Open-Source Molecular Modeling Software in Chemical Engineering

Peter T. Cummings and Justin B. Gilmer
Department of Chemical Engineering and
Multiscale Modeling and Simulation Center
Vanderbilt University
Nashville, Tennessee 37235-1604, United States

Abstract
The molecular modeling community within chemical engineering has been rapidly growing for several decades. This was recognized formally by the American Institute of Chemical Engineers in 2000 with the establishment of a new programming group, the Computational and Molecular Science and Engineering From (CoMSeF). Many researchers in this community have embraced the principles of the open source software, and have made significant contributions to the portfolio of open-source software available for use by the entire scientific and engineering community. In this article, we briefly review the concepts of molecular modeling and open-source software, and provide an overview of some of these contributions from the chemical engineering community. Additional information is provided in an extensive supplementary information (SI) addendum.

1. Introduction
Modern chemical engineering is characterized by its emphasis on creating products by exploiting the molecular/atomic nature of materials. While many academic departments of chemical engineering have added "biomolecular" to their name, in recognition of the molecular approach to bioengineering that is the hallmark of chemical engineering, this emphasis on understanding the molecular basis of phenomena is true across the whole discipline. As just one example, reaction engineering, and particularly catalysis, have been transformed by understanding the molecular basis for reactivity in bulk and at surfaces. This has been achieved by a combination of experimental probes that convey molecular-level information (such as neutron and x-ray scattering techniques, and single-molecule fluorescence spectroscopy for biocatalysis) and computational studies, based on molecular and atomic level techniques such as density functional theory (DFT) and orbital-based quantum chemistry (OBQC) methods. The continuing evolution of DFT, in particular, along with the continued exponential increase in computing power, is leading to the emergence of the field of computational catalysis, with the goal of designing catalytic pathways and systems entirely on the computer [1,2]. DFT and OBQC are examples of techniques that fall under the general rubric of molecular modeling (MM), a term to describe theoretically-derived computational methods for predicting the properties of molecules individually and/or collectively. DFT and OBQC are among the most fundamental MM methods, since they make the fewest assumptions, and include the computation of electron density in a given system, essentially by solving a variant of the Schrodinger equation or its equivalent, thereby permitting the study of reactions. Typical results from these calculations are the total energy of the system, optimized geometries of molecules, and electronic properties of the system (including distribution of charges on atoms).

Perhaps the most widely used and best known MM method is molecular dynamics (MD) simulation, in which the dynamics of atoms or molecules in a system are solved numerically, typically with a time step of $O(10^{-15}\mathrm{s})$ for a length of time ranging from picoseconds to microseconds. The positions and velocities of all of these atoms over time, known as a trajectory,

can be analyzed by averaging to compute microscopic (e.g., structure) and macroscopic properties (e.g., pressure and energy). The key input to an MD simulation is the model for the potential energy $U(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$ of the system in terms of all the positions $\mathbf{r}_i$ of every atom; this potential energy, known as a force field, includes intermolecular interactions (e.g., van der Waals attraction and repulsion and electrostatic interactions) and intramolecular interactions (e.g., bond stretching, bond bending, and torsional interactions). The trajectory is obtained by numerically solving Newton's equations,

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \mathbf{F}_i = -\nabla_{\mathbf{r}_i} U(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) \quad (1)$$

or some variant thereof. In this equation, m_i and $\mathbf{F}_i$ are respectively the mass of, and the force on, atom i, and $\nabla_{\mathbf{r}_i}$ is the gradient operator with respect to the position of atom i. Solving Newton's equations yields a system at fixed number of molecules N, volume V, and total (potential + kinetic) energy E, often referred to as NVE. In statistical mechanics, this corresponds to the microcanonical ensemble. Other ensembles, such as the canonical ensemble (NVT) and isobaric-isothermal (NPT), where T is temperature and P is pressure, are generated by variants of Newton's equations designed to constrain the property or properties of interest. Another MM simulation methodology that also employs force fields is Monte Carlo (MC) simulation, in which the "trajectory" is now a series of configurations of the system generated stochastically via a Markov chain algorithm designed to ensure that asymptotically the configurations are distributed according to the known statistical mechanical distribution for the ensemble being simulated (e.g., the Boltzmann distribution for the NVT ensemble, in which the probability of a configuration α is given by $P_\alpha \propto \exp(-U_\alpha/k_BT)$, where U_α is the potential energy of the configuration and k_B is Boltzmann's constant). In MC simulations, there is no formal concept of time, and no velocities are computed, as the momentum degrees of freedom are assumed to have their equilibrium distribution. While simple conceptually, MD and MC have evolved over the last 6-7 decades since their inception in the 1950s. Today, using various specialized techniques, and taking advantage of fast, multiprocessor computers, MD and MC are used routinely to predict the properties of complex systems, such as polymers, biomolecular systems and, in particular, materials whose functionality is derived from nanoscale structure. For many years, it was typical for each research group to maintain "the group code" that was capable of performing MD or MC simulations of the systems of interest to that group. Such codes were typically not shared, as they represented the group's competitive advantage. However, the need to model increasingly complex systems leads naturally to making use of the best available computational resources. The advent of massively parallel supercomputers in the 1990s, and more recently the entry of graphical processing units (GPUs) into the high performance computing environment, complicated dramatically the process of writing and maintaining MD and MC codes that could efficiently exploit these new architectures. This led to the rise of general purpose community MD codes, such as LAMMPS (initial release 1995) [3,4], GROMACS (1991) [5,6] and HOOMD-Blue (2008) [7,8], the latter specifically optimized for GPU architectures), and MC codes such as MCCS Towhee (1999) [9,10] and Cassandra (2011) [11,12]. It should be noted that LAMMPS and GROMACS now leverage GPU computing as well. The general purpose MD codes noted above were preceded by MD codes focused specifically on simulating biological systems, CHARMM (1983) [13,14] and AMBER (1981) [15,16] that particularly implement forcefields of the same name; NAMD (1994) [16,17] was the first biology-centric code to be developed from the ground up for parallel supercomputers.

Many of these community codes are made available under open source licenses. Open source is a general term used for software that typically is freely distributed in the form of source code according to one of several specific licenses (see accompanying SI). LAMMPS, GROMACS, HOOMD-Blue, Towhee and Cassandra are all open source: the source code is freely downloadable and usable by anyone, and can be re-used subject to the particulars of the open

source license used. Communities have grown up around these codes whereby users of the codes can become contributors, finding/correcting bugs and extending capabilities that can in turn be distributed in future versions of the codes. The codes are often hosted on open-source version-controlled code repositories such as GitHub (https://github.com). In contrast, CHARMM and NAMD are freeware, available to academic users for free, but not to commercial users; AMBER has a minimal cost for academic users. The source codes are generally not available, and the developer group is typically closed. Commercial MD and MC simulation packages are also available, but will not be discussed here.

The *advantages* of open-source software are many – bugs can be discovered early, since the user communities can see the source code, and versions that correct a particular bug can be quickly disseminated between major releases, often overnight. Since the codes are available to anyone, published results obtained with the codes can in principle be reproduced by others willing to go through the steps of installing and running the code; commercial or distribution-limited codes can only be reproduced only by the paid licensees of the code. In science and engineering research reproducibility is an important consideration. Indeed, reproducibility in scientific research has become a prominent issue, to the extent that some have opined that science has a "reproducibility crisis" [18]; while a code being open source does not guarantee reproducibility, since this may depend on knowing additional details about how a computation is performed that may not be provided in a typical journal article, it makes it far more likely. The primary *disadvantage* of open source software is that it usually requires some expertise in coding to implement and use. Generally, open source software does not support graphical user interfaces (GUIs), but are run using scripts: from a reproducibility point of view, this may be viewed as an advantage - one of the most vocal proponents of reproducibility in computational science [19], Lorena Barba of George Washington U, has gone as far asserting that GUIs are the enemy of reproducibility. Hence, open source software is often used by experienced researchers, rather than novice users.

Beyond MM, open source software plays a role in other areas of chemical engineering. For example, GNU Octave [20], which originated as a project led by Jim Rawlings and then graduate student John Eaton of the University of Texas, is an open-source code with much of the functionality of the commercial code MatLab. COCO [21] is an open-source process flowsheet simulator, although it is currently only available for running on Microsoft windows.

In the remainder of this short paper, we document some of the open-source MM efforts being led from within the chemical engineering MM community, many members of which have embraced the open source model. More details are provided in the supplementary information (SI), along with other key open source codes for MM (many of which are used within the open source projects, described below). A general review of open source molecular modeling software, with a particular emphasis on visualization, is provided by Pirhadi *et al.*[22].

2. Open Source MM Projects in Chemical Engineering

Here we briefly describe open-source MM projects being led within the chemical engineering community. This list is likely incomplete, as new open-source MM projects are always emerging. Also, many chemical engineering researchers, who are committed to open source ideals are not developing major codes or utilities. Such individuals will post online input files, scripts, plug-ins, etc. that they developed in their own work using a major open-source community code (such as LAMMPS). As just one of many examples, we cite Jeremy Palmer of the University of Houston who provides links on his research web site [23] to scripts and codes used in his recent publications.

2.1. MD Codes

As noted above, there are many MD codes available as open-source, freeware or near-freeware. Open-source codes such as LAMMPS and GROMACS are very efficient on parallel architectures, and, thanks to decades of community development, have many sophisticated features. Thus, developing a new "rival" to these codes is a formidable challenge. It is perhaps not surprising that only one essentially new MD code has emerged from the chemical engineering community, and that is the already mentioned **HOOMD-Blue** [7,8]. HOOMD-Blue is an MPI- and GPU-enabled, general-purpose MD code (with MC capabilities as well), which was initiated by Josh Anderson, who is now part of the Sharon Glotzer group at the University of Michigan. HOOMD-Blue was the first successful open-source MD code written, from inception, to take advantage of many-core GPUs; the latter, while originally developed for accelerating graphics in computer games, were being recognized in the early 2000s for their potential to perform scientific computation at speeds far higher than conventional CPUs, albeit with a more complex programming model. Since essentially all of the world's leadership class supercomputers now come with GPUs on every compute node, HOOMD-Blue is an attractive code to use on such architectures. The central use of GPUs in HOOMD-Blue means that expensive CPU calculations can be quickly calculated on the many cores available in single or multiple GPUs. HOOMD-Blue also features some integration algorithms not supported in other codes, including MC moves on complex hard particles, a particular focus in the Glotzer group. An example of the use of HOOMD-Blue is the very surprising computational discovery that icosahedral quasicrystals, a form of matter that is ordered but not periodic in any direction and has only previously been found in intermetallic compounds, could be formed by self assembly from a liquid phase of a single component fluid interacting via spherically symmetric intermolecular potentials [24]. Additional information on how to access HOOMD-Blue is provided in the Supplementary Information (SI).

2.2. MC Codes

It is in the area of MC simulation that chemical engineers have been particularly active. In the authors' opinion, the reason for this is twofold. First, chemical engineers are often interested in situations where there is some kind of equilibrium between phases – for example, between liquid and vapor phases (relevant to distillation), between two liquid phases (immiscibility), between liquid and solid phases (relevant to crystallization), and between bulk fluid and confined phases (relevant to adsorption and, e.g., the solvent structure between colloidal particles in suspension). Many of the simulation algorithms for such situations are predominantly MC methods (Gibbs ensemble MC [25] - GEMC - for phase equilibrium between fluid phases; Gibbs-Duhem integration [26] – GDI - for fluid-fluid and fluid-solid equilibrium; grand canonical MC [27] – GCMC – for systems in equilibrium with a bulk phase, and so at fixed bulk chemical potential); we should note that GDI can be performed with both MC and MD. Second, chemical engineers have generally been more applications-driven than their counterparts in the physical sciences, and so force fields that reproduce engineering-relevant experimental properties (including phase equilibria and transport properties) over wide ranges of state conditions have been a priority for chemical engineers; in contrast, for example, simulators of biological systems are normally satisfied with force fields that are accurate at physiologically relevant state conditions – i.e., temperatures in the narrow range of 20-40^{o}C and at ambient and near-ambient pressure. Hence chemical engineers have been at the forefront of developing force field for complex systems that are accurate over wider ranges of temperatures and pressures, coupled to codes that have the capability to calculate properties of interest. Some of the force fields used by chemical engineers are not supported by the mainstream open-source MC codes, and hence some researchers have developed their own codes that they have subsequently shared.

Examples of such codes are the open-source MC codes **MCCS Towhee** [9,10] and **Cassandra** [11,12]. Both implement sophisticated configurational bias moves to improve sampling efficiency and algorithms such as GEMC and GCMC. MCCS Towhee originated with the Monte Carlo for Complex Systems (MCCS) code developed in the Ilja Siepmann group at the University of Minnesota. MCCS implemented Siepmann's transferable potentials for phase equilibria (TraPPE) family of force fields (http://chem-siepmann.oit.umn.edu/siepmann/trappe/index.html), which were the first force fields designed to reproduce phase equilibria data. Marcus Martin, a Siepmann student, turned MCCS into the fully open source MCCS Towhee while working at Sandia National Laboratory. A typical use of MCCS Towhee is reported by Teich-McGoldrick *et al*. [28] in their molecular simulation of swelling in smectite clay minerals. GCMC was used to determine water adsorption isotherms at 300 K. Interestingly, this paper also used LAMMPS for MD simulations, and is a good example of the value of open source MM software. Cassandra, developed in the Ed Maginn group at Notre Dame, is a much more recent addition to the open source MM software portfolio, yet is beginning to attract users. An example is Abedini *et al*. [29] who modeled the CO_2 separation characteristics of ionic polyimides (i-PIs) using GROMACS MD simulations to prepare samples of the i-PIs, both neat and in ionic liquid solvents; these configurations were then used in GCMC calculations using Cassandra to compute CO_2 solubilities in the i-PI+ionic liquid mixtures.

RASPA [30,31] is a more specialized open-source MM package, developed primarily in the Randy Snurr group at Northwestern University, and more recently with the involvement of David Dubbeldam of the University of Amsterdam, Sofia Calero of the University Pablo de Olavide in Sevilla, Spain, and Thijs Vlugt of the Delft University of Technology. RAPSA is designed for simulating adsorption (via GCMC) and diffusion (via MD) of molecules in flexible nanoporous materials. RASPA has been used in the computational discovery of new adsorbents for gas storage, an example of which is Moghadam *et al*.[32]. In this work, computational screening (primarily by GCMC) is performed on 2932 metal–organic framework materials to assess their potential for oxygen storage. One of the outcomes of the study is the identification of UMCM-152, which delivers 22.5% more oxygen storage than the previous record holder.

A relative newcomer to the suite of available open-source MM packages is the GPU-optimized Monte Carlo code **GOMC** [33,34]. Developed primarily by Jeff Potoff at Wayne State University, GOMC is a general purpose MC code, similar in capability to Cassandra and MCCS Towhee, but it has been written to be optimized on GPUs. In a sense, then, it is the MC analog of HOOMD-Blue in the MD realm. GOMC was used by Soroush Barhaghi and Potoff [35] to calculate free energies of transfer of n-alkanes from vapor into various organic solvents using isobaric-isothermal GEMC simulations.

No discussion of open source MM modeling in chemical engineering would be complete without including **Etomica** [36][37]**,** developed by David Kofke and Andrew Shultz at the University at Buffalo. Etomica differs from the codes described above because its primary focus is on education, to teach people about molecular simulation, but also more generally to provide insight into the molecular origin of many properties, such as pressure (demonstrated through teaching module that interactively uses MD to simulate gas molecules in a piston-cylinder apparatus). Etomica modules [38] are written in Java, and can run in browser windows as well as from the desktop. Its object-oriented structure and ability to run simulations interactively makes it suitable for quick adaptation and testing of new molecular simulation methods, a key focus in the Kofke research group. Etomica originated in the mid-1990s with the goal of developing interactive illustrations for a web-based textbook on molecular simulation. The limitations of the web in the mid-1990s prevented the web-based textbook from reaching its ambitious goals, but Etomica has continued as a stand-alone, award-winning project supported through several National Science

Foundation (NSF) grants. Etomica is used by many chemical engineering faculty who teach simulation courses since, e.g., one can illustrate MD interactively, including viewing the evolution of structural and thermodynamic properties; variables, including ensembles, can be easily changed and the consequences understood. Etomica also has unique capabilities, unavailable in any other code, such as evaluation of virial coefficients by Mayer sampling Monte Carlo [39].

Additional information about these codes is contained in the SI.

2.3. Integrative Environments

As detailed in Section 2 of the SI, there are a number of open-source MM tools that are not simulation codes *per se*, but provide essential functionality for being able in practice to perform research-caliber molecular simulations. These include system set up tools (such as PACKMOL [40]) and trajectory analysis tools (such as MDTraj [41]). These and additional set up and trajectory analysis tools are described in the SI, sections 2.2 and 2.5 respectively. Within chemical engineering, a large effort is underway to build a robust Python-based, open-source integrated software framework for performing simulations of soft matter systems with the goal of implementing best practices and enabling reproducibility**,** known as the Molecular Simulation Design Framework (**MoSDeF**). With the support of several NSF grants, MoSDeF was originally developed by Peter Cummings, Clare M^{c}Cabe and Chris Iacovella and their students in chemical engineering at Vanderbilt University, in collaboration with software engineering faculty researchers (Akos Ledeczi and Gabor Karsai) and their groups from the Institute for Software Integrated Systems (http://www.isis.vanderbilt.edu/), also at Vanderbilt, supported by National Science Foundation grants (#1047828 and #1535150). MoSDeF provides a core foundation and includes libraries for programmatic system construction (mBuild) [42,43], for encoding force field usage rules and their application (foyer) [44–46], and interfaces easily with the **Signac-flow** framework [47,48], developed in the Glotzer group at the University of Michigan, for execution workflow management. MoSDeF currently supports LAMMPS, GROMACS and HOOMD-Blue by automatically generating the required chemical information files for these codes in the required engine-specific syntax. In particular, since all the utilities are fully scriptable, MoSDeF supports screening of soft material systems. Scientifically, MoSDeF has been use to enable screening studies of monolayer lubrication [39,49] and diffusivity of ionic liquids in many organic solvents [50]. MoSDeF is also designed to support reproducibility, by automating many steps. The details of simulations studies performed using MoSDeF (scripts, input files, system configurations, etc.) can be distributed via an open source repository (e.g., GitHub) such that other researchers can duplicate a published simulation in full detail. A recently awarded NSF grant (#1835874) funds a collaboration between Vanderbilt and the universities of Michigan, Minnesota, Notre Dame, Delaware, and Houston, Wayne State University and Boise State University to dramatically expand the capabilities of MoSDeF, in part by expanding its support to the additional codes CP2K (see SI), Cassandra, and GOMC, among others, and through the addition of many workflows for specific simulation applications.

3. Conclusions

The molecular modeling community in chemical engineering has broadly embraced the open source model for software creation and distribution, as well as utilizing many of the excellent open-source codes and utilities developed outside the field. Progress in the field overall has been facilitated by this collaborative spirit, and we expect this to continue into the future. It also creates opportunities for interdisciplinary collaborations, as users from outside the chemical engineering community make use of the open source codes and tools developed within it.

4.Acknowledgments

The authors efforts in preparing this article were supported by a National Science Foundation grant, OAC-1835874. We thank Clare McCabe, David Kofke, Matthew Thompson, Ray Matsumoto and Alex Yang for their critical comments on an earlier draft of the manuscript. We thank former students Christoph Klein (now with Counsyl) and Andrew Summers (now with Enthought), who played major roles in shaping MoSDeF, and we acknowledge Chris Iacovella's central role in MoSDeF's inception, its present and its future.

Disclosure

The authors declare no conflict of interest.

REFERENCES

[1] J.K. Nørskov, T. Bligaard, J. Rossmeisl, C.H. Christensen, Towards the computational design of solid catalysts., Nat. Chem. 1 (2009) 37–46. doi:10.1038/nchem.121.

[2] A. Asthagiri, M.J. Janik, eds., Computational Catalysis, Royal Society of Chemistry, Cambridge, 2013. doi:10.1039/9781849734905.

[3] S. Plimpton, Fast Parallel Algorithms for Short-Range Molecular Dynamics, J. Comput. Phys. 117 (1995) 1–19. doi:10.1006/jcph.1995.1039.

•• [4] LAMMPS Molecular Dynamics Simulator. https://lammps.sandia.gov.

Home page for the open-source LAMMPS molecular dynamics code. See ref. [3] above.

[5] H.J.C. Berendsen, D. van der Spoel, R. van Drunen, GROMACS: A message-passing parallel molecular dynamics implementation, Comput. Phys. Commun. 91 (1995) 43–56. doi:10.1016/0010-4655(95)00042-E.

•• [6] Gromacs - Gromacs. http://www.gromacs.org/ (accessed January 27, 2019).

Home page for the open-source GROMACS molecular dynamics code. See ref. [5] above.

[7] J.A. Anderson, C.D. Lorenz, A. Travesset, General purpose molecular dynamics simulations fully implemented on graphics processing units, J. Comput. Phys. 227 (2008) 5342–5359. doi:10.1016/j.jcp.2008.01.047.

•• [8] HOOMD-blue - Home. https://glotzerlab.engin.umich.edu/hoomd-blue/ (accessed January 27, 2019).

Home page for the open-source HOOMD-Blue molecular simulation code. See ref. [7] above.

[9] M.G. Martin, MCCCS Towhee: a tool for Monte Carlo molecular simulation, Mol. Simul. 39 (2013) 1212–1222. doi:10.1080/08927022.2013.828208.

• [10] Towhee. http://towhee.sourceforge.net (accessed December 6, 2018).

Home page for the open-source MCCS Towhee molecular simulation code. See ref. [9] above.

•• [11]J.K. Shah, E. Marin-Rimoldi, R.G. Mullen, B.P. Keene, S. Khan, A.S. Paluch, N. Rai, L.L. Romanielo, T.W. Rosch, B. Yoo, E.J. Maginn, Cassandra: An open source Monte Carlo package for molecular simulation, J. Comput. Chem. 38 (2017) 1727–1739. doi:10.1002/jcc.24807.

Key reference for the Cassandra open-source Monte Carlo simulation code. See home page below, ref. [12].

[12] Cassandra Home Page. https://cassandra.nd.edu/ (accessed January 10, 2019).

[13] B.R. Brooks, C.L. Brooks, A.D. Mackerell, L. Nilsson, R.J. Petrella, B. Roux, Y. Won,
• G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A.R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R.W. Pastor, C.B. Post, J.Z. Pu, M. Schaefer, B. Tidor, R.M. Venable, H.L. Woodcock, X. Wu, W. Yang, D.M. York, M. Karplus, CHARMM: The
• biomolecular simulation program, J. Comput. Chem. 30 (2009) 1545–1614. doi:10.1002/jcc.21287.

[14] Home - CHARMM. https://www.charmm.org/ (accessed January 28, 2019).

[15] D.A. Case, T.E. Cheatham, T. Darden, H. Gohlke, R. Luo, K.M. Merz, A. Onufriev, C. Simmerling, B. Wang, R.J. Woods, The Amber biomolecular simulation programs, J. Comput. Chem. 26 (2005) 1668–1688. doi:10.1002/jcc.20290.

[16] NAMD Scalable Molecular Dynamics. http://www.ks.uiuc.edu/Research/namd/ (accessed January 28, 2019).

[17] J.C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R.D. Skeel, L. Kalé, K. Schulten, Scalable molecular dynamics with NAMD, J. Comput. Chem. 26 (2005) 1781–1802. doi:10.1002/jcc.20289.

[18] M. Baker, 1,500 scientists lift the lid on reproducibility, Nature. 533 (2016) 452–454. doi:10.1038/533452a.
[19] L.A. Barba, The hard road to reproducibility, Science. 354 (2016) 142–142. doi:10.1126/science.354.6308.142.
[20] GNU Octave. https://www.gnu.org/software/octave/ (accessed January 28, 2019).
[21] COCO - the CAPE-OPEN to CAPE-OPEN simulator. https://www.cocosimulator.org/index.html (accessed January 28, 2019).
[22] S. Pirhadi, J. Sunseri, D.R. Koes, Open source molecular modeling, J. Mol. Graph. Model. 69 (2016) 127–143. doi:10.1016/j.jmgm.2016.07.008.
[23] Palmer Group@UH: Condensed Matter Theory and Simulation. http://palmer.chee.uh.edu/.
[24] M. Engel, P.F. Damasceno, C.L. Phillips, S.C. Glotzer, Computational self-assembly of a one-component icosahedral quasicrystal, Nat. Mater. 14 (2015) 109–116. doi:10.1038/nmat4152.
[25] A.Z. Panagiotopoulos, Direct determination of phase coexistence properties of fluids by Monte Carlo simulation in a new ensemble, Mol. Phys. 61 (1987) 813–826. doi:10.1080/00268978700101491.
[26] D.A. Kofke, Gibbs-Duhem integration: a new method for direct evaluation of phase coexistence by molecular simulation, Mol. Phys. 78 (1993) 1331–1336. doi:10.1080/00268979300100881.
[27] L. Rowley, D. Nicholson, N. Parsonage, Monte Carlo grand canonical ensemble calculation in a gas-liquid transition region for 12-6 Argon, J. Comput. Phys. 17 (1975) 401–414. doi:10.1016/0021-9991(75)90042-X.
[28] S.L. Teich-McGoldrick, J.A. Greathouse, C.F. Jové-Colón, R.T. Cygan, Swelling Properties of Montmorillonite and Beidellite Clay Minerals from Molecular Simulation: Comparison of Temperature, Interlayer Cation, and Charge Location Effects, J. Phys. Chem. C. 119 (2015) 20880–20891. doi:10.1021/acs.jpcc.5b03253.
[29] A. Abedini, E. Crabtree, J.E. Bara, C.H. Turner, Molecular analysis of selective gas adsorption within composites of ionic polyimides and ionic liquids as gas separation membranes, Chem. Phys. 516 (2019) 71–83. doi:10.1016/j.chemphys.2018.08.039.
[30] D. Dubbeldam, S. Calero, D.E. Ellis, R.Q. Snurr, RASPA: molecular simulation software for adsorption and diffusion in flexible nanoporous materials, Mol. Simul. 42 (2016) 81–101. doi:10.1080/08927022.2015.1010082.
• [31] RASPA: Molecular Simulation Software for Adsorption and Diffusion in Flexible Nanoporous Materials. https://www.iraspa.org/RASPA/.
Home page for the open-source RASPA molecular simulation code. See ref. [30] above.
• [32]P.Z. Moghadam, T. Islamoglu, S. Goswami, J. Exley, M. Fantham, C.F. Kaminski, R.Q. Snurr, O.K. Farha, D. Fairen-Jimenez, Computer-aided discovery of a metal–organic framework with superior oxygen uptake, Nat. Commun. 9 (2018) 1378. doi:10.1038/s41467-018-03892-8.
Use of RASPA for computational screening of metal-organic frameworks for oxygen storage.
[33] Y. Nejahi, M. Soroush Barhaghi, J. Mick, B. Jackman, K. Rushaidat, Y. Li, L. Schwiebert, J. Potoff, GOMC: GPU Optimized Monte Carlo for the simulation of phase equilibria and physical properties of complex fluids, SoftwareX. 9 (2019) 20–27. doi:10.1016/j.softx.2018.11.005.
• [34] GOMC: GPU optimized Monte Carlo molecular simulation engine. http://gomc.eng.wayne.edu/.
Home page for the open-source GOMC MC simulation code. See ref. [333] above.
[35]M. Soroush Barhaghi, J.J. Potoff, Prediction of phase equilibria and Gibbs free energies of transfer using molecular exchange Monte Carlo in the Gibbs ensemble,

Fluid Phase Equilib. 486 (2019) 106–118. doi:10.1016/j.fluid.2018.12.032.
[36] A.J. Schultz, D.A. Kofke, Etomica : An object-oriented framework for molecular simulation, J. Comput. Chem. 36 (2015) 573–583. doi:10.1002/jcc.23823.
[37] Etomica : Molecular Simulation API. http://www.etomica.org/index.html (accessed January 30, 2019).
•• [38] Modules - Wiketomica. http://rheneas.eng.buffalo.edu/wiki/Modules (accessed January 31, 2019).
Home page for Etomica-based educational modules, illustrating physical and chemical phenomena by molecular simulation. See ref. [36] above.
[39] A.Z. Summers, C.R. Iacovella, P.T. Cummings, C. McCabe, Examining Structure-Property Relationships in Lubricating Monolayer Films through Molecular Dynamics Screening, ACS Nano. submitted (2019).
[40] L. Martínez, R. Andrade, E.G. Birgin, J.M. Martínez, PACKMOL: A package for building initial configurations for molecular dynamics simulations, J. Comput. Chem. 30 (2009) 2157–2164. doi:10.1002/jcc.21224.
[41] R.T. McGibbon, K.A. Beauchamp, M.P. Harrigan, C. Klein, J.M. Swails, C.X. Hernández, C.R. Schwantes, L.-P. Wang, T.J. Lane, V.S. Pande, MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories, Biophys. J. 109 (2015) 1528–1532. doi:10.1016/j.bpj.2015.08.015.
[42] C. Klein, J. Sallai, T.J. Jones, C.R. Iacovella, C. McCabe, P.T. Cummings, A Hierarchical, Component Based Approach to Screening Properties of Soft Matter, in: R.Q. Snurr, C.S. Adjiman, D.A. Kofke (Eds.), Found. Mol. Model. Simulation. Mol. Model. Simul. (Applications Perspect., Springer, Singapore, Singapore, 2016: pp. 79–92. doi:10.1007/978-981-10-1128-3_5.
• [43] mBuild Github repository. https://github.com/mosdef-hub/mbuild (accessed April 17, 2018).
Repository for the mBuild code used to build molecular systems.
• [44] Foyer Github repository. https://github.com/mosdef-hub/foyer (accessed April 17, 2018).
Repository for the foyer code used to atom-type molecular systems.
[45]C.R. Iacovella, J. Sallai, C. Klein, T. Ma, Idea Paper : Development of a Software Framework for Formalizing Forcefield Atom-Typing for Molecular Simulation, in: 4th Work. Sustain. Softw. Sci. Pract. Exp., 2016.
[46] C. Klein, A.Z. Summers, M.W. Thompson, J. Gilmer, C. McCabe, P.T. Cummings, J. Sallai, C.R. Iacovella, Formalizing Atom-typing and the Dissemination of Force Fields with Foyer, (2018). http://arxiv.org/abs/1812.06779 (accessed January 25, 2019).
• [47] C.S. Adorf, P.M. Dodd, V. Ramasubramani, S.C. Glotzer, Simple data and workflow management with the signac framework, Comput. Mater. Sci. 146 (2018) 220–229. doi:10.1016/j.commatsci.2018.01.035.
Introduction to the signac workflow management tool, available at website in Ref. [48].
[48] Signac documentation. https://signac.readthedocs.io/en/latest/ (accessed April 17, 2018).
[49] C. Klein, J. Sallai, T.J. Jones, C.R. Iacovella, C. McCabe, P.T. Cummings, A Hierarchical, Component Based Approach to Screening Properties of Soft Matter, in: R. Snurr, C. Adjiman, D.A. Kofke (Eds.), Found. Mol. Model. Simulation. Mol. Model. Simul. (Applications Perspect., Springer, Singapore, 2016: pp. 79–92. doi:10.1007/978-981-10-1128-3_5.
• [50] M.W. Thompson, R. Matsumoto, R.L. Sacci, N.C. Sanders, P.T. Cummings, Scalable Screening of Soft Matter: Case Study of Mixtures of Ionic Liquids and Organic Solvents, J. Phys. Chem. B. (2019). doi:10.1021/acs.jpcb.8b11527.
Example of the use of MoSDeF and signac to perform computational screening, in this

case of the transport properties of an ionic liquid in organic solvents.