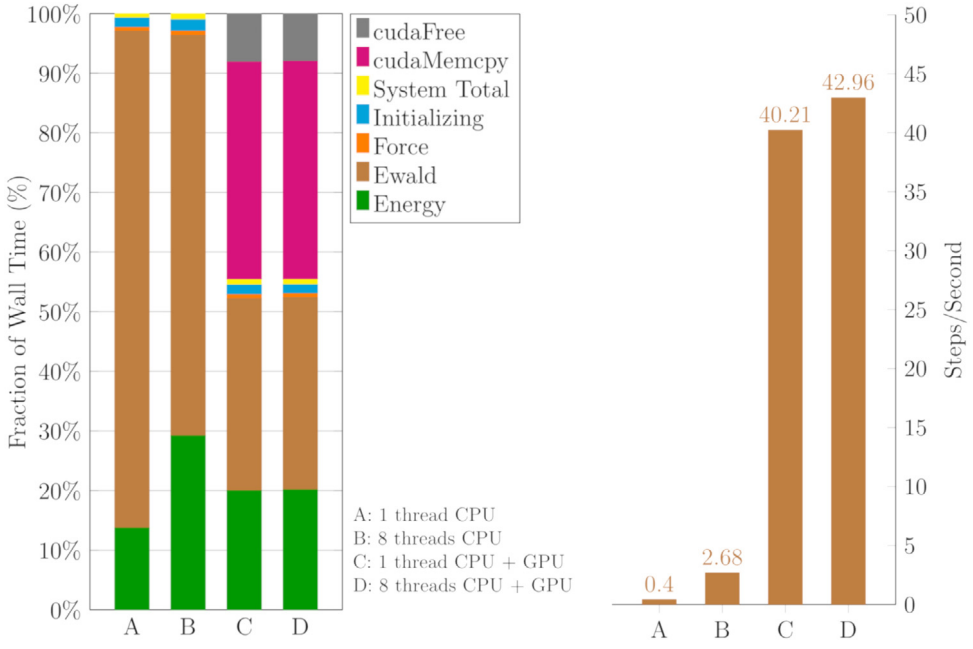

Fig. 5. Gibbs ensemble Monte Carlo simulation of SPC/E water. Left: Reciprocal part of Ewald summation is the most expensive calculation in GOMC. (B) OpenMP can reduce the wall-time (A) spends in this region. The GPU can further reduce the time required for this calculation. (C) and (D) shows the impact of OpenMP compared to that of the GPU. Right: The performance in GOMC significantly improved using a combination of the GPU and OpenMP. Although the impact of GPU is much larger than OpenMP.

multi-threaded application using OpenMP, or to use many-core heterogeneous CPU–GPU architectures using OpenMP and CUDA.

GOMC has a hierarchical architecture where each class is decomposed into subclasses, as shown in Fig. 1. The **main**() function creates a **Simulation** class that contains the Monte Carlo step number, handles output to files, timer and an instance of **System** class. **Simulation** also reads all of the input files (control, PDB and PSF) and initializes the **System** subclass. The **System** class holds more comprehensive information, including box dimensions, move settings, intermolecular potential type, coordinates of atoms, and their center of mass. **System** also includes the

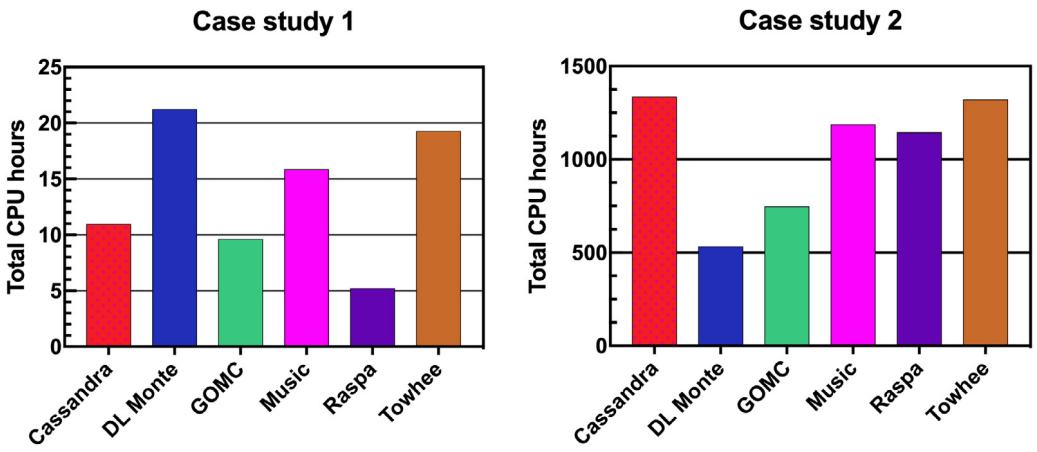

Fig. 6. Performance comparison of GOMC with other Monte Carlo codes [54]. The total time is calculated by running an entire adsorption isotherm of eight pressure points.

subclasses of **MoleculeLookup** to perform molecule operations, **CalculateEnergy** to calculate different types of energies, **Ewald** to perform the Ewald summation, **CellList** to utilize the cell list implementation, and **PRNG** to generate random numbers.

ChooseAndRunMove() in **System** runs a single step, using **PickMove**() to randomly select a move, and **RunMove**() to perform the move. **SetParams**() prepares each move and initializes parameters, **Transform**() generates the trial move, **CalcEn**() calculates the energy of the trial move, and **Accept**() determines whether the trial move should be accepted or not and makes the according changes.

3.2. Parallelization strategy

In GOMC, substantial improvements in computational performance have been achieved through the incorporation of a cell list, and caching components of the Ewald Sum [84,85], which significantly reduces the number of calculations that are required for each attempted move. GOMC makes extensive use of polymorphism to reduce branching within the code.

Computations in GOMC are parallelized by distributing pairwise interactions to CPU threads and/or GPU streams using a combination of OpenMP and NVIDIA CUDA. A schematic of how computations are parallelized in GOMC is shown in Fig. 2. For trial moves that involve a single molecule, van der Waals and real-space components of the electrostatic interactions are distributed among available CPU threads. For trial moves that require the recalculation of energies for the entire system, such as the volume move, all pairwise interactions are distributed to GPU streams. For all moves, the reciprocal space contribution to the electrostatic energy is computed on the GPU. If no GPU is available, reciprocal space contributions are computed on the CPU.

4. Illustrative examples

To demonstrate the accuracy of algorithms implemented in GOMC, calculations were performed to predict the coexistence curves for n-octane [20] and SPC/E [86] water using GCMC+histogram reweighting methods [77,78] and GEMC [28]. In Fig. 3-A, B, and C, liquid and vapor densities and vapor pressures determined from GEMC for n-octane and water are compared to the values predicted by the GCMC+histogram reweighting method, NIST Chemistry WebBook [50] and previous simulations [20,51]. For both GEMC and GCMC simulations, potentials were truncated at 10 Å, and analytical tail corrections were applied [87]. The coupled–decoupled configurational-bias Monte Carlo [62] (CD-CBMC) algorithm with 10 trials for the first atom, 8 trials for remaining atoms, 100 angle trials, and 50 dihedral trials was used to improve sampling efficiency. For polar systems, Coulombic interactions were calculated using the Ewald summation [88] with a tolerance of 1×10^{-5}. For GCMC simulations, a cubic box size of 15,625 Å^3 was used for water; while, a box size of 64,000 Å^3 was used for n-octane. GCMC simulations were equilibrated for 5×10^6 MC steps, followed by 4.5×10^7 production steps. GEMC calculations were performed on systems containing 1000 molecules, and each system was equilibrated for 2.5×10^7 MC steps, followed by 2.5×10^7 production steps.

In Fig. 3-D, a comparison is presented between results generated with GOMC and that of Yazaydin et al. [52] for the adsorption of CO_2 in IRMOF-1 at 298 K. The atomic coordinates for the IRMOF-1 atoms were taken from Eddaoudi et al. [89]. The TraPPE [66] and DREIDING [90] force fields were used for carbon dioxide and IRMOF-1, respectively. The simulation box was a $2\times2\times2$ orthogonal supercell ($\alpha = 90$, $\beta = 90$, $\gamma = 90$). Lennard-Jones potentials were truncated at 12.8 Å without an analytical tail correction. Excellent agreement was achieved between the predictions from GOMC and prior calculations.

5. Performance and scaling

To demonstrate the efficiency and performance of GOMC, grand canonical and Gibbs Ensemble Monte Carlo simulations were performed on n-octane and SPC/E water [91]. Simulations were performed for 256 to 65,536 molecules to demonstrate the scaling of GOMC. Calculations were performed on a system with a 10-core Intel Xeon E5-2680 v2 @ 2.80 GHz processor, 16 GB system memory, and an NVIDIA Tesla K40c running CentOS 6.8 Linux. The code was compiled with the Intel Compiler v16 (2016 initial version) and CUDA toolkit 8.0. The "–O3" compiler flag was used to optimize the performance of the CPU code. Potentials were truncated at 10 Å and analytical tail corrections were applied. The temperature was set to 400 K for n-octane and 300 K for SPC/E water. The distribution of moves for GCMC simulations were 20% displacement, 20% rotation (50% target acceptance), and 60% molecule swap move. For GEMC simulations the distribution of moves was 40% displacement, 40% rotation, 19% molecule swap, and 1.0% volume exchange.

In Fig. 4 the execution time for GCMC and GEMC simulations of n-octane and water are presented as a function of system size for computations performed on a single CPU core. Additional calculations were performed on systems containing 65,000 molecules (520,000 interaction sites) to evaluate the speed up achieved by executing the code on multicore CPUs and GPUs. For GCMC simulations of n-octane, the use of the cell list greatly reduces the number of pairwise interactions to be calculated. Because of the relatively low computational load, any gain in performance from the addition

of multicore processing is erased by communication overhead (thread startup and lock management) between cores. For GEMC simulations of n-octane, the computational effort increases with system size due to the inclusion of a volume exchange, which scales as $O(N^2)$. For a 65,000 molecule system, significant performance gains can be achieved by running on up to four CPU cores.

Code performance for simulations of SPC/E water scales very differently than n-octane because of the need to compute electrostatic interactions via the Ewald sum. GCMC simulations of water show a linear scaling of computational effort with system size, while for simulations of n-octane the computational effort was invariant. For a 65,000 molecule system, the use of up 8 CPU cores results in a speed-up of 7 times that of a single CPU core. Using additional cores reduces code execution time, but with lower parallel efficiency. Using 16 CPU cores results in the speed-up of 11 times that of a single CPU core. Execution on one CPU core plus an NVIDIA K40 GPU results in a factor of 8 speed-up compared to single CPU core, while using four CPU cores plus the GPU produces a speed up of 12. The main challenge of optimizing code execution time on the GPU is reducing the number of copies between GPU and CPU, since this communication becomes the bottleneck. Fig. 5 shows that cudaMemcpy() and cudaFree() consumes almost 45% of the execution time.

The application of the GPU to GEMC simulations of water results in significant performance increases. On the GPU, the $O(N^2)$ volume move can be flattened to O(N) [92]. Additionally, all reciprocal space k-vectors may be computed in parallel. Unlike CPU computation, on the GPU it is faster to recalculate all k-vectors for each move than store them in memory. For a system containing 65,000 water molecules, calculations performed on the GPU and one CPU core were approximately 100 times faster than that of a single CPU core; adding more CPU cores did not improve performance. For code running on multi-core CPUs, using 16 cores produced a speed up of approximately 10 times that of code executed on a single CPU core.

Fig. 5 demonstrates the anatomy of GOMC performance for 65,000 molecules of SPC/E water. The reciprocal part of Ewald summation is responsible for most of the computational cost of the calculation. The use of OpenMP (B) to parallelize the calculation of the reciprocal space term can reduce the wall-clock execution time. The GPU (C, D) can further reduce wall-clock execution time, improving the performance significantly. Even though GPUs add significant overhead due to the memory copy between host and GPU, sufficient workload created by reciprocal part of Ewald can hide this overhead.

In Fig. 6, the performance of GOMC is compared to other Monte Carlo codes for the simulation of absorption of CO_2 in IRMOF-1 [91]. The simulation protocol, and timing data for all simulations, except for GOMC, were taken from Gowers et al. [54]. Calculations were performed on one core of an Intel Xeon CPU E5-2620 v3 @ 2.40 GHz processor, which has identical single core performance as the Intel Xeon CPU E5-2630 v3 @ 2.4 GHz processor used by Gowers et al. Case study 1 used only Lennard-Jones interactions between CO_2 molecules and IRMOF-1. Case study 2 includes both Lennard-Jones and electrostatic interactions. The data show GOMC has single core performance that is competitive with other Monte Carlo codes. It should be noted that these test cases use a very large potential cut-off of 25.0 Å, which limits the effectiveness of the cell-list used in GOMC. Substantially better performance may be obtained for systems with a 10 or 14 Å cut-off.

6. Conclusions

GOMC software has competitive performance, with other codes, and has been optimized for high performance on single, multi, or many threaded architectures. Work is ongoing to add more advanced sampling algorithms, such as Continuous Fractional Component Monte Carlo [93–96], support for simulations of adsorption in flexible MOF, parallelization across multiple nodes via MPI, and support for polarizable force fields that use Drude oscillators.

Acknowledgments

Financial support from the National Science Foundation, USA (ACI-1148168 and OAC-1642406) and hardware donations from NVIDIA and Silicon Mechanics are gratefully acknowledged. Computations in this work were performed with resources from the Grid Computing initiative at Wayne State University.

Conflict of interest

The authors declare that there is no conflict of interest.

References

[1] Glotzer SC, Solomon MJ. Anisotropy of building blocks and their assembly into complex structures. Nat Mater 2007;6:557–62.
[2] van Anders G, Klotsa D, Karas AS, Dodd PM, Glotzer SC. Digital alchemy for materials design: Colloids and beyond. Acs Nano 2015;9:9542–53.
[3] Zhang ZL, Glotzer SC. Self-assembly of patchy particles. Nano Lett 2004;4:1407–13.
[4] LeBard DN, Levine BG, Mertmann P, Barr SA, Jusufi A, Sanders S, Klein ML, Panagiotopoulos AZ. Self-assembly of coarse-grained ionic surfactants accelerated by graphics processing units. Soft Matter 2012;8:2385–97.
[5] Zhang Y, Schulten K, Gruebele M, Bansal PS, Wilson D, Daly NL. Disulfide bridges: Bringing together frustrated structure in a bioactive peptide. Biophys J 2016;110:1744–52.
[6] Perilla JR, Schulten K. Physical properties of the HIV-1 capsid from all-atom molecular dynamics simulations. Nature Commun 2017;8:15959.
[7] Klauda JB, Kucerka N, Brooks BR, Pastor RW, Nagle JF. Simulation-based methods for interpreting X-ray data from lipid bilayers. Biophys J 2006;90:2796–807.
[8] Jorgensen WL. The many roles of computation in drug discovery. Science 2004;303:1813–8.
[9] Jorgensen WL. Efficient drug lead discovery and optimization. Accounts Chem Res 2009;42:724–33.
[10] Zhong S, Chen X, Zhu X, Dziegielewska B, Bachman KE, Ellenberger T, Ballin JD, Wilson GM, Tomkinson AE, MacKerell AD. Identification and validation of human DNA ligase inhibitors using computer-aided drug design. J Med Chem 2008;51:4553–62.
[11] Wilmer CE, Snurr RQ. Towards rapid computational screening of metal–organic frameworks for carbon dioxide capture: Calculation of framework charges via charge equilibration. Chem Eng J 2011;171:775–81.
[12] Kim J, Lin LC, Martin RL, Swisher JA, Haranczyk M, Smit B. Large-scale computational screening of zeolites for ethane/ethene separation. Langmuir 2012;28:11914–9.
[13] Banerjee D, Simon CM, Plonka AM, Motkuri RK, Liu J, Chen X, Smit B, Parise JB, Haranczyk M, Thallapally PK. Metal–organic framework with optimally selective xenon adsorption and separation. Nature Commun 2016;7:11831.
[14] Wilmer CE, Leaf M, Lee CY, Farha OK, Hauser BG, Hupp JT, Snurr RQ. Large-scale screening of hypothetical metal–organic frameworks. Nat Chem 2012;4:83–9.
[15] Martin RL, Lin LC, Jariwala K, Smit B, Haranczyk M. Mail-Order metal-organic frameworks (MOFs): Designing isoreticular MOF-5 analogues comprising commercially available organic molecules. J Phys Chem C 2013;117:12159–67.
[16] Keskin S, Liu J, Rankin RB, Johnson JK, Sholl DS. Progress, opportunities, and challenges for applying atomically detailed modeling to molecular adsorption and transport in metal-organic framework materials. Ind Eng Chem Res 2009;48:2355–71.
[17] Klauda JB, Venable RM, Freites JA, O'Connor JW, Tobias DJ, Mondragon-Ramirez C, Vorobyov I, MacKerell AD, Pastor RW. Update of the CHARMM all-atom additive force field for lipids: Validation on six lipid types. J Phys Chem B 2010;114:7830–43.
[18] Savelyev A, MacKerell AD. Balancing the interactions of ions, water, and DNA in the drude polarizable force field. J Phys Chem B 2014;118:6742–57.
[19] Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Zhong S, Shim J, Darian E, Guvench O, Lopes P, Vorobyov I, MacKerell AD. CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. J Comput Chem 2010;31:671–90.

[20] Potoff JJ, Bernard-Brunel DA. Mie potentials for phase equilibria calculations: application to alkanes and perfluoroalkanes. J Phys Chem B 2009;113:14725–31.
[21] Jambeck JPM, Lyubartsev AP. Derivation and systematic validation of a refined all-atom force field for phosphatidylcholine lipids. J Phys Chem B 2012;116:3164–79.
[22] Siu SWI, Pluhackova K, Bockmann RA. Optimization of the OPLS-AA force field for long hydrocarbons. J Chem Theory Comput 2012;8:1459–70.
[23] Dickson CJ, Rosso L, Betz RM, Walker RC, Gould IR. GAFFlipid: a general amber force field for the accurate molecular dynamics simulation of phospholipid. Soft Matter 2012;8:9617–27.
[24] Wang JM, Wolf RM, Caldwell JW, Kollman PA, Case DA. Development and testing of a general amber force field. J Comput Chem 2004;25:1157–74.
[25] Laury ML, Wang LP, Pande VS, Head-Gordon T, Ponder JW. Revised parameters for the AMOEBA polarizable atomic multipole water model. J Phys Chem B 2015;119:9423–37.
[26] Darve E, Pohorille A. Calculating free energies using average force. J Chem Phys 2001;115:9169–83.
[27] Chodera JD, Swope WC, Noe F, Prinz JH, Shirts MR, Pande VS. Dynamical reweighting: Improved estimates of dynamical properties from simulations at multiple temperatures. J Chem Phys 2011;134.
[28] Panagiotopoulos AZ. Direct determination of phase coexistence properties of fluids by Monte-Carlo simulation in a new ensemble. Mol Phys 1987;61:813–26.
[29] Balsera M, Stepaniants S, Izrailev S, Oono Y, Schulten K. Reconstructing potential energy functions from simulated force-induced unbinding processes. Biophys J 1997;73:1281–7.
[30] Chen B, Siepmann JI. Monte Carlo algorithms for simulating systems with adiabatic separation of electronic and nuclear degrees of freedom. Theor Chem Acc 1999;103:87–104.
[31] Lamoureux G, Roux B. Modeling induced polarization with classical Drude oscillators: Theory and molecular dynamics simulation algorithm. J Chem Phys 2003;119:3025–39.
[32] Jiang W, Hardy DJ, Phillips JC, MacKerell AD, Schulten K, Roux B. High-Performance scalable molecular dynamics simulations of a polarizable force field based on classical drude oscillators in namd. J Phys Chem Lett 2011;2:87–92.
[33] Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, Schulten K. Scalable molecular dynamics with NAMD. J Comput Chem 2005;26:1781–802.
[34] Brown WM, Wang P, Plimpton SJ, Tharrington AN. Implementing molecular dynamics on hybrid high performance computers - short range forces. Comput Phys Commun 2011;182:898–911.
[35] Salomon-Ferrer R, Case DA, Walker RC. An overview of the Amber biomolecular simulation package. Wires Comput Mol Sci 2013;3:198–210.
[36] Salomon-Ferrer R, Gotz AW, Poole D, Gr S Le, Walker RC. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. Explicit solvent particle mesh ewald. J Chem Theory Comput 2013;9:3878–88.
[37] Nguyen TD, Phillips CL, Anderson JA, Glotzer SC. Rigid body constraints realized in massively-parallel molecular dynamics on graphics processing units. Comput Phys Commun 2011;182:2307–13.
[38] Abraham MJ, Murtola T, Schulz R, Páll S, Smith JC, Hess B, Lindahl E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 2015;1–2:19–25.
[39] Brooks BR, Brooks CL, Mackerell AD, Nilsson L, Petrella RJ, Roux B, Won Y, Archontis G, Bartels C, Boresch S, Caflisch A, Caves L, Cui Q, Dinner AR, Feig M, Fischer S, Gao J, Hodoscek M, Im W, Kuczera K, Lazaridis T, Ma J, Ovchinnikov V, Paci E, Pastor RW, Post CB, Pu JZ, Schaefer M, Tidor B, Venable RM, Woodcock HL, Wu X, Yang W, York DM, Karplus M. CHARMM: The biomolecular simulation program. J Comput Chem 2009;30:1545–614.
[40] Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, Merz KM, Onufriev A, Simmerling C, Wang B, Woods RJ. The amber biomolecular simulation programs. J Comput Chem 2005;26:1668–88.
[41] Hynninen AP, Crowley MF. New faster CHARMM molecular dynamics engine. J Comput Chem 2014;35:406–13.
[42] Plimpton S. Fast parallel algorithms for short-range molecular-dynamics. J Comput Phys 1995;117:1–19.
[43] Anderson JA, Lorenz CD, Travesset A. General purpose molecular dynamics simulations fully implemented on graphics processing units. J Comput Phys 2008;227:5342–59.
[44] Errington JR, Panagiotopoulos AZ. Computational methodologies and potentials for phase equilibria. 1999, Available from: http://paros.princeton.edu/ppe/index.html.
[45] MUSIC: http://zeolites.cqe.northwestern.edu/Music/.
[46] ProtoMS: www.essexgroup.soton.ac.uk/ProtoMS.
[47] Deublein S, Eckl B, Stoll J, Lishchuk SV, Guevara-Carrion G, Glass CW, Merker T, Bernreuther M, Hasse H, Vrabec J. ms2: A molecular simulation tool for thermodynamic properties. Comput Phys Commun 2011;182:2350–67.
[48] BOSS: http://zarbi.chem.yale.edu/software.html.
[49] Hey T, Payne MC. Open science decoded. Nat Phys 2015;11:367–9.
[50] Lemmon EW, M.M. O, Friend DG. Thermophysical properties of fluid systems, in: P.J. Linstrom and W.G. Mallard (Eds.), NIST Chemistry WebBook, NIST Standard Reference Database Number 69, National Institute of Standards and Technology, Gaithersburg MD, 20899, http://webbook.nist.gov.
[51] Boulougouris GC, Economou IG, Theodorou DN. Engineering a molecular model for water phase equilibrium over a wide temperature range. J Phys Chem B 1998;102:1029–35.
[52] Yazaydin AO, Snurr RQ, Park TH, Koh K, Liu J, LeVan MD, Benin AI, Jakubczak P, Lanuza M, Galloway DB, Low JJ, Willis RR. Screening of metal-organic frameworks for carbon dioxide capture from flue gas using a combined experimental and modeling approach. J Am Chem Soc 2009;131:18198–9.
[53] Martin MG. MCCCS Towhee: a tool for Monte Carlo molecular simulation. Mol Simul 2013;39:1184–94.
[54] Gowers RJ, Farmahini AH, Friedrich D, Sarkisov L. Automated analysis and benchmarking of GCMC simulation programs in application to gas adsorption. Mol Simul 2018;44:309–21.
[55] Anderson JA, Jankowski E, Grubb TL, Engel M, Glotzer SC. Massively parallel Monte Carlo for many-particle simulations on GPUs. J Comput Phys 2013;254:27–38.
[56] Schultz AJ, Kofke DA. Etomica: An object-oriented framework for molecular simulation. J Comput Chem 2015;36:573–83.
[57] Hatch HW, Mahynski NA, Shen VK. FEASST: Free energy and advanced sampling simulation toolkit. J Res Natl Inst Stan 2018;123.
[58] Glass CW, Reiser S, Rutkai G, Deublein S, Koster A, Guevara-Carrion G, Wafai A, Horsch M, Bernreuther M, Windmann T, Hasse H, Vrabec J. ms2: A molecular simulation tool for thermodynamic properties, new version release. Comput Phys Commun 2014;185:3302–6.
[59] Rutkai G, Koster A, Guevara-Carrion G, Janzen T, Schappals M, Glass CW, Bernreuther M, Wafai A, Stephan S, Kohns M, Reiser S, Deublein S, Horsch M, Hasse H, Vrabec J. ms2: A molecular simulation tool for thermodynamic properties, release 3.0. Comput Phys Commun 2017;221:343–51.
[60] Dubbeldam D, Calero S, Ellis DE, Snurr RQ. RASPA: molecular simulation software for adsorption and diffusion in flexible nanoporous materials. Mol Simul 2016;42:81–101.
[61] Shah JK, Marin-Rimoldi E, Mullen RG, Keene BP, Khan S, Paluch AS, Rai N, Romanielo LL, Rosch TW, Yoo B, Maginn EJ. Cassandra: An open source Monte Carlo package for molecular simulation. J Comput Chem 2017;38:1727–39.
[62] Martin MG, Siepmann JI. Novel configurational-bias Monte Carlo method for branched molecules. Transferable potentials for phase equilibria. 2. United-atom description of branched alkanes. J Phys Chem B 1999;103:4508–17.
[63] Chen B, Potoff JJ, Siepmann JI. Monte Carlo calculations for alcohols and their mixtures with alkanes. 5. Transferable potentials for phase equilibria. United-atom description of primary, secondary, and tertiary alcohols. J Phys Chem B 2001;105:3093–104.
[64] Lubna N, Kamath G, Potoff JJ, Rai N, Siepmann JI. Transferable potentials for phase equilibria. 8. United-atom description for thiols, sulfides, disulfides, and thiophene. J Phys Chem B 2005;109:24100–7.
[65] Martin MG, Siepmann JI. Transferable potentials for phase equilibria. 1. United-atom description of n-alkanes. J Phys Chem B 1998;102:2569–77.
[66] Potoff JJ, Siepmann JI. Vapor–liquid equilibria of mixtures containing alkanes, carbon dioxide, and nitrogen. Aiche J 2001;47:1676–82.
[67] Stubbs JM, Potoff JJ, Siepmann JI. Transferable potentials for phase equilibria. 6. United-atom description for ethers, glycols, ketones, and aldehydes. J Phys Chem B 2004;108:17596–605.
[68] Wick CD, Martin MG, Siepmann JI. Transferable potentials for phase equilibria. 4. United-atom description of linear and branched alkenes and alkylbenzenes. J Phys Chem B 2000;104:8008–16.
[69] Wick CD, Stubbs JM, Rai N, Siepmann JI. Transferable potentials for phase equilibria. 7. Primary, secondary, and tertiary amines, nitroalkanes and nitrobenzene, nitriles, amides, pyridine, and pyrimidine. J Phys Chem B 2005;109:18974–82.
[70] Mick JR, Barhaghi MS, Potoff JJ. Prediction of radon-222 phase behavior by Monte Carlo simulation. J Chem Eng Data 2016;61:1625–31.
[71] Mick JR, Soroush Barhaghi M, Jackman B, Rushaidat K, Schwiebert L, Potoff JJ. Optimized Mie potentials for phase equilibria: Application to noble gases and their mixtures with n-alkanes. J Chem Phys 2015;143:114504.
[72] Potoff JJ, Kamath G. Mie potentials for phase equilibria: application to alkenes. J Chem Eng Data 2014;59:3144–50.
[73] Barhaghi MS, Mick JR, Potoff JJ. Optimised mie potentials for phase equilibria: application to alkynes. Mol Phys 2017;115:1378–88.
[74] Mick JR, Barhaghi MS, Jackman B, Schwiebert L, Potoff JJ. Optimized mie potentials for phase equilibria: Application to branched alkanes. J Chem Eng Data 2017;62:1806–18.
[75] Jorgensen WL, Maxwell DS, TiradoRives J. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. J Am Chem Soc 1996;118:11225–36.
[76] Marrink SJ, Risselada HJ, Yefimov S, Tieleman DP, de Vries AH. The MARTINI force field: Coarse grained model for biomolecular simulations. J Phys Chem B 2007;111:7812–24.
[77] Ferrenberg AM, Swendsen RH. New Monte-Carlo technique for studying phase-transitions. Phys Rev Lett 1988;61:2635–8.

[78] Ferrenberg AM, Swendsen RH. Optimized Monte-Carlo data-analysis. Phys Rev Lett 1989;63:1195–8.
[79] Humphrey W, Dalke A, Schulten K. VMD: visual molecular dynamics. J Mol Graph Model 1996;14:33–8.
[80] GOMC: GPU Optimized Monte Carlo GitHub Repository, https://github.com/GOMC-WSU/, 2018.
[81] GOMC Project Website, http://gomc.eng.wayne.edu, 2018.
[82] GOMC Examples, https://github.com/GOMC-WSU/GOMC_Examples, 2018.
[83] GOMC Tutorials, https://github.com/GOMC-WSU/Workshop, 2018.
[84] Wells BA, Chaffee AL. Ewald summation for molecular simulations. J Chem Theory Comput 2015;11:3684–95.
[85] Vlugt TJH, Garcia-Perez E, Dubbeldam D, Ban S, Calero S. Computing the heat of adsorption using molecular simulations: The effect of strong Coulombic interactions. J Chem Theory Comput 2008;4:1107–18.
[86] Berendsen HJC, Grigera JR, Straatsma TP. The missing term in effective pair potentials. J Phys Chem 1987;91:6269–71.
[87] Frenkel D, Smit B. Understanding Molecular Simulation: From Algorithms To Applications. Academic Press; 2002.
[88] Ewald PP. The calculation of optical and electrostatic grid potential. Ann Phys 1921;64:253–87.
[89] Eddaoudi M, Kim J, Rosi N, Vodak D, Wachter J, O'Keeffe M, Yaghi OM. Systematic design of pore size and functionality in isoreticular mofs and their application in methane storage. Science 2002;295:469–72.
[90] Mayo SL, Olafson BD. Dreiding - a generic force-field for molecular simulations. J Phys Chem 1990;94:8897–909.
[91] GOMC Benchmarks: https://github.com/GOMC-WSU/Benchmarks, 2018.
[92] Mick J, Hailat E, Russo V, Rushaidat K, Schwiehert L, Potoff J. GPU-accelerated Gibbs ensemble Monte Carlo simulations of Lennard-Jonesium. Comput Phys Commun 2013;184:2662–9.
[93] Torres-Knoop A, Balaji SP, Vlugt TJH, Dubbeldam D. A comparison of advanced Monte Carlo methods for open systems: CFCMC vs CBMC. J Chem Theory Comput 2014;10:942–52.
[94] Torres-Knoop A, Burtch NC, Poursaeidesfahani A, Balaji SP, Kools R, Smit FX, Walton KS, Vlugt TJH, Dubbeldam D. Optimization of particle transfers in the gibbs ensemble for systems with strong and directional interactions using cbmc, cfcmc, and cb/cfcmc. J Phys Chem C 2016;120:9148–59.
[95] Shi W, Maginn EJ. Improvement in molecule exchange efficiency in Gibbs ensemble Monte Carlo: Development and implementation of the continuous fractional component move. J Comput Chem 2008;29:2520–30.
[96] Shi W, Maginn EJ. Continuous fractional component Monte Carlo: An adaptive biasing method for open system atomistic simulations. J Chem Theory Comput 2007;3:1451–63.