SoftwareX 13 (2021) 100627

Contents lists available at ScienceDirect

SoftwareX

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

Original Software Publication

Update 2.70 to "GOMC: GPU Optimized Monte Carlo for the simulation of phase equilibria and physical properties of complex fluids"

Younes Nejahi [a], Mohammad Soroush Barhaghi [b], Gregory Schwing [a], Loren Schwiebert [a], Jeffrey Potoff [b,*]

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

ARTICLE INFO

Article history:
Received 6 November 2020
Accepted 10 November 2020

Keywords:
Multi-particle
Exp-6 potential
Alchemical free energy
Cyclic molecules
Crankshaft move
Molecular Exchange Monte Carlo

ABSTRACT

Major updates in version 2.70 of GOMC include new Monte Carlo moves to enhance the sampling of phase space, such as Molecular Exchange Monte Carlo (MEMC), configurational-bias for molecules that contain rings, the crankshaft move, and a force/torque-biased multi-particle move. Support for force fields governed by exp-6 potentials, and free energy calculations using thermodynamic integration or free energy perturbation has been added. The GPU performance of the multi-particle move has been improved significantly from version 2.50, and memory usage has been reduced significantly.

Code metadata

Current Code version	v2.70
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-20-00079
Legal Code License	APGL-3.0
Code Versioning system used	Git
Software Code Language used	C, C++, CUDA, OpenMP, MPI
Compilation requirements, Operating environments & dependencies	ANSI C and C++14, CUDA compiler, cmake
If available Link to developer documentation/manual	https://gomc-wsu.github.io/Manual/
Support email for questions	gomc@eng.wayne.edu

Software metadata

Current software version	v2.31
Permanent link to executables of this version	https://github.com/GOMC-WSU/GOMC/releases/tag/v2.31
Legal Software License	AGPL-3.0
Computing platform/Operating System	Linux, macOS, Microsoft Windows
Installation requirements & dependencies	
If available Link to user manual – if formally published include a reference to the publication in the reference list	https://gomc-wsu.github.io/Manual/
Support email for questions	gomc@eng.wayne.edu

1. Motivation and significance

GOMC is a general-purpose Monte Carlo simulation engine for the simulation of molecular systems with molecular mechanics force fields based on the 12-6 Lennard-Jones, or Mie potentials [1]. It has support for simulations in all common ensembles, including the Gibbs ensemble Monte Carlo algorithm. GOMC was designed with a focus on high performance and has support for simulations on multicore CPUs and graphics processing units (GPUs). This paper describes a number of enhancements to GOMC, including new types of Monte Carlo moves, as well as support for alchemical free energy calculations, and new

DOI of original article: https://doi.org/10.1016/j.softx.2018.11.005.
* Corresponding author.
E-mail address: jpotoff@wayne.edu (Jeffrey Potoff).

https://doi.org/10.1016/j.softx.2020.100627

intermolecular potentials, which expand significantly the scope of research problems that may be studied with GOMC.

2. Updates to Monte Carlo sampling algorithms

Three new Monte Carlo sampling algorithms have been added to GOMC. These include a crankshaft move [2,3], an extension to the coupled–decoupled configurational-bias (CD-CBMC) algorithm [4] to support molecules that contain rings, and a generalized version of the swap+identity (swatch) algorithm [5], called Molecular Exchange Monte Carlo (MEMC) [6,7], and a multi-particle move [8].

The crankshaft move is designed to improve conformational sampling of long chain molecules, such as polymers, by allowing for internal rearrangements of atoms. In the crankshaft move, a trial rotation is performed around the shaft formed between two selected atoms. The probability of generating such a new configuration is

$$P_{\text{gen}}(i) = \exp\left(-\beta\left(U_{\text{bend}}(i) + U_{\text{tors}}(i) + U_{\text{LJ}}(i)\right)\right)/W \quad (1)$$

where $W = \sum_{i=0}^{\text{rot-trial}} \exp\left(-\beta\left(U_{\text{bend}}(i) + U_{\text{tors}}(i) + U_{\text{LJ}}(i)\right)\right)$. The naïve implementation using the CBMC algorithm is computationally inefficient, since very few of the trial rotations result in reasonable bond angles, requiring large numbers of rotational trials to be generated. This also requires a large number of expensive calculations for the non-bonded interaction energies. To improve the efficiency of the crankshaft move, a coupled, biased selection was used for the intermolecular energy, and a decoupled, biased selection was used for bending and torsional energies. In this approach, the probability of generating a trial configuration is

$$P_{gen}(i) = \left[\exp\left(-\beta U_{\text{LJ}}(i)\right) W_B(i)/W_{\text{NB}}\right] \times \left[\exp\left(-\beta\left(U_{\text{bend}}(j) + U_{\text{tors}}(j)\right)\right)/W_{\text{B}}\right], \quad (2)$$

where $W_{\text{NB}} = \sum_{i=0}^{\text{LJ-trial}} \exp\left(-\beta U_{\text{LJ}}(i)\right) W_{\text{B}}(i)$ and $W_{\text{B}}(i) = \sum_{j=0}^{\text{rot-trial}} \exp\left(-\beta\left(U_{\text{bend}}(j) + U_{\text{tors}}(j)\right)\right)$. This implementation has a higher acceptance probability and is computationally more efficient, since it requires significantly fewer intermolecular energy calculations.

Monte Carlo simulations of molecules containing rings require specialized configurational-bias algorithms to properly sample phase space. Two examples are self-adapting fixed endpoint CBMC (SAFE-CBMC) [9] and reservoir Monte Carlo [10]. These methods have some limitations. For example, SAFE-CBMC requires preexisting knowledge of the distance distribution between atoms in the ring. The reservoir method does not require preexisting knowledge of distance distribution, but can require large amounts of memory to store the required library of conformers.

The GOMC implementation of configurational-bias for rings attempts to address these aforementioned weaknesses. It does not require any knowledge of atom distance distributions, nor does it require a reservoir of conformers. Atoms that belong to the ring are inserted rigidly using CBMC, while the rest of the atoms attached to the ring are grown using CD-CBMC. Angles and dihedrals that belong to the ring are kept fixed, while the rest of the angles and dihedrals are generated using CD-CBMC. During the simulation, crankshaft moves are used to sample angles and dihedrals within the ring. This methodology allows for the simulation of vapor–liquid equilibria for polycyclic compounds, including rings connected by flexible linkers, as shown in Fig. 1. While this method works for a large number of ring-containing molecules, it does not work for molecules that contain three or more flexible rings that share one or more atoms (e.g. acenaphthene). This is because the crankshaft move cannot be applied to alter the bond angle (with the shared atom at center), without also changing bond lengths.

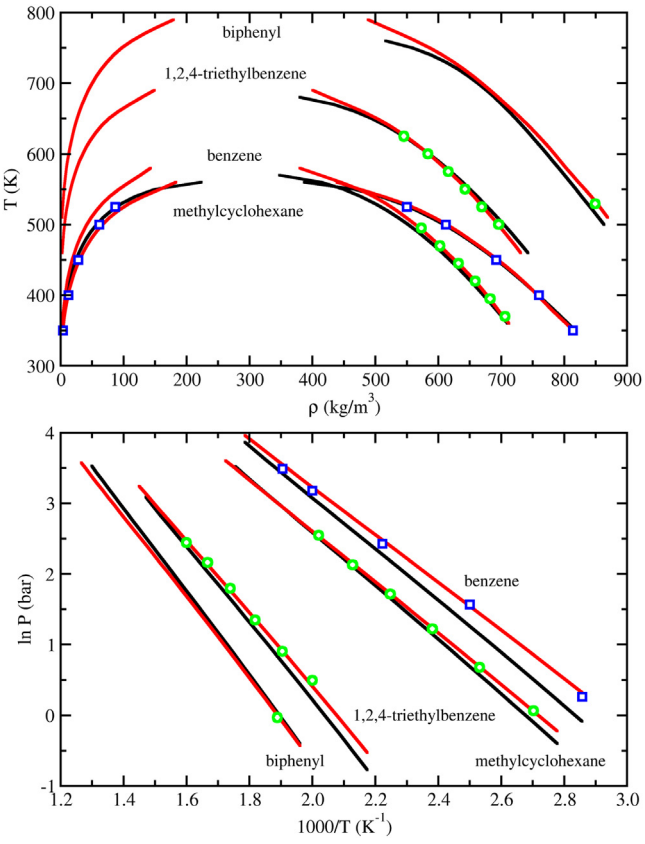

Fig. 1. Vapor–liquid coexistence curves predicted by GOMC (red lines) for methylcyclohexane, benzene, 1,2,4-triethylbenzene, and biphenyl compared to the work of Wick et al. [11] (blue squares) and Yiannourakou et al. [12] (green circles). Solid black lines correspond to experimental data [13].

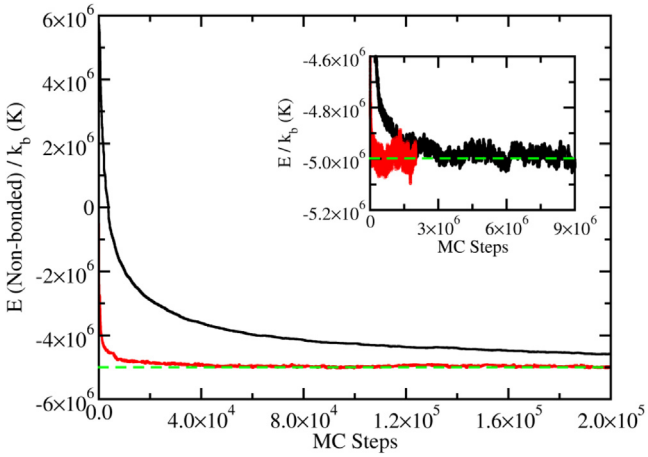

Fig. 2. NVT ensemble Monte Carlo simulations of SPC water at 982 kg/m^3 and 298 K using single molecule translations and displacements (black line) and the multi-particle move (red line). Average energy determined from equilibrated simulation (green dashed line).

Molecular Exchange Monte Carlo is a generalized version of the combined swap and identity exchange proposed by Martin and Siepmann [5] to enhance molecule insertion/deletion in dense system, and has been implemented for both the grand canonical (GCMC) [7] and Gibbs ensemble Monte Carlo (GEMC) [6]. In MEMC, the molecules to be exchanged are not required

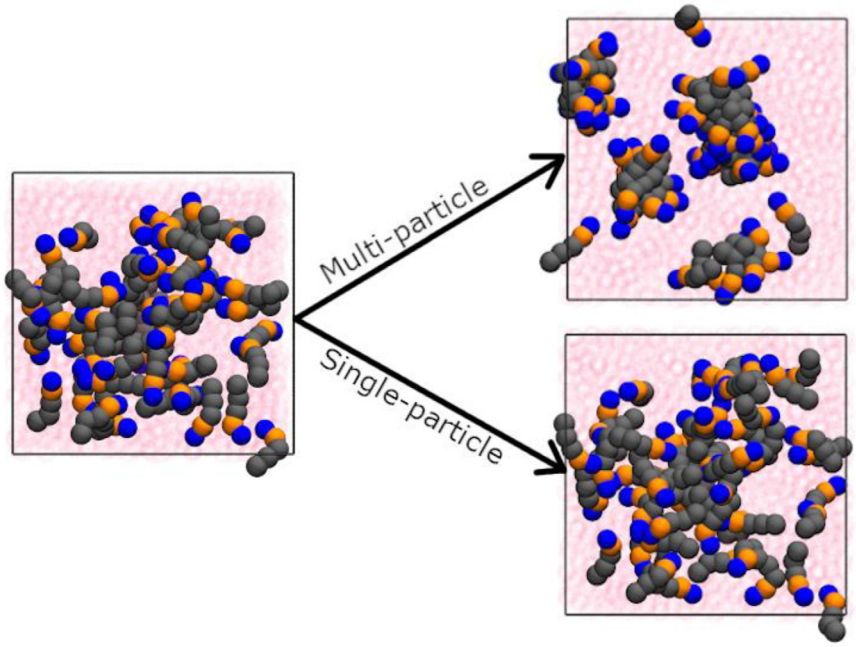

Fig. 3. Snapshots from NVT ensemble Monte Carlo simulations of dodecyl-phosphocholine (DPC) modeled with the MARTINI force field [14] using the multi-particle move after 50M steps (upper) and single molecule translations and rotations after 100M steps (lower).

to have common atom types or coordinates, and molecules to be exchanged can be exchanged as rigid bodies or regrown atom by atom using the CD-CBMC algorithm. The MEMC move enhances sampling in multi-component mixtures or pure fluids using the technique of impurity atoms proposed by Bai et al. [15]. For example, in simulations to determine the vapor–liquid coexistence curve of water with GCMC simulations, the MEMC move was shown to have a computational efficiency over 38 times that of standard CD-CBMC insertions [7]. These improvements in sampling efficiency for molecule exchanges enable GCMC simulations to be used to generate phase diagrams at temperatures as low as $0.5T_c$, while GCMC simulations using standard CD-CBMC insertions are typically limited to $0.7T_c$ [7]. In Gibbs ensemble Monte Carlo simulations, the MEMC move enables the efficient calculation of solvation free energies [6].

Simulations of heterogeneous systems, such as lipid bilayers or micelles, require Monte Carlo moves that can update the positions for collections of molecules simultaneously. Additionally, simulations of polarizable force fields require a complete recalculation of the electrostatic interactions whenever a single molecule is moved, resulting in very poor computational performance in Monte Carlo simulations. Molecular dynamics simulations of polarizable force fields, such as Drude oscillators, typically use an extended Lagrangian approach to reduce computational effort [16,17], which is efficient because all of the molecules in the system are being moved in a single time step. To address the limitations of standard Monte Carlo moves, when applied to self-assembly and polarizable force fields, the force/torque biased multi-particle move proposed by Moucka et al. has been implemented in GOMC [18]. To improve the efficiency of the multi-particle move on GPUs, the counter-based random number generator Random123 library was used [19]. The multi-particle implementation in GOMC supports independently translating all molecules or rotating all molecules at once. Multi-particle moves that combine displacement and rotation in the same move were found to cause stalling of the simulation, leading to errors in the calculation of ensemble averages, and therefore are not recommended. The multi-particle move reduces substantially the number of Monte Carlo steps required for equilibration in self-assembling systems. In NVT simulations of SPC water [20] at 298 K with the multi-particle move, shown in Fig. 2, equilibration was achieved in an order of magnitude fewer steps than Monte Carlo simulation with single molecule displacements and rotations. Additionally, NVT Monte Carlo simulations of dodecyl-phosphocholine (DPC) at 300 K, modeled with the MARTINI force field [14] (54 DPC molecules, 5900 waters and 500 anti-freeze particles), presented in Fig. 3, show self-assembly when simulated with the multi-particle move for 50 million MC steps, whereas single molecule translations and rotations fail to show any self-assembly after 100 million MC steps.

3. Free energy calculations

Support for free energy calculations using either thermodynamic integration or free energy perturbation has been added to GOMC [21]. Soft-core scaling is used for the Lennard-Jones interactions, while linear scaling is used for the Coulombic interactions. Separate coupling parameters, λ_{LJ} and λ_{Coul}, are used to independently control the scaling of Lennard-Jones and Coulombic interactions, respectively. During the simulation, the change in energy $\Delta U_{i\rightarrow j}$ between the current intermediate state (λ_i) and all other intermediate states ($\lambda_{j\neq i}$), and the derivative of Lennard-Jones and Coulomb potential with respect to lambda ($dU_{\mathrm{Coul}}/d\lambda_{\mathrm{Coul}}$, $dU_{\mathrm{LJ}}/d\lambda_{\mathrm{LJ}}$), are evaluated and stored for post-simulation analysis. The output from GOMC is formatted so it can be analyzed with both alchemlyb [22] and alchemical-analysis [23]. Since alchemical-analysis is no longer supported by its authors, the GOMC parser for it was stored in a separate GitHub repository [24].

4. Conclusion and future updates

This update highlights a number of new sampling algorithms and code features that have been added in this release. Future versions of GOMC will include support for replica exchange simulations, polarizable force fields using Drude oscillators, a Brownian dynamics multi-particle move, and improved performance on multicore and GPU architectures.

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.

Acknowledgments

Financial support from the National Science Foundation, USA 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. GPU performance improvements are based on guidance received during SDSC and ONL hackathons from Muaaz Awan, Robert Sinkovits, and Kwangmin Yu.

References

[1] Nejahi Y, Barhaghi MS, Mick J, Jackman B, Rushaidat K, Li Y, et al. GOMC: GPU optimized Monte Carlo for the simulation of phase equilibria and physical properties of complex fluids. SoftwareX 2019;9:20–7.
[2] Baumgartner A, Binder K. Monte-Carlo Studies on the freely jointed polymer-chain with excluded volume interaction. J Chem Phys 1979;71(6):2541–5.
[3] Pertsin AJ, Hahn J, Grossmann HP. Incorporation of bond-length constraints in Monte-Carlo simulations of cyclic and linear-molecules - conformational sampling for cyclic alkanes as test systems. J Comput Chem 1994;15(10):1121–6.
[4] 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(21):4508–17.
[5] Martin MG, Siepmann JI. Predicting multicomponent phase equilibria and free energies of transfer for alkanes by molecular simulation. J Amer Chem Soc 1997;119(38):8921–4.
[6] Barhaghi MS, Potoff JJ. Prediction of phase equilibria and Gibbs free energies of transfer using molecular exchange Monte Carlo in the Gibbs ensemble. Fluid Phase Equilib 2019;486:106–18.
[7] Barhaghi MS, Torabi K, Nejahi Y, Schwiebert L, Potoff JJ. Molecular exchange Monte Carlo: A generalized method for identity exchanges in grand canonical Monte Carlo simulations. J Chem Phys 2018;149(7):072318.
[8] Moucka F, Rouha M, Nezbeda I. Efficient multiparticle sampling in Monte Carlo simulations on fluids: Application to polarizable models. J Chem Phys 2007;126(22):224106.
[9] Wick CD, Siepmann JI. Self-adapting fixed-end-point configurational-bias Monte Carlo method for the regrowth of interior segments of chain molecules with strong intramolecular interactions. Macromolecules 2000;33(19):7207–18.
[10] Shah JK, Maginn EJ. A general and efficient Monte Carlo method for sampling intramolecular degrees of freedom of branched and cyclic molecules. J Chem Phys 2011;135(13):134121.
[11] 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(33):8008–16.
[12] Yiannourakou M, Ungerer P, Lachet V, Rousseau B, Teuler JM. United atom forcefield for vapor–liquid equilibrium (VLE) properties of cyclic and polycyclic compounds from Monte Carlo simulations. Fluid Phase Equilib 2019;481:28–43.
[13] Frenkel M, Chirico RD, Diky V, Yan XJ, Dong Q, Muzny C. Thermodata engine (TDE): Software implementation of the dynamic data evaluation concept. J Chem Inf Model 2005;45(4):816–38.
[14] 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(27):7812–24.
[15] Bai P, Siepmann JI. Assessment and optimization of configurational-bias Monte Carlo particle swap strategies for simulations of water in the gibbs ensemble. J Chem Theory Comput 2017;13(2):431–40.
[16] 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(2):87–92.
[17] Lemkul JA, Roux B, van der Spoel D, MacKerell AD. Implementation of extended Lagrangian dynamics in GROMACS for polarizable simulations using the classical drude oscillator model. J Comput Chem 2015;36(19):1473–9.
[18] Moucka F, Rouha M, Nezbeda I. Efficient multiparticle sampling in Monte Carlo simulations on fluids: application to polarizable models. J Chem Phys 2007;126(22):224106.
[19] Salmon JK, Moraes MA, Dror RO, Shaw DE. Parallel random numbers: as easy as 1, 2, 3. In: Proceedings of 2011 International conference for high performance computing, networking, storage and analysis. Seattle, Washington: Association for Computing Machinery; 2011, Article 16.
[20] Berendsen HJC, Postma JPM, van Gunsteren WF, Hermans J. Intermolecular forces. Reidel: Dordrecht; 1981,
[21] Barhaghi MS, Luyet C, Potoff JJ. Effect of fluorination on the partitioning of alcohols. Mol Phys 2019;117(23–24):3827–39.
[22] Dotson D, K. I, Beckstein O. alchemistry/alchemlyb: Release 0.1.0. 2017, http://dx.doi.org/10.5281/zenodo.583647.
[23] Klimovich PV, Shirts MR, Mobley DL. Guidelines for the analysis of free energy calculations. J Comput Aided Mol Des 2015;29(5):397–411.
[24] Soroush Barhaghi M. Alchemical-analysis GOMC parser. 2020, https://github.com/msoroush/alchemical-analysis.