Best Practices for Developing Monte Carlo Methodologies in Molecular Simulations [Article v1.0]

Harold W. Hatch[1*], **David S. Corti**[2], **David A. Kofke**[3], **Vincent K. Shen**[1]

[1]Chemical Informatics Research Group, Chemical Sciences Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8380, USA; [2]Purdue University, Davidson School of Chemical Engineering, West Lafayette, IN 47907, USA; [3]University at Buffalo, Department of Chemical and Biological Engineering, Buffalo, NY 14260, USA

This LiveCoMS document is maintained online on GitHub at https://github.com/usnistgov/best-practices-mc; to provide feedback, suggestions, or help improve it, please visit the GitHub repository and participate via the issue tracker.

This version dated August 4, 2025

Abstract Although Monte Carlo (MC) is a very powerful molecular simulation method in statistical mechanics, the development and application of novel MC trials to optimize sampling in complex systems is hindered by the difficulty in deriving their acceptance probabilities. We present a checklist approach to deriving acceptance probabilities, and apply this approach to a variety of trials in the canonical, isothermal-isobaric, grand-canonical, semi-grand canonical and Gibbs ensembles. The ideal gas is then shown to be a useful test case to compare the results of simulations with those from theoretical expectations, providing a computational benchmark that can be easily and rapidly implemented for determining if the acceptance criteria were derived correctly. More complex models and trials are also considered with this checklist approach, including configurational bias, cavity bias, energy bias, aggregation volume bias, dual-cut configurational bias and rigid cluster moves. The result is a framework designed to help researchers implement and test specialized MC trials that expand the model complexity and length scales currently available in open-source MC molecular simulation software. Sample code is also provided in the GitHub repository [https://github.com/usnistgov/best-practices-mc].

***For correspondence:**
harold.hatch@nist.gov (HWH)

1 Introduction

Monte Carlo (MC) molecular simulation has for the last seventy years provided a very powerful tool for statistical mechanics. Beginning with Metropolis MC of two-dimensional hard disks in the canonical (*NVT*) ensemble [1, 2], MC expanded to include chains [3], the isothermal-isobaric (*NPT*) ensemble [4], the grand-canonical ensemble (μVT) [5, 6], the semi-grand-canonical (semi-μVT) ensemble [7] and the Gibbs ensemble [8]. Specialized techniques were developed to improve sampling of the various relevant microstates, such as configurational bias (CB) [3, 9], expanded ensembles [10] and growth expanded ensemble methods [11]. A variety of MC methods were also developed to simulate phase behavior and equations of state, such as Gibbs-Duhem [12], Wang-Landau [13], transition-matrix [14, 15], Mayer-sampling [16] and continuous fractional component [17].

The usefulness of MC molecular simulations is also evidenced by a number of available software packages, including but not limited to Towhee [18], BOSS [19], MCPRO [19], Etomica [20], MUSIC [21], DL_MONTE [22], Cassandra [23], Faunus [24], RASPA [25], GOMC [26], SIMONA [27], ESPResSo [28], MCCCS-MN [29] and FEASST ~~[30?]~~[30]. However, analysis of citation data shows there are many more widely used molecular dynamics (MD) [31] software than MC software [32]. Comparison between MC and MD may help in understanding the reasons for the current popularity of MD and the need for a more accessible approach to developing and implementing MC.

The major differences between MC and MD is that MD uses the physical laws of motion to sample configurations of the system, while MC does not have such a requirement [33, 34]. Visualization of MD trajectories is more conceptually simple for a user to understand and present to a general audience. MD is also capable of computing time-dependent dynamic quantities such as diffusion or viscosity, while non-physical trial moves typically confine MC to only equilibrium thermodynamic properties. These benefits in MD come with a few trade-offs relative to MC. MD simulations may not be feasible for the required timescale of a physical process of interest. In contrast, MC is not constrained to any physical laws of motion, and because of this, MC may be quite powerful because “non-physical" trial moves may be performed, which significantly speed up the equilibration of the system or free a system trapped in a local free energy minimum, thus making it a powerful tool for determining equilibrium properties [34–36].

In MD, the sampling of collective motion with many bonded potentials and holonomic constraints is a general, broadly applicable approach to simulate a variety of complex molecules [37, 38]. In comparison, MC requires specialized trials that are chosen depending upon the molecules being simulated. Providing access to this specialization while maintaining broad applicability increases the software complexity of MC codes relative to MD. Still, there are some common features among different trials that MC codes can exploit to provide a robust, extensible framework.

Invention of new MC trials is an avenue for the developer to apply imagination and creativity to make simulations much more efficient, or even make "impossible" calculations newly feasible. For example, recent methodological MC developments allow proteins to be modeled as a single anisotropic site with docking trials [39] and while retaining atomistic resolution [40]. While there is a broad scope for the invention of MC trials, constraints exist that ensure they provide correct sampling of the ensemble. Formulating a systematic approach for deriving computationally efficient MC trials while adhering to these constraints is the challenge we hope to address in this article.

In the simplest cases of an *NVT* simulation of hard spheres or Lennard-Jones particles with periodic boundaries, MC is easier to implement than MD, which requires forces, velocities, integrators and thermostats. Instead, MC uses a random number generator to propose and decide acceptance of configuration changes according to the expected probability distributions. Also, as opposed to MD, MC is capable of simulating continuous, discontinuous and mixed potentials with the same basic algorithm. MD simulations may also struggle to reproduce the exact same trajectory as a previous simulation because the smallest differences in initial conditions lead to divergence via the Lyapunov instability [41], while exact reproduction of MC with a given random number generator seed may be simpler than in MD.

Hybrid MC and MD methods may combine the strengths of both methods [35, 42–47]. MD techniques have also been influenced by similar efforts in MC methods, including replica exchange [48–51], thermostats and barostats [52] and metadynamics [53]. Many of these MD techniques lead to non-physical motion, which complicates the computation of dynamic quantities [54]. From this perspective, some of the most popular uses of MD resemble MC while utilizing the physical laws of motion as an effective microcanonical ensemble sampling algorithm. Hence, improvements in the efficiency of MC simulations may also apply to hybrid MD/MC simulations.

Because the position of all particles are updated simultaneously in MD, parallelization with domain decomposition [55] scales with the number of processors better than single-particle position updates in the typical MC approach. MC is not constrained to single-particle perturbations and may include collective motion of all particles with MC trials [56–59] or hybrid MD. Although MD provides many more examples of the largest molecular simulations ever conducted, MC in principle has more available parallelization strategies than MD. Both MC and MD are amenable to domain decomposition [60, 61] and replica exchange [48], but MC may also parallelize with CB [62, 63], waste recycling [64], event-chain [65] and prefetching [66–68].

Although parallelization of very large systems is seen in MD more than in MC, another effective approach is to circumvent the need for large systems altogether. For example, while MD simulations of phase coexistence often require a system size many times larger than the interface, Gibbs ensemble MC avoids explicitly simulating phase boundaries [8], which enables the use of smaller system sizes. Similarly, μVT simulations remove the spurious effect of the constraint on the number of particles in self-assembling systems [69, 70], while simulations with a constant number of particles require larger system sizes to ensure that the effect of this non-physical constraint is negligible. Sampling with μVT flat-histogram methods improves the efficiency of simulations of phase separation, self-assembly and adsorption, relative to *NVT* sampling, even when insertion and deletion acceptances are seemingly small [71].

One of the goals of this article is to make it easier for researchers to derive novel MC trials that are needed to improve the sampling of their models under the thermodynamic conditions of interest. We provide a framework and checklist for how to derive the acceptance criteria for a wide variety of MC trials, while also demonstrating how to test the derivations. We explain a checklist approach for deriving acceptance criteria that identifies probabilities of every step in the MC trial implementation where random number generators are involved. Our hope is that this checklist approach is less prone to error than a purely theoretical one that is more decoupled from the implementation. For example, in the semi-μVT ensemble, there is a clear difference in the acceptance probability based on how the particles are chosen [7]. And in μVT trial derivations, the checklist approach, where particles are distinguishable, is shown to give exactly the same acceptance criteria as the purely theoretical approach [34], where particles are considered indistinguishable.

A variety of existing MC trials are derived to demonstrate the checklist approach to obtaining the acceptance criteria. Because the vast majority of MC trials derived in this article were published previously, interested readers should also reference those original papers. At least three aspects of the derivations in this article may not be available in the literature as far as we are aware. The first is the derivation of insertions and deletions without the requirement of indistinguishable particles. The second is the use of (dual-cut) CB (DCCB) [72] for *NVT* displacements. And the third is a slightly different expression for cavity bias.

Computational tests using an ideal gas are also demonstrated to be an important first validation step for any proposed MC trial [73]. They are fast and easily implemented, requiring in nearly all cases only a few dozen lines of code (which are provided in this article). In addition, the results of these ideal gas simulations can be readily compared to known theoretical results. With an ideal gas, all terms in the derived acceptance probabilities may be tested except for those depending on the potential energy.

The Living Journal of Computational Molecular Science (LiveCoMS) is an ideal forum for this work because many in the MC community may comment on or contribute to LiveCoMS articles. It is our hope that this document is a useful compendium that contains numerous MC trials which are otherwise derived in different ways and scattered throughout many research journals. In addition, any errors found in this article may be immediately commented upon by the community and addressed.

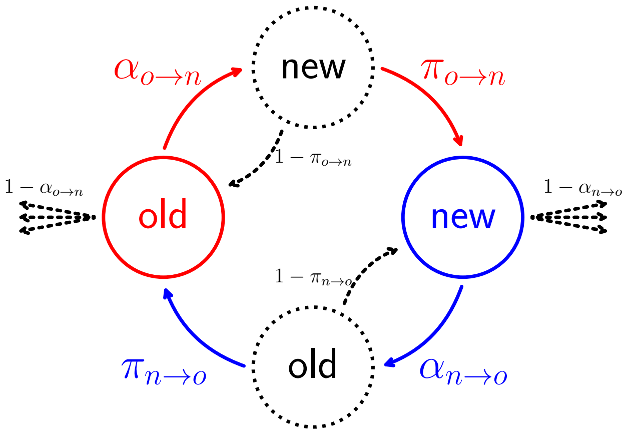

Figure 1. Probabilities for all possible transitions between two states, old and new, are illustrated. Proposed configurations and other possible transitions are shown with black dashed circles and arrows, respectively. Detailed balance is achieved when, for every pair of states, the probability to be in one of the states and then to accept a transition into another state is equal to the probability to be in that other state and then to accept the reverse transition.

This article is organized as follows. In Section 2, the general equation for an MC trial acceptance probability is derived using detailed balance, resulting in the product of ratios of microstate and transition probabilities in the new and old microstates. A checklist approach for deriving an MC trial acceptance criterion is then developed in Section 3. After the description of the variables required to define a configuration are presented in Section 4, unbiased trials are derived, using the checklist approach, in the *NVT*, *NPT*, μVT, semi-μVT and Gibbs ensembles in Sections 5 to 9, respectively. An MC trial is considered unbiased when the transition probabilities do not depend upon the configuration. Each of these unbiased MC trials is also validated with an ideal-gas simulation provided in this article, which serves as a test of the acceptance criteria. Afterward, biased μVT MC trials are derived using the checklist approach in Section 10, including CB, DCCB, cavity bias, energy bias and aggregation volume bias (AVB) [74]. Then, in Section 11, biased *NVT* trials are derived. Finally, implementation issues typically encountered in MC simulations are summarized in Section 12, followed by conclusions in Section 13.

2 Metropolis MC acceptance probability and detailed balance

In this section, the probability of accepting an attempted MC trial move, or the Metropolis MC trial acceptance probability, denoted by χ, is derived [1, 7, 33, 34, 75]. Ideally, χ should be close to 1, so that the trial move in either direction has a high likelihood of being accepted. The formulation of trial moves is often directed toward this end, while also promoting broader sampling of microstates.

The derivation of χ in this article requires the condition of detailed balance, shown in Fig. 1, for a transition from an old microstate, *o* to a new microstate, *n*,

$$\Pi_o \alpha_{o\to n} \pi_{o\to n} = \Pi_n \alpha_{n\to o} \pi_{n\to o}, \tag{1}$$

where Π_o and Π_n are the probabilities of being in the old and new microstates, respectively, $\alpha_{o\to n}$ and $\alpha_{n\to o}$ are the probabilities of attempting a transition from the old to the new microstate, and vice versa, and $\pi_{o\to n}$ and $\pi_{n\to o}$ are the probabilities of accepting those transitions. Symbols in this article are chosen to follow closely with Ref. [34], but note the different definition of π. Detailed balance is a sufficient, but not necessary, condition for ergodic sampling [76]. For simplicity, we use detailed balance throughout the derivations of this article. While the microstate probabilities, Π_o and Π_n, are determined solely by the statistical mechanical ensemble, the transition probabilities $\alpha_{o\to n}$ and $\alpha_{n\to o}$ depend upon how the MC trial move is implemented. Thus, Eq. 1 provides the route to obtain the trial acceptance probability after the ensemble, trial implementation and acceptance criterion are defined. Here, *trial implementation* encompasses a great variety of potential moves, such as displacements, volume changes, temperature changes, particle insertions and deletions, alchemical transformations, and more.

In a typical MC simulation, several types of trial moves are employed, with one chosen at random at the beginning of each MC trial. It is possible that two or more of these trials are capable of producing the same transition between two microstates, albeit in different ways, and with different probabilities. At first, one might think that the transition probability between two microstates must account for all such ways when determining the acceptance probability χ to satisfy detailed balance. Fortunately, this is not the case. Instead, it is sufficient that each transition method by itself satisfy detailed balance. It is easy to show that overall detailed balance is satisfied if this “super-detailed balance" condition is met [64]. In this article, each reversible pair of MC trials is derived as if the pair are the only trials implemented in the simulation. In special cases, a reversible pair of trials can be obtained by simply repeating the same trial again. In this special case, the transition probabilities are symmetric.

In general, the Metropolis acceptance criterion is given by,

$$\pi_{o\to n} = \min(1, \chi), \tag{2}$$

where χ is the Metropolis acceptance probability. The reverse acceptance probability is $\pi_{n\to o} = \min(1, 1/\chi)$. The ratio of π is then simplified as,

$$\frac{\pi_{o\to n}}{\pi_{n\to o}} = \frac{\min(1,\chi)}{\min(1,1/\chi)} = \chi, \tag{3}$$

by considering each of the three possibilities, $\chi < 1$, $\chi > 1$ and $\chi = 1$, separately. While the Metropolis acceptance criterion is used throughout this article, other alternatives are possible while maintaining detailed balance.[77]

Eq. 1 may be rearranged to solve for the Metropolis acceptance probability using Eq. 3,

$$\chi = \left(\frac{\Pi_n}{\Pi_o}\right)\frac{\alpha_{n\to o}}{\alpha_{o\to n}}, \tag{4}$$

where χ is expressed as the product of two ratios that we refer to throughout this article.

The first ratio in Eq. 4 is the ratio of microstate probabilities, Π_n/Π_o, that may be derived from a given statistical mechanical ensemble, or computed at each step of a converging MC simulation, even if the partition function is not known. The second ratio may be derived for a given MC trial in a systematic way that is tied very closely to the implementation of the trial. An important caveat is the ratio Π_n/Π_o must be expressed in terms of the same set of variables that change during the trial.

3 Best Practices Checklist for Deriving MC Trial Acceptance

The checklist to obtain MC trial acceptance probabilities in a way that minimizes the possibility of error is provided on the next page. The acceptance probabilities are derived in a step-by-step manner that is closely related to the simulation algorithms. By starting from the list of instructions to fully describe a forward and reverse trial, a researcher is able to obtain straightforwardly the required transition probabilities, and identify cases where irreversible changes must be rejected, in a systematic fashion. We apply this checklist in subsequent sections to derive a variety of unbiased and biased MC trials in various ensembles. This process of generating the needed acceptance probabilities was inspired by and built upon a set of MC lecture notes [78] and MC ideal gas simulation tests of various theoretically-based ensemble predictions [73].

4 Description of a configuration with particles, coordinates and periodic boundaries

In this article, a configuration refers to the particles, coordinates and periodic boundaries. For an *n*-component system, the total number of particles, $N = \sum_i^n N_i$, where N_i is the number of particles of type *i*. In bulk systems, there are no interactions of the particles with boundaries. To avoid spurious

BEST PRACTICES CHECKLIST FOR DERIVING MC TRIAL ACCEPTANCE

□ **Write the MC trial algorithm as an ordered list of all steps required to fully describe both the forward and reverse trials.**

□ **Check detailed balance.**

- □ Ensure all forward trial steps transition from the old state to a new state.
- □ Consider many different cases, including edge cases such as the largest or smallest values.
- □ Ensure all steps conducted in the reverse trial are capable of returning from the new state to the exact same old state where the forward trial began.

□ **Determine if each step is deterministic, stochastic or needs to be broken into multiple steps.**

- □ Stochastic steps may choose a particle, position, volume or model parameter.
- □ Stochastic steps make one choice from multiple options.
- □ Stochastic steps often use a random number generator (seeded for reproducibility).
- □ Deterministic steps make no choices.
- □ If more than one choice is made in one step, break that step into multiple steps.

□ **Determine the probability for each step.**

- □ Determine a probability for choosing a discrete variable (E.g. if the step requires choosing a particle from among N particles, then the probability is $1/N$).
- □ Determine a probability for choosing a continuous variable (E.g., if the step requires choosing a position in a volume V, then the probability is $d\mathbf{r}/V$).
- □ If the step is deterministic, then the probability is 1.
- □ Determine the forward trial probability, $\alpha_{o \to n}$, from the old to the new state.
- □ Determine the reverse trial probability, $\alpha_{n \to o}$, from the new to the old state.
- □ Obtain the total probability of the forward trial as the product of the forward probabilities for each step.
- □ Repeat with the previous step, for the total reverse probability.

□ **Determine the ensemble of the trial.**

- □ List all possible variables that could change during each step (Note if the particle positions, number of particles, volume or model parameters change).
- □ From the list of all variables that could change, determine the ensemble.
- □ Use NVT ensemble microstate probabilities if only particle positions change.
- □ Use NPT ensemble microstate probabilities if only particle positions and volume change.
- □ Use (semi-)μVT ensemble microstate probabilities if the number of particles changes.
- □ Use Gibbs ensemble microstate probabilities if multiple systems are coupled.
- □ Use Expanded ensemble microstate probabilities if temperature or model parameter change (beyond the scope of this article).

□ **Use Eq. 4 to obtain the trial acceptance probability.**

- □ The ensemble of the trial determines the ratio of microstate probabilities.
- □ The probabilities of each step of the trial implementation determines the ratio of the forward and reverse transition probabilities.

□ **Test the implementation of the trial acceptance probability.**

- □ Verify that the acceptance probability leads to simulation results that agree with theory and trusted sources.
- □ Start with the simplest possible simulations that are fast, such as ideal gases with tractable theoretical expressions.

□ **Document your derivation or software implementation for later reference.**

- □ Include documentation and references for future readers (e.g., yourself) to understand the derivation.

effects of walls or a container, and thereby minimizing the size of the simulated system needed to model the bulk, periodic boundary conditions are imposed such that particles interact with their nearest neighboring image [33]. To simplify the notation in the following derivations, the periodic boundaries align with the Cartesian axes to form a cuboid of D linear dimensions L_d, where the volume, V, is given by $V = \prod_d^D L_d$.

The position of each particle is represented by a single vector coordinate $\mathbf{r}$, and all internal and/or orientational coordinates of the particle are represented by a single vector $\boldsymbol{\omega}$. A particle may be composed of multiple interaction sites that are mutually bound, flexibly or rigidly or both, and the site-site interactions could be orientationally-dependent ~~. But if~~ (e.g., point dipoles). The orientation of a particle might be given implicitly (not part of $\boldsymbol{\omega}$) via the Cartesian coordinates of its interaction sites. Alternatively, it may be specified explicitly via an appropriate orientation vector, with the location of the interaction sites (if a multi-site particle) constructed on-the-fly from the position, orientation, and internal coordinates. In a 3-dimensional space, the orientation coordinate for a cylindrically-symmetric object requires just two independent values, e.g., a unit vector describing the cylindrical axis direction. For particles with no such symmetry, three independent values are needed to specify the orientation. Euler angles are sometimes invoked for this, but they are not particularly useful in the context of molecular simulation. A better choice is a quaternion, which is a normalized, four-element vector [79, 80].

If the particle is not isotropic or rigid, the derivations in this article leave a placeholder variable for the orientation contributions that are not further detailed and are beyond the scope of this article.

5 Unbiased canonical ensemble (*NVT*) trials

Canonical-ensemble trials are defined as trials that do not change the total number of particles of each type, N_i, the volume, V, or the temperature, T, and unbiased trials have transition probabilities that do not depend upon the configuration (e.g., not configurational bias).

In order to obtain trial acceptance in the *NVT* ensemble, first the ratio of microstate probabilities (i.e., the first ratio in Eq. 4), the probabilities of each microstate must be known proportionally with respect to the variables that change during the trial (e.g., only the coordinates, $\mathbf{r}$). The probability of a microstate in the *NVT* ensemble is proportional to [33, 34]

$$\Pi\, \mathrm{d}\mathbf{r}^N \mathrm{d}\boldsymbol{\omega}^N \propto e^{-\beta U} \mathrm{d}\mathbf{r}^N \mathrm{d}\boldsymbol{\omega}^N, \tag{5}$$

where U is the total potential energy, $\beta = \frac{1}{k_B T}$, and k_B is the Boltzmann constant. The integration variables $\mathrm{d}\mathbf{r}^N$ and $\mathrm{d}\boldsymbol{\omega}^N$ are made explicit because these are the variables which change in an *NVT* trial, and their randomly generated values for perturbations must be chosen uniformly (e.g., for $\mathrm{d}\mathbf{r}$, choosing uniformly in each dimension, and for $\mathrm{d}\boldsymbol{\omega}$, choosing uniformly on the surface of a unit sphere). Because Eq. 4 requires information about only the ratio of microstate probabilities, we do not need to include any coordinate-independent terms that cancel when forming the ratio. Thus, terms such as the partition function, the thermal de Broglie wavelength and the accounting of indistinguishability do not contribute to the *NVT* ratio of probabilities.

Given Eq. 5, the ratio of *NVT* microstate probabilities, the first ratio in Eq. 4, is given by

$$\frac{\Pi_n}{\Pi_o} = e^{-\beta \Delta U}, \tag{6}$$

where $\Delta U = U_n - U_o$, U_n and U_o are the potential energies of the new and old state, respectively, and N, V and T are constant. This expression is used for any trial that changes only the positions and orientations of the particles.

5.1 Particle ~~displacement~~ translation trial

An MC ~~particle-displacement~~ translation trial move is defined as follows. A particle is randomly chosen among all of the mobile, non-fixed particles in the old microstate. The total number of mobile particles is $N_m = \sum_i^{mobile} N_i$, where i is the particle type. The randomly chosen particle with position $\mathbf{r}_o$ in the old microstate is then ~~displaced~~ translated by a random amount selected uniformly within $\pm\delta/2$ in each of D dimensions, to a new position labeled $\mathbf{r}_n$, thereby defining the new microstate.

For this trial, the transition probability, $\alpha_{o\to n}$, may be obtained as the product of the transition probabilities for each step. Each step that involves the generation of a random number should be considered. In this ~~particle-displacement~~ translation trial, there are two such steps. In the first, a particle is chosen randomly among N_m mobile particles. This means that the probability of choosing that particular particle, $1/N_m$, is the transition probability of the first step. In the second step, $\mathbf{r}_n$ is chosen uniformly within a region of volume δ^D. Thus, the transition probability density of choosing $\mathbf{r}_n$ in the second step is $1/\delta^D$, and the transition probability is $\mathrm{d}\mathbf{r}/\delta^D$. In a computer experiment, positions may be determined only within machine precision, which is represented here by $\mathrm{d}\mathbf{r}$. Formally, $\mathrm{d}\mathbf{r}$ is required to quantify the probability of choosing a position in continuous space. Thus,

$$\alpha_{o\to n} = \frac{\mathrm{d}\mathbf{r}}{N_m \delta^D}. \tag{7}$$

In order to solve for the trial acceptance probability, χ, the reverse trial transition probability from the new microstate

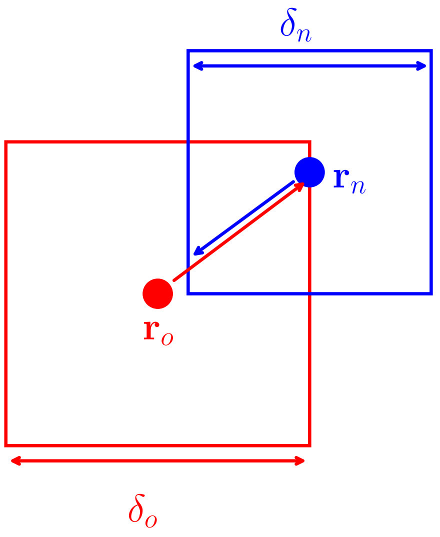

Figure 2. Detailed balance is violated if the tunable ~~displacement~~ translation parameter, δ, decreases. If the particle is ~~displaced~~ translated from the old position, $\mathbf{r}_o$, within δ_o in each dimension, then the particle may not be able to return to $\mathbf{r}_o$ in the reverse trial if the next ~~displacement~~ translation is within $\delta_n < \delta_o$. This is why δ should change only during equilibration and not during production simulations.

Forward	$\alpha_{o\to n}$
Choose from N_m	$1/N_m$
Choose $\mathbf{r}_n$	$\mathbf{dr}/\delta^D$

Reverse	$\alpha_{n\to o}$
Choose from N_m	$1/N_m$
Choose $\mathbf{r}_o$	$\mathbf{dr}/\delta^D$

Table 1. ~~Displacement~~ Translation transition probabilities.

Code Block 1. MC simulation of a single particle in one dimension attached to a harmonic spring using Python 3, where the statistics module is provided in Section 12.2.1. Note that a random number generator seed of "43279857" is provided so that the simulation is reproducible.

```
import random
import numpy as np
from statistics import block
random.seed(43279857)
beta = random.uniform(1, 2)
print("Expected energy:", 0.5/beta)
positions = np.zeros(int(1e6))
energies = np.zeros(len(positions))
for trial, _ in enumerate(positions[1:]):
    delta = 0.5*random.uniform(-1., 1.)
    position_new = positions[trial-1] + delta
    energy_new = (position_new**2)/beta
    delta_energy = energy_new - energies[trial - 1]
    if random.random() < np.exp(-beta*delta_energy):
        positions[trial] = position_new
        energies[trial] = energy_new
    else:
        positions[trial] = positions[trial - 1]
        energies[trial] = energies[trial - 1]
print("Position:", block(positions))
print("Energy:", block(energies))
```

to the old microstate, $\alpha_{n\to o}$, must also be obtained in order to utilize Eq. 4. The reverse transition must always be capable of returning to the old microstate; otherwise detailed balance is violated. For example, detailed balance is violated if δ is tuned to reach a desired acceptance probability. When δ is made smaller after a ~~displacement~~ translation trial, it may be impossible for the next trial to reverse back to the old state, as illustrated in Fig. 2. For this reason, δ should be changed only during equilibration and not during production simulations.

Throughout this article, we tabulate these forward and reverse transition probabilities, π, step-by-step with a corresponding table, as shown for a ~~displacement~~ translation trial in Table ~~??~~1. This table should indicate every use of a random number generator in both the forward and reverse transitions. Again, the overall transition probability for a trial is the product of the probabilities for a given direction.

The reverse transition probability is given by

$$\alpha_{n\to o} = \frac{\mathbf{dr}}{N_m\delta^D}. \tag{8}$$

Substituting Eqs. 6, 7 and 8 into Eq. 4 yields,

$$\chi = e^{-\beta\Delta U} \tag{9}$$

for the Metropolis acceptance of ~~particle-displacement~~ translation trials.

Code Block 1 tests Eq. ~~??~~ 9 for a single particle subject to an ~~the~~ external harmonic potential centered about the origin in one dimension. The expected average position is the origin. Also, according to the equipartition theorem, the expected average energy is $0.5k_BT$.

One major difference between MD and MC is that all particles are moved simultaneously in one MD time step, while typically only a single particle is moved in one MC trial. If multiple particles were moved in one MC trial, then only one unfavorable overlap from one particle could cause the rejection of an entire trial. Thus, it is typically more efficient to perturb only one particle at a time. However, there are special cases where it is still efficient for more than one particle to move in a single trial, as discussed in Section 11.5.

The tunable ~~displacement~~ translation parameter may have an optimal value in a dense fluid. Larger ~~displacements~~ translations are likely to overlap with another particle, while smaller ~~displacements are too~~ translations are more correlated with the old microstate, reducing sampling efficiency. Tunable ~~displacement~~ translation parameters are optimized by changing δ during an equilibration period to achieve a desired acceptance, often 25 % [68, 81–84]. Although Metropolis *et. al* [1] used 50 %, it seems that higher efficiencies may be achieved with slightly lower acceptances for particle ~~displacements~~ translation because rejects can be less expensive (e.g., hard overlap detected, updating neighbor lists, etc.) and larger ~~displacements~~ translations can lead to large configuration changes, although too low of acceptance also runs the risk of poor sampling.

An efficient implementation may have a ~~displacement~~ translation parameter, δ_i, specifically tuned for each particle type, *i*. This could be important to improve sampling in cases where the particle sizes or interactions are different. In addition, particles are chosen from all mobile particles, N_m~~, for sampling efficiency. A possibly inefficient~~. An alternative would be to define separate trials that select among specific particle types. In that case, if there are more particles of one type than another, the species with fewer numbers ~~would~~ could have a relatively larger probability of ~~displacement~~translation. The relative efficiency of this alternative approach compared to the one derived here is beyond the scope of this article. If the minor species is of more interest, the alternative could be more efficient. On the other hand, inefficiencies could arise if the minor species were caged in solvent that was not as well sampled.

The goal of ~~particle-displacement~~ translation trials is to give every particle ~~an equal~~ a chance to move. In principle, any state may be visited, which is a key element of the ergodic hypothesis. Although sequential updates have been shown to converge faster than random updates in an Ising model, sequential updates break detailed balance and follow a weaker balance condition [60]. Trials that do not obey detailed balance are outside the scope of this article.

This ~~displacement~~ translation algorithm could also be carefully modified to select particles only in a specific region of space. But detailed balance violations could occur if the set of available particles changes. For example, it is possible for the ~~displacement~~ translation to take a particle out of this region. If so, the trial must be immediately rejected, because the particle that was taken out of the region cannot be chosen for the reverse transition and therefore detailed balance is violated.

5.2 Particle ~~orientation change~~ orientation-change trial

Orientations are typically fixed during particle ~~displacements~~ translation trials so as not to introduce a second tunable parameter for orientation moves that cannot be decoupled from the ~~displacement~~ translation parameter during optimization. Instead, the orientation of a single particle may be perturbed in a separate trial move~~which proceeds as follows. A~~. A robust and intuitive way to accomplish this is via the axis-angle method: (1) a particle is randomly chosen among N_m mobile particles~~. A new orientation of that particle is then chosen according to the probability $P_{\omega n}\,\mathrm{d}\omega$. The~~; (2) a rotation axis $\hat{\mathbf{u}}$ is generated by sampling uniformly on the surface of a sphere (Sec. 12.2.2); (3) a rotation angle θ is generated uniformly on $\pm\delta\theta_{max}/2$ (with $\delta\theta_{max}$ a tunable parameter). With the rotation so defined, a rotation matrix can be computed

$$\mathbf{R} = \begin{bmatrix} C+(1-C)u_x^2 & (1-C)u_xu_y - Su_z & (1-C)u_xu_z + Su_y \\ (1-C)u_yu_x + Su_z & C+(1-C)u_y^2 & (1-C)u_yu_z - Su_x \\ (1-C)u_zu_x - Su_y & (1-C)u_zu_y + Su_x & C+(1-C)u_z^2 \end{bmatrix} \tag{10}$$

where $C = \cos\theta$ and $S = \sin\theta$. The new coordinate of interaction site *j* on particle *i* at $\mathbf{r}_i$ is given by $\mathbf{r}_{ij,\mathrm{n}} = \mathbf{r}_i + \mathbf{R}(\mathbf{r}_{ij,\mathrm{o}} - \mathbf{r}_i)$. If the particle's centroid is chosen for $\mathbf{r}_i$, then the centroid will be unchanged by the rotation (complementing the translation move, which leaves the orientation unchanged). However, using the centroid for $\mathbf{r}_i$ is not a requirement; any well-defined center for the rotation may be employed (e.g., the position of a specific interaction site).

Alternatively, if the orientation coordinate is held explicitly (e.g., as a quaternion), then it can be updated from $\hat{\mathbf{u}}$ and θ using an appropriate formula.

Regardless of the selected center for the rotation, the transition probabilities are given in terms of the axis-angle selection process outlined above. The elements of the transition probabilities for the particle orientation trial are summarized in Table 2. Together they yield the forward and reverse probabilities, which in this case are again equal

$$\alpha_{o\to n} = \alpha_{n\to o} = \frac{\mathrm{d}\omega\,\mathrm{d}\theta}{N_m 4\pi\delta\theta_{max}}. \tag{11}$$

Substituting Eq. 6 into Eq. 4 and using the transition probabilities in Table 2 yields,

$$\chi = \frac{P_{\omega o}}{P_{\omega n}} e^{-\beta\Delta U} \tag{12}$$

for the Metropolis acceptance of particle orientation changes. ~~For a rigid rotation of a particle in three dimensions, the orientation may be given by Euler angles or quaternions. However, when obtaining a random orientation, it is important not to select random values uniformly within the range of each of these variables. For example, independent and uniform selection of Euler angles will bias orientations toward the poles and makes the calculation of $P_{\omega n}$ and $P_{\omega o}$ nontrivial. Instead, the orientation should be uniformly chosen on the surface of a sphere [33]. In that case,~~

Forward	$\alpha_{o\to n}$
Choose from N_m	$1/N_m$
Choose rotation axis	$d\boldsymbol{\omega}/4\pi$
Choose rotation angle	$d\theta/\delta\theta_{max}$

Reverse	$\alpha_{n\to o}$
Choose from N_m	$1/N_m$
Choose rotation axis	$d\boldsymbol{\omega}/4\pi$
Choose rotation angle	$d\theta/\delta\theta_{max}$

Table 2. Orientation-change transition probabilities.

~~$P_{\omega n} = P_{\omega o}$. The d$\boldsymbol{\omega}$ terms in the probability of states, Π, include the orientations of the particles, which are sampled uniformly over the entire surface of a sphere and are normalized with the appropriate Jacobian. An example of randomly picking points on the surface of a sphere is shown in Section 12.2.2.~~

~~A more efficient approach for sampling particle orientations is to perturb the rigid orientation using a tunable parameter, δ. This must also be done in a careful way and not simply by changing the Euler angles uniformly and independently. One example algorithm to perturb orientation proceeds as follows. Generate a random quaternion, $\mathbf{q}$, uniformly on the surface of a sphere [79]. If the original particle orientation is represented by a quaternion, $\mathbf{q}_o$, then the newly proposed orientation is given by the quaternion $\mathbf{q}_n = \frac{\mathbf{q}_o+\delta\mathbf{q}}{|\mathbf{q}_o+\delta\mathbf{q}|}$. If the original particle orientation is represented by a collection of coordinates, use $\mathbf{q}_o = [0,0,0,1]$ (which corresponds to an identity rotation matrix) to generate a rotation matrix from $\mathbf{q}_n$ to perturb the coordinates. Because of the use of the tunable parameter,~~ To simplify notation, all remaining trial moves which include changes in orientation will simply refer to these transition probabilities as $P_{\omega n}$ and $P_{\omega o}$ ~~are equal to each other and therefore cancel one another in Eq. 12~~for the new and old states, respectively.

A larger tunable parameter, ~~δ~~$\delta\theta_{max}$, leads to a larger rotation, and thus ~~δ~~ $\delta\theta_{max}$ may be tuned in the same manner as the ~~displacement~~ translation tunable parameter to reach a target acceptance for each species. To avoid violating detailed balance, this tunable parameter should remain constant during the production simulations for the same reason as shown in Fig. 2.

If desired, generation of orientations from scratch, without perturbing on an existing orientation, is best accomplished by random generation of a quaternion. An example of randomly picking quaternions, and thus orientations, is shown in Section 12.2.2. ~~For particles with multiple interaction sites, one site is typically chosen as the center of the rigid body rotation. Because picking this one fixed site is deterministic and does not involve a random number generator,~~

Code Block 2 tests Eqs. 10 and 12 with the simulation of the orientation of a single rigid body given by two orthogonal unit vectors in 3 dimensions. Because there is no ~~associated probability that factors into Eq. 12.~~

~~For particles with multiple sites that are flexibly bonded, if the particle is not rigidly rotated but instead attempted to be regrown with configurational bias, starting from a given position of one of the interaction sites, then $P_{\omega n}$ and $P_{\omega o}$ are related to the product of the probability of choosing each of those interaction site positions [3, 34]. Particle regrowth is the subject of many studies [85–88], but is beyond the scope of this article, which concerns only the placement of the center of the first bead~~energy of interaction, the unit vectors are expected to be uniformly distributed on the surface of a sphere. In spherical coordinates, this manifests as a uniform distribution of the azimuthal angle $\in [-\pi, \pi]$, which is defined as the angle between the x-axis and the projection of the vector on the x-y plane, and a sine distribution for the polar angle $\in [0, \pi]$, which is the angle between the z-axis and the vector. While this azimuthal and polar angle of the first vector fully specifies the orientations of rigid bodies with an axis of symmetry, generic rigid bodies also require analysis of a third angle. That third angle was chosen to be the azimuthal angle of the second vector in the reference frame of the first vector. The reference frame was defined by the minimal rotation of the coordinate axes such that the z-axis aligns with the first vector. These analytical expectations are compared against the MC simulation in Fig. 3.

6 Unbiased isothermal-isobaric ensemble trials

Isothermal-isobaric ensemble trials change the volume, V, with constant number of particles, N, pressure, P and temperature, T. Volume-change trials are conducted while simultaneously scaling the coordinates of the particles uniformly. This is done for a variety of reasons that are discussed in Section 6.2. In order to derive the appropriate MC acceptance criteria for this trial, the microstate probability must be expressed in scaled coordinates given by $s_d = r_d/L_d$, where d is the dimensionality of the system. In the following two sections, we consider the NPT ensemble both without and with a "shell" particle [73, 89, 90], where a shell particle is used to remove redundant microstates from

Code Block 2. MC simulation of a single rigid body represented by two orthogonal unit vectors in three dimensions implemented in Python 3 using 5×10^5 random angular displacements with $\delta\theta_{max} = \pi/10$. Angle distributions were computed over 300 equally-spaced bins with 20 independent simulations (totaling 10^7 MC trials) to compare against theoretical expectations with approximately 68 % of the distribution within one standard deviation.

```
import random
import numpy as np
import unit_sphere
import rotation
random.seed(43279857)
num_sims = 20
num_trials = int(5e5)
num_bins = int(3e2)
def atan2(xs): return np.atan2(xs[:, 0, 1], xs[:, 0, 0])
def acos(xs): return np.acos(xs[:, 0, 2])
def psi(xs):
    """
    Return the azimuthal angle of second vector in a reference frame
    given by the minimal rotation of the z-axis into the first vector.
    """
    angs = np.zeros(len(xs))
    z = [0, 0, 1]
    for i,vecs in enumerate(xs):
        axis = np.cross(z, vecs[0])
        axis /= np.linalg.norm(axis)
        rtmx = rotation.axis_angle(axis, np.acos(np.dot(z, vecs[0])))
        angs[i] = np.atan2(np.dot(vecs[1], np.matmul(rtmx, [1, 0, 0])),
                           np.dot(vecs[1], np.matmul(rtmx, [0, 1, 0])))
    return angs
def azimuth(an): return 0*an + 0.5/np.pi
def polar(an): return 0.5*np.sin(an)
angs = {"azimuth": {"min":-np.pi, "fn":atan2, "dist":azimuth},
        "polar": {"min":0, "fn":acos, "dist":polar},
        "psi": {"min":-np.pi, "fn":psi, "dist":azimuth}}
hist = np.zeros(shape=(num_sims, len(angs), 2, num_bins))
for sim in range(num_sims):
    x = [[1, 0, 0], [0, 1, 0]]
    xyz = np.zeros(shape=(num_trials, 2, 3))
    dt_max = 2*np.pi # randomize first orientation
    for ixyz,_ in enumerate(xyz):
        angle = dt_max*random.uniform(-0.5, 0.5)
        dt_max = np.pi/10 # smaller perturbations after first
        axis = unit_sphere.uniform_surface()
        for ix,_ in enumerate(x):
            x[ix] = np.matmul(rotation.axis_angle(axis, angle), x[ix])
            xyz[ixyz][ix] = x[ix]
    for ia,a in enumerate(angs):
        counts, bins = np.histogram(angs[a]["fn"](xyz), bins=num_bins,
            density=True, range=[angs[a]["min"],np.pi])
        hist[sim][ia][0] = bins[1:] - 0.5*(bins[1] - bins[0])
        hist[sim][ia][1] = counts
for ia,a in enumerate(angs):
    inside = 0
    for ib,b in enumerate(hist[0, ia, 0, :]):
        mean = np.mean(hist[:, ia, 1, ib])
        std = np.std(hist[:, ia, 1, ib])/np.sqrt(num_sims)
        inside += np.abs(mean - angs[a]["dist"](b)) < std
    print("Percent", a, "within a stdev:", inside/num_bins)
```

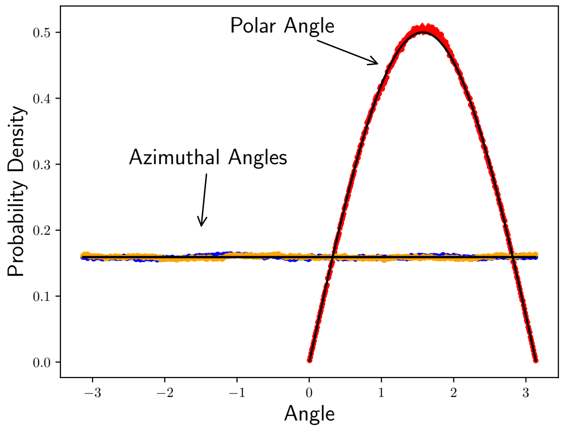

Figure 3. The (blue) azimuthal and (red) polar probability distributions of the first vector and (orange) azimuthal angle of the second vector in the reference frame of the first vector from MC simulations described in Code Block 2 with error bars from the standard deviation of the mean of 20 independent simulations, and analytical expectations shown by the black lines.

the isothermal-isobaric ensemble partition function.

6.1 Isothermal-isobaric (*NPT*) ensemble without a shell particle

The microstate probability in the *NPT* ensemble without a shell particle is proportional to [4, 33, 34]

$$\Pi\, d\mathbf{s}^N d\omega^N dV \propto V^N e^{-\beta(U+PV)}\, d\mathbf{s}^N d\omega^N dV. \tag{13}$$

The ratio of microstate probabilities, the first ratio in Eq. 4, is given by

$$\frac{\Pi_n}{\Pi_o} = \left(\frac{V_n}{V_o}\right)^N e^{-\beta(\Delta U + P\Delta V)}, \tag{14}$$

for trials that select random changes uniformly in V, because V is the explicit variable in Eq. 13. If an *NVT* trial is attempted in the *NPT* ensemble, $\Delta V = 0$, and Eq. 13 reduces to Eq. 6.

Eq. 13 and 14 no longer apply if changes are uniform in $\ln V$ instead of V. In this case, substituting $dV = V\, d\ln V$ into Eq. 13, the first ratio in Eq. 4 is

$$\frac{\Pi_n}{\Pi_o} = \left(\frac{V_n}{V_o}\right)^{N+1} e^{-\beta(\Delta U + P\Delta V)}. \tag{15}$$

See Code Block 3 for a test of Eqs. 14 and 15.

6.2 Isotropic volume change trial

Trial moves that attempt isotropic volume changes proceed as follows. The positions of the particles and periodic boundary lengths are scaled by a factor of $(1 + \Delta V/V_o)^{1/D}$ in each of D dimensions, where ΔV is chosen as a random amount selected uniformly within $\pm\Delta V_{\max}$. The particle positions are typically scaled with the volume in off lattice

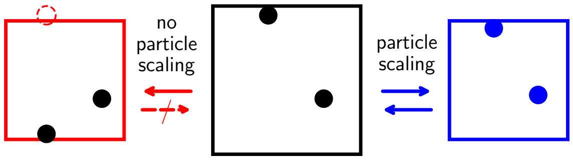

Figure 4. Detailed balance is violated if the particle positions are not scaled with the volume and one of the particles is wrapped across periodic boundaries. On the left, the top-most particle is wrapped to the bottom due to the decrease in volume when particle positions are not scaled. The reverse move, a subsequent increase in volume without scaling, will not return the system to its original microstate, shown in the center. On the right, detailed balance is obeyed when the particle positions are scaled with the volume.

Forward	$\alpha_{o\to n}$
Choose ΔV	$d\mathbf{r}/(2\Delta V_{\max})$
Reverse	$\alpha_{n\to o}$
Choose $-\Delta V$	$d\mathbf{r}/(2\Delta V_{\max})$

Table 3. Volume change transition probabilities.

simulations. Careful consideration of exactly how volume is changed is required. For example, as shown in Fig. 4, if particle positions are wrapped during volume changes that do not scale the positions, then detailed balance may be violated. This trial is summarized in Table 3.

Substituting Eq. 14 (without a shell particle) into Eq. 4, and using Table 3 for the second ratio in Eq. 4 results in

$$\chi = \left(\frac{V_n}{V_o}\right)^N e^{-\beta(\Delta U+P\Delta V)}. \tag{16}$$

This equation is also valid for uniform changes in one of the side lengths, L_d, with scaled particle positions because $dV \propto dL_d$ in Eq. 13 results in the same Eq. 14 for the first ratio in Eq. 4.

Eq. 16 is not valid for uniformly random changes in $\ln V$. Specifically, uniform changes in $\ln V$ occur when the particle coordinates and periodic boundary lengths are scaled by a factor of $e^{(\delta \ln V)/D}$, where $\delta \ln V$ is chosen as a random amount selected uniformly within $\pm\delta \ln V_{\max}$, and $\Delta V = V_o(e^{\delta \ln V} - 1)$. As discussed in Ref. [34] (without a shell particle) and Ref. [91] and Section 6 (with a shell particle), uniform changes in $\ln V$ use Eq. 15 instead of Eq. 14, resulting in the following acceptance probability

$$\chi = \left(\frac{V_n}{V_o}\right)^{N+1} e^{-\beta(\Delta U+P\Delta V)}. \tag{17}$$

Code Block 3 uses Eq. 16 or Eq. 17 for an ideal gas to compare against theory for uniform changes in V or $\ln V$, respectively.

Code Block 3. MC simulation of an ideal gas in the *NPT* ensemble using Python 3 with a shell particle and uniform changes in V or $\ln V$, and the statistics module is provided in Section 12.2.1.

```
import random
import numpy as np
from statistics import block
random.seed(43279857)
number = random.uniform(2, 3)
beta_pressure = random.uniform(0.25, 0.5)
print("Ideal gas volume:", number/beta_pressure)
for sampling in ["V", "lnV"]:
    vol = number/beta_pressure
    vols = np.zeros(int(1e6))
    for trial, _ in enumerate(vols):
        new_vol = vol
        if sampling == "V":
            new_vol += random.uniform(-3, 3)
            factor = number - 1
        else:
            new_vol *= np.exp(random.uniform(-3, 3))
            factor = number
        if new_vol > 0:
            bpdv = beta_pressure*(new_vol - vol)
            ln_prob = -bpdv + factor*np.log(new_vol/vol)
            if random.random() < np.exp(ln_prob):
                vol = new_vol
        vols[trial] = vol
    print("Volume:", block(vols))
```

6.3 Isothermal-isobaric ensemble with a shell particle

If a shell particle is introduced to remove redundant microstates in the partition function [73, 89–95], then there is one less particle that is scaled by the volume and Eqs. 13, 14 and 15 become

$$\Pi\, d\mathbf{s}^{N-1}\, d\omega^N\, dV \propto V^{N-1} e^{-\beta(U+PV)}\, d\mathbf{s}^{N-1}\, d\omega^N\, dV, \tag{18}$$

$$\frac{\Pi_n}{\Pi_o} = \left(\frac{V_n}{V_o}\right)^{N-1} e^{-\beta(\Delta U+P\Delta V)}, \tag{19}$$

and

$$\frac{\Pi_n}{\Pi_o} = \left(\frac{V_n}{V_o}\right)^{N} e^{-\beta(\Delta U+P\Delta V)}, \tag{20}$$

respectively. The acceptance probabilities shown in Section 6.2 without a shell particle would change according to these new microstate probabilities when a shell particle is included. Specifically, Eqs. 16 and 17 become

$$\chi = \left(\frac{V_n}{V_o}\right)^{N-1} e^{-\beta(\Delta U+P\Delta V)}. \tag{21}$$

and

$$\chi = \left(\frac{V_n}{V_o}\right)^{N} e^{-\beta(\Delta U+P\Delta V)}, \tag{22}$$

for uniform changes in V and $\ln V$, respectively.

7 Unbiased grand canonical ensemble (μVT) trials

In this section, we will derive insertion and deletion trials in two cases. In the first case, distinguishable particles in unscaled coordinates are considered. In the second case, indistinguishable particles in scaled coordinates are considered. Both cases will be shown to have identical acceptance probabilities.

7.1 Grand canonical ensemble with distinguishable particles in unscaled coordinates

A μVT trial changes the number of particles of type *i*, given by N_i with constant chemical potential, μ_i, constant number of other particles, $N - N_i$, volume, *V* and temperature, *T*. The μVT probability of a microstate is given by [5, 6, 33, 34]

$$\Pi(N_i)\,\mathrm{d}\mathbf{r}^N\,\mathrm{d}\boldsymbol{\omega}^N \propto \frac{e^{\beta\mu_i N_i - \beta U}}{\Lambda_i^{DN_i}}\,\mathrm{d}\mathbf{r}^N\,\mathrm{d}\boldsymbol{\omega}^N, \tag{23}$$

where Λ_i is the de Broglie wavelength of species *i*. In classical computer experiments, the particles are treated as distinguishable (e.g., each particle is distinguished by an index in an ordered array). In this case, the indistinguishable particle correction of $1/N!$ does not appear. We will show that identical acceptance probabilities are obtained whether we treat the particles as distinguishable or indistinguishable by including the $N!$ term in an alternative derivation in the following Section 7.3.

It is convenient to make Eq. 23 more compact by introducing the activity, z_i, of a particle of type *i*,

$$z_i = \frac{e^{\beta\mu_i}}{\Lambda_i^D}. \tag{24}$$

which has the units of inverse volume and is equivalent to the ideal gas number density. The number of particles of type *i* in an ideal gas, N_i^{id}, may be derived as the partial derivative of the grand potential with respect to the chemical potential of *i* at constant *V* and *T*, resulting in

$$N_i^{\mathrm{id}} = V\frac{e^{\beta\mu_i}}{\Lambda_i^D} = Vz_i. \tag{25}$$

Substituting Eq. 24 into Eq. 23 results in

$$\Pi(N_i)\,\mathrm{d}\mathbf{r}^N\,\mathrm{d}\boldsymbol{\omega}^N \propto z_i^{N_i} e^{-\beta U}\,\mathrm{d}\mathbf{r}^N\,\mathrm{d}\boldsymbol{\omega}^N. \tag{26}$$

Using Eq. 26, the ratio of μVT microstate probabilities in Eq. 4 is given by

$$\frac{\Pi_n}{\Pi_o} = (z_i\,\mathrm{d}\mathbf{r}\,\mathrm{d}\omega)^{\Delta N_i} e^{-\beta\Delta U}. \tag{27}$$

where $\Delta N_i = N_{in} - N_{io}$, and N_{in} and N_{io} are the number of particles of type *i* in the new and old microstate, respectively. If an *NVT* trial is attempted in the μVT ensemble, $\Delta N_i = 0$, and Eq. 27 is reduced to Eq. 6.

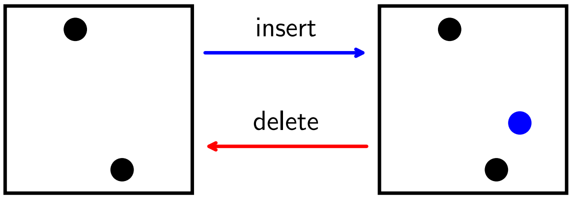

Figure 5. Insertion trials randomly add a particle, here shown in blue, uniformly within the simulation volume. Deletion trials remove a random particle, and are the reverse of insertions.

7.2 Insertion and deletion trials with distinguishable particles in unscaled coordinates

Because particle insertion and deletion trials [5, 6] are the reverse of each other, as shown in Fig. 5, they must be considered together to derive the acceptance probability. These two trials are collectively referred to as particle transfers (from an ideal gas state). Insertion and deletion trials for particles of type *i* proceed as follows. First, a particle insertion or deletion trial is chosen with equal probability.[1] If a particle insertion trial is chosen, then a particle type *i* is placed at a random position selected uniformly in the simulation volume, *V*. If a particle deletion trial is chosen, then a particle of type *i* is randomly chosen from among the N_i particles for removal. If multiple particle types are to be inserted and deleted, then the MC simulation could have a trial for each particle type.

In order to apply Eq. 4, reverse trials must be considered as having occurred immediately after the forward trial. This is important to note when deriving the probability of the reverse move, and to clarify that N_i is defined at the beginning of the insertion trial. The reverse trial deletion then has $N_i + 1$ particles of type *i*, including the particle that was added during the forward transition. Transition probabilities for insertion and deletion are not symmetric [5, 75]. The transition probability of an insertion trial as the forward transition is then summarized in Table 4.

Substituting Eq. 27 into the first ratio in Eq. 4, and using Table 4 for the second ratio in Eq. 4 results in

$$\chi = \frac{Vz_i}{P_{\omega n}(N_i+1)} e^{-\beta\Delta U}, \tag{28}$$

where $P_{\omega n} = 1$ (or incorporated into the definition of z_i) if a completely random orientation of a rigid body is chosen (see Section 5.2), or is otherwise related to the Rosenbluth

[1] More generally, insertions may be selected with a probability of P_{bias} that is not necessarily 1/2, and deletions with a probability of $1 - P_{bias}$. In multiple places in this article, we use a 1/2 probability for choosing between two different trials to simplify the equations.

Forward	$\alpha_{o\to n}$
Choose insert	1/2
Choose position in V	$d\mathbf{r}/V$
Choose orientation	$P_{\omega n}\, d\boldsymbol{\omega}$

Reverse	$\alpha_{n\to o}$
Choose delete	1/2
Choose particle of type i	$1/(N_i + 1)$

Table 4. Particle insertion transition probabilities.

Forward	$\alpha_{o\to n}$
Choose delete	1/2
Choose particle of type i	$1/N_i$

Reverse	$\alpha_{n\to o}$
Choose insert	1/2
Choose position in V	$d\mathbf{r}/V$
Choose orientation	$P_{\omega o}\, d\boldsymbol{\omega}$

Table 5. Particle deletion transition probabilities.

factor in CB partial regrowth of the particle (as discussed in Section 5.2).

Now consider the deletion as the forward transition. Deletion acceptance is nearly identical to what arises for the reverse of insertions, with one caveat. With insertion as the forward transition, N_i was defined at the beginning of the insertion trial (e.g., the old microstate, or before the particle was inserted). For deletion as the forward transition, the old microstate has N_i particles to choose among. To summarize, the transition probabilities with deletion as the forward transition and insertion as the reverse is given in Table 5.

Thus, the acceptance probability for deletions is given by

$$\chi = \frac{P_{\omega o} N_i}{V z_i} e^{-\beta \Delta U}, \tag{29}$$

where $P_{\omega o} = 1$ (or incorporated into the definition of z_i) if rigid bodies are inserted with random orientations (using quaternions, as discussed in Section 5.2 and 12.2.2), or otherwise related to the Rosenbluth factor in CB [34]. Code Block 4 uses Eqs. 28 and 29 for an ideal gas to compare against theory, Eq. 25.

7.3 Grand canonical ensemble with indistinguishable particles in scaled coordinates

Consider an alternative approach that includes the indistinguishable particle $N!$ correction, and uses scaled coordinates,

Code Block 4. MC μVT simulation of an ideal gas using Python 3, and the statistics module is provided in Section 12.2.1.

```
import random
import numpy as np
from statistics import block
random.seed(43279857)
Vz = random.uniform(0.1, 2)
print("Ideal gas number:", Vz)
number = int(Vz)
numbers = np.zeros(int(1e6))
for trial, _ in enumerate(numbers):
    if random.choice(["ins", "del"]) == "ins":
        if random.random() < Vz/(number + 1):
            number += 1
    else:
        if number > 0 and random.random() < number/Vz:
            number -= 1
    numbers[trial] = number
print("Number:", block(numbers))
```

as described in Ref. [34]. In this case, the μVT microstate probability is

$$\Pi(N_i)\, d\mathbf{s}^N\, d\boldsymbol{\omega}^N \propto \frac{V^N z_i^{N_i}}{N!} e^{-\beta U}\, d\mathbf{s}^N\, d\boldsymbol{\omega}^N. \tag{30}$$

The ratio of μVT microstate probabilities for the special case of indistinguishable particles and scaled coordinates is then

$$\frac{\Pi_n}{\Pi_o} = \frac{N_{io}!}{N_{in}!}\left(V z_i\, d\mathbf{s}\, d\boldsymbol{\omega}\right)^{\Delta N_i} e^{-\beta \Delta U}, \tag{31}$$

and is valid for changing the number of indistinguishable particles and choosing a newly inserted particle position in scaled coordinates, **s**, not **r**.

7.4 Insertion and deletion trials with indistinguishable particles in scaled coordinates

As opposed to deriving insertion and deletion acceptances with distinguishable particles in non-scaled coordinates, as in Sections 7.1 and 7.2, consider a similar trial conducted with indistinguishable particles in scaled coordinates, as in Ref. [34]. In this case, the first ratio in Eq. 4 is given by Eq. 31. But since the transition probabilities also change, the same acceptances are nonetheless obtained.

The insertion and deletion trial proceeds as follows. First, a particle insertion or deletion trial is randomly chosen with equal probability. If a particle insertion is chosen, then a particle of a given type i is placed at a random position selected uniformly in the scaled coordinates with a probability of $d\mathbf{s}$. If a particle deletion is chosen, then a random particle of a given type is chosen for removal. Because the particles are indistinguishable, the probability associated with the selection of a particle is 1. The forward and reverse transitions are summarized in Table 6.

Forward	$\alpha_{o\to n}$
Choose insert	1/2
Choose position in scaled coordinates	d**s**
Choose orientation	$P_{\omega n}\,d\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose delete	1/2
Choose from indistinguishable particles	1

Table 6. Alternative particle insertion transition probabilities.

The product of the first and second ratios in Eq. 4 is then identical to Eq. 28 because the first ratio from Eq. 31 now supplies the terms missing from the second ratio. Thus, the use of indistinguishable particles and scaled coordinates shifts the terms from the transition probabilities to the microstate probabilities. Although either approach is equally valid, the remainder of this article uses distinguishable particles, as implemented in computer simulations, and μVT derivations will use unscaled coordinates as described in Section 7.1, and not Section 7.3.

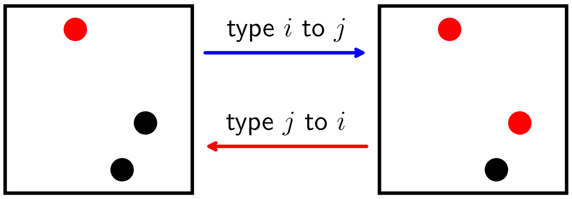

Figure 6. Semi-μVT change of a particle of type i, shown in red, into a particle of type j, shown in black. The total number of particles remains constant.

Forward	$\alpha_{o\to n}$
Choose a particle of any type	$1/N_m$
Choose new particle type j	$1/(n-1)$
Choose orientation	$P_{\omega j}\,d\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose particle of any type	$1/N_m$
Choose new particle type i	$1/(n-1)$
Choose orientation	$P_{\omega i}\,d\boldsymbol{\omega}$

Table 7. Particle type change transition probabilities.

8 Unbiased semi-grand canonical (semi-μVT) ensemble trials

Semi-μVT trials have fixed total numbers of particles, but the composition may change [7]. For a trial in which particles of type i may be replaced with particles of type j, and vice versa, the probability of a microstate in the semi-μVT ensemble is proportional to [7]

$$\Pi(N_i, N_j)\,d\mathbf{r}^N\,d\boldsymbol{\omega}^N \propto z_i^{N_i} z_j^{N_j} e^{-\beta U}\,d\mathbf{r}^N\,d\boldsymbol{\omega}^N. \qquad (32)$$

Using Eq. 32, the ratio of microstate probabilities of the semi-μVT ensemble in Eq. 4 is given by

$$\frac{\Pi_n}{\Pi_o} = z_i^{\Delta N_i} z_j^{\Delta N_j} e^{-\beta \Delta U}. \qquad (33)$$

with the constraint that $\Delta N_i + \Delta N_j = 0$.

8.1 Changing particle type

The semi-μVT ensemble described in Section 8 allows particles of one type to change into another, as shown in Fig. 6. A particle type change trial proceeds as follows. First, the trial is defined with a number of particle types, n that many change into each other. Thus, there are $N_m = \sum_i^{mobile} N_i$ changeable particles. One of those particles is randomly chosen with probability $1/N_m$. If the type of the chosen particle is given by i, then choose a random particle of type $j \neq i$ from among the $n-1$ other particle types. Then, add a particle of type j with a predetermined interaction site at the same coordinates as a predetermined interaction site in the chosen particle. Finally, remove the chosen particle, and randomly orient the newly added particle. The transition probabilities are summarized in Table 7.

Using Table 7 and substituting Eq. 33 into Eq. 4, the acceptance probability for the $i \to j$ attempt as the forward transition is

$$\chi = \left(\frac{P_{\omega j}}{P_{\omega i}}\right)\frac{z_j}{z_i} e^{-\beta \Delta U}. \qquad (34)$$

If the particles are isotropic, or rigid and random orientations are chosen, $P_{\omega j} = P_{\omega i}$. Otherwise, $\frac{P_{\omega j}}{P_{\omega i}}$ is related to the Rosenbluth factors for configuration bias partial regrowth or orientational bias. Eq. 34 is tested against an $n = 2$ component ideal gas mixture in Code Block 5.

8.2 Changing particle type by choosing a specific type

In the previous Section 8.1, particles of multiple types are chosen randomly to attempt a change into another particle type. Instead of choosing from all particles of any type, an alternative method is to choose one type specifically to change into another. However, this alternative algorithm changes the acceptance criteria.

An alternative particle type change trial proceeds as follows. First, a $i \to j$ or $j \to i$ trial is randomly chosen with equal probability. For $i \to j$, a particle of type i is randomly chosen. A particle of type j is added to the system with a fixed

Code Block 5. Semi-μVT MC of a two component ideal gas mixture using Python 3, and the statistics module is provided in Section 12.2.1.

```
import random
import numpy as np
from statistics import block
random.seed(43279857)
Vz1 = random.uniform(10, 20)
Vz2 = random.uniform(10, 20)
N1 = int(Vz1)
N_total = N1 + int(Vz2)
def N2(): return N_total - N1
print("Analytical fraction:", Vz1/(Vz1 + Vz2))
N1s = np.zeros(int(1e6))
for trial, _ in enumerate(N1s):
    if random.random() < N1/N_total:
        if N1 > 0 and random.random() < Vz2/Vz1:
            N1 -= 1
    else:
        if N2() > 0 and random.random() < Vz1/Vz2:
            N1 += 1
    N1s[trial] = N1
print("Numerical fraction:", block(N1s/N_total))
```

Forward	$\alpha_{o\to n}$
Choose $i \to j$	1/2
Choose particle of type i	$1/N_i$
Place j on i	1
Choose orientation of j	$P_{\omega j}\, d\omega$

Reverse	$\alpha_{n\to o}$
Choose $j \to i$	1/2
Choose particle of type j	$1/(N_j + 1)$
Place i on j	1
Choose orientation of i	$P_{\omega i}\, d\omega$

Table 8. Alternative particle type change transition probabilities.

site in j at the same position as a fixed site in i. Particle j is randomly oriented or regrown about its fixed site. Particle i is removed from the system. For $j \to i$, a particle of type j is randomly chosen. A particle of type i is added to the system with a fixed site in i at the same position as a fixed site in j. Particle i is randomly oriented or regrown about its fixed site. Particle j is removed from the system.

For $i \to j$ as the forward transition, the transition probabilities are summarized in Table 8, where N_i and N_j are the numbers of each type of particle before the trial is implemented. Using Table 8.2 and substituting Eq. 33 into Eq. 4, the acceptance probability for the $i \to j$ attempt as the forward transition is

$$\chi = \frac{N_i}{N_j + 1}\left(\frac{P_{\omega j}}{P_{\omega i}}\right)\frac{z_j}{z_i}e^{-\beta\Delta U}. \qquad (35)$$

Code Block 6. Alternative algorithm for semi-μVT MC of an ideal gas mixture using Python 3, and the statistics module is provided in Section 12.2.1.

```
import random
import numpy as np
from statistics import block
random.seed(43279857)
vz1 = random.uniform(10, 20)
vz2 = random.uniform(10, 20)
N1 = int(vz1)
N = N1 + int(vz2)
def N2(): return N - N1
print("Analytical fraction:", vz1/(vz1+vz2))
N1s = np.zeros(int(1e6))
for trial, _ in enumerate(N1s):
    if random.choice(["1to2", "2to1"]) == "1to2":
        if N1 > 0 and random.random() < N1/(N2() + 1)*vz2/vz1:
            N1 -= 1
    else:
        if N2() > 0 and random.random() < N2()/(N1 + 1)*vz1/vz2:
            N1 += 1
    N1s[trial] = N1
print("Numerical fraction:", block(N1s/N))
```

The $j \to i$ trial is the same as Eq. 35 but with i and j indices swapped. If i and j are rigid molecules and random orientations are chosen (or if isotropic), $P_{\omega i} = P_{\omega j}$. Otherwise, $\frac{P_{\omega j}}{P_{\omega i}}$ is related to the Rosenbluth factors for configuration bias partial regrowth. Eq. 35 is tested against an ideal gas mixture in Code Block 6.

9 Unbiased Gibbs ensemble trials

In the Gibbs ensemble, separate domains are simulated for which chemical and mechanical equilibrium at constant temperature is established between them [8], although chemical equilibrium is sufficient to enforce mechanical equilibrium [96]. For notational convenience, we will consider only two domains such that the total volume, $V = V_1 + V_2$, is the sum of two volumes and the total number of particles of type i, N_i, is $N_{i1} + N_{i2}$. Volume changes require scaling, as described in Section 6, and the particle positions must be expressed in terms of the scaled coordinates. We do not consider constant-pressure Gibbs ensemble in this work [97]. Although recent studies show that Gibbs (and all other) ensemble simulations may utilize a shell particle [73], we consider only the application of a shell particle in the NPT ensemble in this article.

The probability of a microstate in the Gibbs ensemble is then given by [8, 34, 73],

$$\Pi \propto \frac{1}{\Lambda_i^{ND}} V_1^{N_{i1}} V_2^{N_{i2}} e^{-\beta(U_1+U_2)}\, d\mathbf{s}^N\, dV_1\, dV_2. \qquad (36)$$

For particle transfer between domains while holding the volume of each domain fixed, the ratio of microstates in the

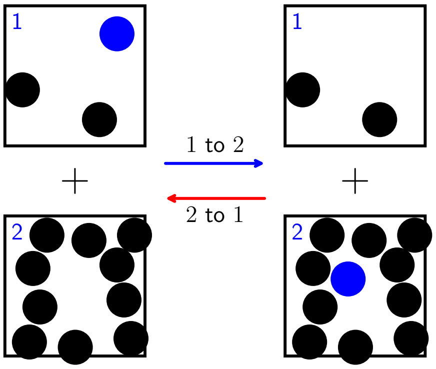

Figure 7. Gibbs ensemble transfer of the particle shown in blue from periodic domain 1 to periodic domain 2, and the reverse trial. The total number of particles remains constant.

Gibbs ensemble with particle transfer, the first ratio of Eq. 4, is given by

$$\frac{\Pi_n}{\Pi_o} = V_1^{N_{ni1}-N_{oi1}} V_2^{N_{ni2}-N_{oi2}} e^{-\beta\Delta(U_1+U_2)}, \tag{37}$$

where N_{ni1} and N_{ni2} are the number of particles of type i in the first and second domain, respectively, in the new microstate, while N_{oi1} and N_{oi2} are in the old microstate.

For volume transfer between domains while holding the number of particles in each domain fixed, the ratio of microstates in the Gibbs ensemble with volume transfer, the first ratio of Eq. 4, is given by

$$\frac{\Pi_n}{\Pi_o} = \left(\frac{V_{n1}}{V_{o1}}\right)^{N_{i1}} \left(\frac{V_{n2}}{V_{o2}}\right)^{N_{i2}} e^{-\beta\Delta(U_1+U_2)}, \tag{38}$$

where V_{n1} and V_{n2} are the volume of the first and second domain, respectively, in the new microstate, while V_{o1} and V_{o2} are in the old microstate.

9.1 Gibbs ensemble with particle transfer trials

In this example, consider particle transfers between two domains, as illustrated in Fig. 7. Particle transfers are conducted similarly to a simultaneous insertion in one domain and deletion in another domain, as described in Section 7.2. The total number of particles remains constant, as well as the volumes of both systems and the temperature.

Particle transfers between domains 1 and 2 in Gibbs ensemble MC proceed as follows. With equal probability, domain 1 is randomly chosen among all the domains to transfer a particle of a given type i. A particle of type i is then chosen for removal from domain 1 with probability $1/N_{i1}$. A particle of the same type is then added to domain 2 at a random position selected uniformly within the volume, V_2. The transition probabilities are summarized in Table 9, where N_{i1} and N_{i2} are the numbers of each type of particle in domains 1 and 2, respectively, before the trial is implemented.

Forward	$\alpha_{o\to n}$
Choose domain 1	1/2
Choose particle of type i in domain 1	$1/N_{i1}$
Choose $\mathbf{s}_n$ in domain 2	d$\mathbf{s}$
Choose orientation of particle i in domain 2	$P_{\omega 2}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose domain 2	1/2
Choose particle of type i in domain 2	$1/(N_{i2}+1)$
Choose $\mathbf{s}_o$ in domain 1	d$\mathbf{s}$
Choose orientation of particle i in domain 1	$P_{\omega 1}\,\mathrm{d}\boldsymbol{\omega}$

Table 9. Gibbs ensemble particle transfer transition probabilities.

Using Table 9.1 and substituting Eq. 37 into Eq. 4, the acceptance probability for the transfer of a particle of type i from domain 1 to domain 2 as the forward transition is

$$\chi = \frac{N_{i1}}{N_{i2}+1}\left(\frac{V_2}{V_1}\right)\frac{P_{\omega 1}}{P_{\omega 2}} e^{-\beta(\Delta U_1+\Delta U_2)}. \tag{39}$$

If particles of type i are rigid molecules and random orientations are chosen (or if isotropic), $P_{\omega 1} = P_{\omega 2}$. Otherwise, $\frac{P_{\omega j}}{P_{\omega i}}$ is related to the Rosenbluth factors for configuration bias partial regrowth. The transfer from domain 2 to domain 1 is identically derived by simply perturbing the indices 1 and 2. Eq. 39 is tested in an ideal gas simulation demonstrated in Code Block 7, where the densities of the two domains should be equivalent [73].

9.2 Gibbs ensemble with volume transfer trials

In this example, consider volume transfers between two domains, as illustrated in Fig. 8. Volume transfers are conducted similarly to a simultaneous expansion in one domain and correlated contraction in another domain, as described in Section 6.2. The total volume, $V = V_1 + V_2$, remains constant, as well as the numbers of particles of each type in both systems and the temperature.

Volume transfers between domains 1 and 2 in Gibbs ensemble MC proceed as follows. As described in Section 6.2, the positions of the particles and periodic boundary lengths are scaled by a factor of $(1+\Delta V/V_o)^{1/D}$ in each of D dimensions, where ΔV is chosen as a random amount selected uniformly within $\pm\Delta V_{\mathrm{max}}$. The scaling occurs with $-\Delta V$ for domain 1, and $+\Delta V$ for domain 2, where V_1 and V_2 are the volumes of domains 1 and 2 before the scaling occurs.

Using the transition probabilities in Table 10, and substituting Eq. 38 into Eq. 4, the acceptance probability for the

Code Block 7. MC simulation of an ideal gas in the Gibbs ensemble with chemical equilibrium using Python 3 should yield two domains with equivalent density, and the statistics module is provided in Section 12.2.1.

```
import numpy as np
import random
from statistics import block
random.seed(43279857)
V1 = random.uniform(4, 6)
V2 = random.uniform(2, 3)
N_total = int(V1 + V2)
N1 = int(V1)
def N2(): return N_total - N1
N1s = np.zeros(int(1e7))
for trial_num, _ in enumerate(N1s):
    if random.choice([1, 2]) == 1:
        if N1 > 0 and random.random() < V2*N1/(N2()+1)/V1:
            N1 -= 1
    else:
        if N2() > 0 and random.random() <V1*N2()/(N1+1)/V2:
            N1 += 1
    N1s[trial_num] = N1
print("Density 1:", block(N1s/V1))
print("Density 2:", block((N_total-N1s)/V2))
```

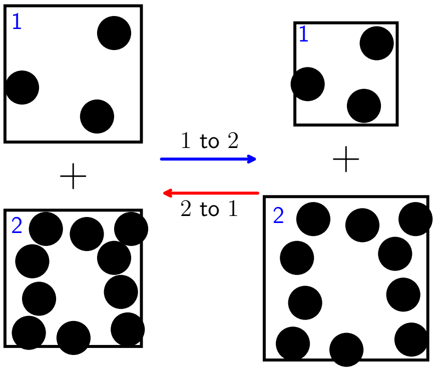

Figure 8. Gibbs ensemble transfer of volume from periodic domain 1 to periodic domain 2, and the reverse trial. The total volume remains constant and the particle positions are scaled as described in Section 6.2.

Forward	$\alpha_{o\to n}$
Choose ΔV	d**r**/(2$\Delta V_{\max}$)
Reverse	$\alpha_{n\to o}$
Choose $-\Delta V$	d**r**/(2$\Delta V_{\max}$)

Table 10. Gibbs ensemble volume transfer transition probabilities.

Code Block 8. MC simulation of an ideal gas in the Gibbs ensemble with mechanical equilibrium using Python 3 should yield two domains with equivalent density, and the statistics module is provided in Section 12.2.1.

```
import random
import numpy as np
from statistics import block
random.seed(43279857)
N1 = random.randint(2, 3)
N2 = random.randint(4, 5)
V_total = float(N1 + N2)
V1 = float(N1)
def V2(): return V_total - V1
V1s = np.zeros(int(1e7))
dVmax = 0.1*V1
for trial_num, _ in enumerate(V1s):
    dV = random.uniform(0, dVmax)
    if random.choice([1, 2]) == 1:
        chi = ((V1-dV)/V1)**N1*((V2()+dV)/V2())**N2
        if V1 > dV and random.random() < chi:
            V1 -= dV
    else:
        chi = ((V2()-dV)/V2())**N2*((V1+dV)/V1)**N1
        if V2() > dV and random.random() < chi:
            V1 += dV
    V1s[trial_num] = V1
print("Density 1:", block(N1/V1s))
print("Density 2:", block(N2/(V_total-V1s)))
```

transfer of volume from domain 1 to domain 2 as the forward transition is

$$\chi = \left(\frac{V_1 - \Delta V}{V_1}\right)^{N_{i1}} \left(\frac{V_2 + \Delta V}{V_2}\right)^{N_{i2}} e^{-\beta(\Delta U_1 + \Delta U_2)}. \qquad (40)$$

Eq. 40 is tested in an ideal gas simulation demonstrated in Code Block 8, where the densities of the two domains should be equivalent [73].

9.3 Gibbs ensemble with particle and volume transfer trials

Gibbs ensemble simulations may include both trials with volume and particle transfers described in the previous sections. Eqs. 39 and 40 are tested in an ideal gas simulation demonstrated in Code Block 9, where the number of particles and volume should be equivalent [73].

10 Biased grand canonical ensemble (μVT) trials

Now that all ensembles in this article have been tested with unbiased trials, we now derive biased μVT trials. For simulations of an ideal gas, these trials reduce to those already described in the previous sections. Thus, ideal gas tests are not explicitly included in the remaining sections for biased trials. However, this does not mean ideal gas tests are not useful for testing biased trials. In fact, ideal gas tests are even

Code Block 9. MC simulation of an ideal gas in the Gibbs ensemble with mechanical and chemical equilibrium using Python 3 should yield two domains with equivalent number of particles and volume.

```
import random
import numpy as np
from statistics import block
random.seed(43279857)
V1 = random.uniform(20, 40)
N1 = int(V1)
V_total = V1 + random.uniform(20, 40)
def V2(): return V_total - V1
def N2(): return N_total - N1
N_total = N1 + int(V2())
N1s = np.zeros(int(1e7))
V1s = np.zeros(len(N1s))
dVmax = 0.1*V1
for trial_num, _ in enumerate(N1s):
    if random.choice(["N", "V"]) == "N":
        if random.choice([1, 2]) == 1:
            if N1 > 1:
                if random.random() < V2()*N1/(N2() + 1)/V1:
                    N1 -= 1
        else:
            if N2() > 1:
                if random.random() < V1*N2()/(N1 + 1)/V2():
                    N1 += 1
    else:
        dV = random.uniform(0, dVmax)
        if random.choice([1, 2]) == 1:
            chi = ((V1-dV)/V1)**N1* ((V2()+dV)/V2())**N2()
            if V1 > dV and random.random() < chi:
                V1 -= dV
        else:
            chi = ((V2()-dV)/V2())**N2()* ((V1+dV)/V1)**N1
            if V2() > dV and random.random() < chi:
                V1 += dV
    N1s[trial_num] = N1
    V1s[trial_num] = V1
print("N1/(N1+N2):", block(N1s/N_total))
print("V1/(V1+V2):", block(V1s/V_total))
```

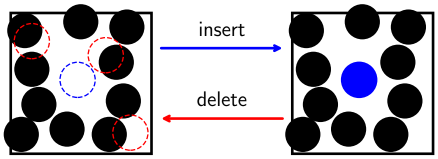

Figure 9. Multiple first bead configurational bias insertion and deletion considers randomly-generated positions for a newly inserted particle, shown by the dashed circles, in an existing fluid of particles, shown by the black circles. The most favorable position has the highest Rosenbluth probability of selection, which is shown in blue due to excluded volume overlap of the other red positions with the existing fluid. The deletion step does not consider other positions or particles for deletion, but to obey detailed balance, the Rosenbluth factor in the deletion step includes the existing position as well as other randomly-generated positions.

more useful for biased trials where the additional complication leads to greater possibility for error.

10.1 Configurational bias (CB)

In this section, we first introduce the concept of CB [3, 9, 98, 99] in an unconventional way. In the literature, CB is typically described as a way to regrow particles with multiple interaction sites bonded together. In this article, the orientational probabilities, P_ω, encompass chain regrowth as well as orientational bias and anisotropic interaction sites. The determination of P_ω is not discussed in detail here beyond cases where a rigid particle is randomly oriented (see Section 5.2). Thus, we consider the application of CB only on the center of the first interaction site of a particle (e.g., multiple first bead CB).

One example of insertion and deletion of particles of type i with CB proceeds as follows. An insertion or deletion trial is randomly chosen with equal probability. If a particle insertion is chosen, then the new position, $\mathbf{r}_n$, of a particle of type i is chosen from c random positions selected uniformly in the volume, V, with a relative probability of π_{cn}. If a particle deletion is chosen, then one of N_i particles of type i is chosen for removal.

The forward and reverse transition probabilities are summarized in Table 11. Because the insertion is chosen as the forward transition, deletions are chosen among $N_i + 1$ particles, where N_i is the number of particles of type i as defined from the old microstate.

Using Table 11 and substituting Eq. 27 into Eq. 4, the acceptance probability for CB particle insertion is

$$\chi = \frac{V z_i}{P_{\omega n} c(N_i + 1)\pi_{cn}} e^{-\beta\Delta U}. \qquad (41)$$

There is remarkable flexibility as to the choice of $\mathbf{r}_n$ from c positions, which determines π_{cn}. Ideally, π_{cn} biases configurations where the energy of interaction of (the first interaction

Forward	$\alpha_{o\to n}$
Choose insert	1/2
Choose c positions in V. Probability that $\mathbf{r}_n$ is in c.	$c\,\mathrm{d}\mathbf{r}/V$
Choose $\mathbf{r}_n$ from c positions	π_{cn}
Choose orientation	$P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose delete	1/2
Choose particle of type i	$1/(N_i+1)$

Table 11. Configurational bias particle insertion transition probabilities.

Forward	$\alpha_{o\to n}$
Choose delete	1/2
Choose particle of type i	$1/N_i$
Reverse	$\alpha_{n\to o}$
Choose insert	1/2
Choose c positions in V. Probability that $\mathbf{r}_o$ is in c.	$c\,\mathrm{d}\mathbf{r}/V$
Choose $\mathbf{r}_o$ from c positions	π_{co}
Choose orientation	$P_{\omega o}\,\mathrm{d}\boldsymbol{\omega}$

Table 12. Configurational bias particle deletion transition probabilities.

site of) a new particle at $\mathbf{r}_n$ with all the particles in the old microstate, U_n, is favorable. A natural choice is to Boltzmann weight the selection of c positions, $\pi_{cn} \propto e^{-\beta U_n}$. Normalizing such a probability results in the Rosenbluth factor,

$$\pi_{cn} = \frac{e^{-\beta U_n}}{W_c}, \tag{42}$$

where

$$W_c = \sum_i^c e^{-\beta U_i}. \tag{43}$$

Substitution of Eq. 42 into Eq. 41 yields

$$\chi = \frac{V z_i W_c}{P_{\omega n} c (N_i + 1)} e^{-\beta(\Delta U - U_n)}. \tag{44}$$

This form of π_{cn} is also convenient because $\Delta U - U_n$ only contains interaction energies beyond the center of the first site (e.g., if there were multiple sites). If there are no other orientational or multiple site interactions beyond the first bead, $\Delta U = U_n$.

The deletion step is nearly the inverse of Eq. 44, except for two details regarding the definition of N_i and π_{co}. Consider the deletion as the forward transition, where the particle is randomly chosen from N_i in the old microstate. The typical Rosenbluth factor, W_c, is calculated to satisfy detailed balance, but plays no role in the choice of the particle for deletion. The trial with the deletion as the forward transition is described in Table 12.

Thus, for deletions, the acceptance criteria using the typical Rosenbluth factor is given by,

$$\chi = \frac{c N_i P_{\omega o}}{V z_i W_c} e^{-\beta(\Delta U + U_o)}, \tag{45}$$

where $\mathbf{r}_o$, the original position of the deleted particle, is among the $c - 1$ other randomly generated positions in π_{co}, and U_o is the interaction energy of the particle at $\mathbf{r}_o$ that the trial is attempting to delete. If there are no other orientational or multiple site interactions beyond the first bead, $\Delta U = -U_o$.

In the ideal gas limit, $U \to 0$, $W_c \to c$, and Eqs. 44 and 45 are equivalent to Eqs. 28 and 29. Ideal gas simulations are helpful in quickly testing a CB implementation for verification. Although c in Eqs. 44 and 45 may be incorporated into the chemical potential, this may lead to confusion and is not recommended if multiple kinds of μVT trials are attempted.

For single-site isotropic particles, such as those interacting via the Lennard-Jones potential, the orientational term (see Section 5.2) results in $P_{\omega n} = 1$ (or can be incorporated into the definition of z_i). Hence, this trial is not particularly efficient compared to simply conducting c traditional insertion or deletion trials sequentially. However, this method becomes more efficient when the calculation of $P_{\omega n}$ for more complex molecules is comparable or more expensive than the calculation of W_c (e.g., using partial regrowth or orientational bias, etc). Another efficient approach is to use a reference potential that is less expensive than the full potential with longer-range interactions, where the reference potential still contains the most relevant terms for sampling (e.g., excluded volume in a dense fluid). Such a method is named dual-cut CB [72] as described in the next section.

10.2 Dual-cut configurational bias (DCCB)

DCCB [72] is derived in a nearly identical fashion as CB presented in Section 10.1. The only difference is that π_{cn} in Eq. 42 no longer uses the energy of the system, U, but rather, the energy of a reference system, U_r, that is, ideally, less computationally expensive to calculate. This reference system could have a shorter cutoff distance, as per the namesake of this method. Or, it could have a different potential or a different number of interaction sites. For example, a simulation of SPC/E water could use a reference potential of hard spheres centered on the oxygen, which is not only short range but also reduces the number of interaction sites. Because π_{cn} contains the reference potential, U_{rn}, the insertion acceptance is given by

$$\chi = \frac{Vz_i W_{cr}}{P_{\omega nr} c(N_i + 1)} e^{-\beta(\Delta U - U_{rn})}, \quad (46)$$

where

$$W_{cr} = \sum_i^c e^{-\beta U_{ri}} \quad (47)$$

and U_{ri} is the reference potential contribution of the particle in microstate i. Similar to Section 10.1, the deletion acceptance is the inverse of the insertion, except that $N_i + 1 \rightarrow N_i$, and $\mathbf{r}_o$ is one of the c randomly generated positions for the calculation of W_{cr}.

The orientational term, $P_{\omega nr}$, may also utilize the reference potential, or not. In either case, U_{rn} in Eq. 46 is the sum of the (reference) interaction energies of each of the microstates chosen from both $P_{\omega nr}$ and W_{cr} when the typical Rosenbluth terms are used in the probabilities of selection.

10.3 Cavity bias

Particle insertions may be biased to specific regions according to the location of cavities in a dense fluid [100–103]. The practical limitation of this method is that it may become computationally expensive to identify the cavities. In one approach, a number of test points are randomly placed in the volume for each trial [100–102]. An ensemble average of the fraction of test points found within cavities, P_c, is then multiplied with the volume to approximate the combined volume of the cavities. An alternative approach is to overlay the simulation domain with a discrete grid of points, as shown in Fig. 10. To improve efficiency, this grid of points could be continuously updated for each trial. In this case, the fraction of the grid that is within at least one of the cavities times the volume of the entire domain is the approximate volume of the cavities, assuming an equally spaced grid with enough resolution. When a randomly chosen grid point within a cavity is chosen, the inserted particle may be placed at a random position selected uniformly within the voxel nearest that grid point. In the case of an absorbent and adsorbent material, grid points that are always in the impenetrable regions of the absorbent material may be ignored [102]. For particles with multiple sites, it may be convenient to consider just one of the sites for this cavity bias step (e.g., only the oxygen in a water molecule [104]). Voronoi tessellations may also be used to identify cavity volumes [105]. In addition, cavity bias can also be implemented efficiently in lattice simulations [106].

In this article, two variants of cavity bias will be derived. In the first, a number of test points are chosen randomly in the domain to test for a cavity. In the second, a predefined grid is used instead.

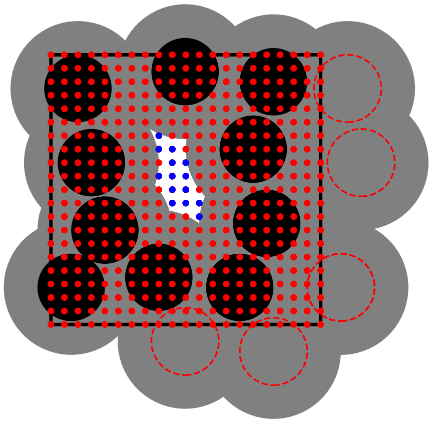

Figure 10. Cavity bias insertion and deletion may utilize a collection of points, here shown by a grid of red and blue points, which are tested for cavities. Cavities are where another particle could fit without excluded volume overlap with existing particles, shown here as black circles with grey excluded volumes. The red circles with dashed lines are periodic images, and the blue dots in the center are the only points in the grid that a new particle could be added onto without overlap.

10.3.1 Cavity bias with random test points

A cavity bias trial with random test points proceeds as follows. With equal probability, insertion or deletion of a particle of type i is randomly chosen. For insertion, N_{test} points are chosen randomly in the simulation volume and n_c of these points are determined to be within a cavity. For hard particle systems, a cavity is well defined. Soft particle systems require an ambiguous cavity definition that could affect sampling and efficiency (e.g., for the purposes of cavity determination, pick an arbitrary hard particle size for the soft particles). Here, n_c is computed for each set of randomly generated test points; the possibility of using an ensemble average of n_c is discussed at the end of this section. A particle of type i is placed on one of the n_c points. For deletion, randomly choose a particle of type i for removal.

With insertion as the forward transition, the transition probabilities are summarized in Table 13. Because the insertion is the forward transition, deletions are chosen among the $N_i + 1$ particles, where N_i is the number as defined from the old microstate.

Using Table 13 and substituting Eq. 27 into Eq. 4, the acceptance probability for cavity bias particle insertion is

$$\chi = \left(\frac{n_c}{N_{test}}\right) \frac{z_i V}{P_{\omega n}(N_i + 1)} e^{-\beta \Delta U}. \quad (48)$$

For comparison to previously published articles, the probability that a random test point is in the cavity is given by $P_c = n_c/N_{test}$ [100–102]. With deletion as the forward transition, the transition probabilities are summarized in Table 14.

Forward	$\alpha_{o \to n}$
Choose insert	1/2
Choose N_{test} points in V. Probability $\mathbf{r}_n$ is in N_{test}.	$N_{test}\,\mathrm{d}\mathbf{r}/V$
Choose $\mathbf{r}_n$ from n_c	$1/n_c$
Choose orientation	$P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n \to o}$
Choose delete	1/2
Choose particle of type i	$1/(N_i + 1)$

Table 13. Cavity bias particle insertion transition probabilities with random test points.

Forward	$\alpha_{o \to n}$
Choose delete	1/2
Choose particle of type i	$1/N_i$
Reverse	$\alpha_{n \to o}$
Choose insert	1/2
Choose N_{test} points in V. Probability $\mathbf{r}_o$ is in N_{test}.	$N_{test}\,\mathrm{d}\mathbf{r}/V$
Choose $\mathbf{r}_o$ from n_c	$1/n_c$
Choose orientation	$P_{\omega o}\,\mathrm{d}\boldsymbol{\omega}$

Table 14. Cavity bias particle deletion transition probabilities with random test points.

Using Table 14 and substituting Eq. 27 into Eq. 4, the acceptance probability for cavity bias particle deletions is

$$\chi = \frac{N_{test}}{n_c} \left(\frac{N_i P_{\omega o}}{z_i V} \right) e^{-\beta \Delta U}, \tag{49}$$

where n_c must be computed after the particle is deleted and the N_{test} points includes the original position of the particle, $\mathbf{r}_o$. For hard particle systems, a cavity is well defined and n_c in the deletion step would never be zero because it includes $\mathbf{r}_o$. But for soft interactions, depending on the definition of the cavity, n_c may be zero. If n_c is zero, the trial should be rejected or else detailed balance is violated because the reverse transition is impossible.

For comparison to previously published articles [100–102], it is not clear if $\mathbf{r}_o$ was included in one of the N_{test} points during the deletion step. The previous work also computed an ensemble average of n_c over many trials rather than computing an instantaneous n_c. An ensemble average of n_c has the benefit of possibly requiring fewer N_{test} points. On the other hand, it is beyond the scope of this article to prove whether or not an ensemble average n_c introduces biases via violation of detailed balance [103]. The previous work also interpolated the cavity probability from other state points in special cases, and introduced unbiased insertion and deletion trials when $n_c = 0$.

10.3.2 Cavity bias with a predefined list

While Section 10.3.1 used random points, a predefined list or grid, as shown in Fig. 10, is considered in this section. The benefit of using a predefined list is that certain points could be removed if, for example, those grid points could never be in a cavity because of the location of a rigid wall or obstacle (e.g., an adsorbent material). A grid is created with spacing δ in each dimension, and after points not amenable to insertion are removed, there are N_l points remaining in the predefined list.

A cavity bias trial with a predefined list proceeds as follows. With equal probability, insertion or deletion of a particle of type i is randomly chosen. For an insertion, N_{test} points are randomly chosen from the fixed list of N_l points, and n_c of those N_{test} points are determined to be within a cavity. One of these n_c points is randomly chosen, denoted as $\mathbf{r}_{nl}$. The inserted particle of type i is then placed at a random position selected uniformly within $\pm 0.5\delta$ of $\mathbf{r}_{nl}$ in each dimension. For a deletion, randomly choose a particle of type i and remove it. In order to obey detailed balance, deletion attempts of particles that are not within $\pm 0.5\delta$ of a point in N_l must be immediately rejected (e.g., if some grid points were removed yet fluid particles managed to exist in the vicinity of those removed points). Otherwise, particles could be removed from the system where insertions could not immediately place them back for the reverse trial.

Forward	$\alpha_{o\to n}$
Choose insert	1/2
Choose N_{test} points from N_l. Probability $\mathbf{r}_{nl}$ is in N_{test}	N_{test}/N_l
Choose $\mathbf{r}_{nl}$ from n_c	$1/n_c$
Choose $\mathbf{r}_n$ about $\mathbf{r}_{nl}$	$\mathrm{d}\mathbf{r}/\delta^D$
Choose orientation	$P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose delete	1/2
Choose particle of type i	$1/(N_i+1)$

Table 15. Cavity bias particle insertion transition probabilities with a predefined list.

With insertion as the forward transition, the transition probabilities are summarized in Table 15. Because the insertion is the forward transition, deletions are chosen among the $N_i + 1$ particles, where N_i is the number of particles of type i as defined from the old microstate.

Using Table 15 and substituting Eq. 27 into Eq. 4, the acceptance probability for cavity bias particle insertion with N_{test} points is

$$\chi = N_l\delta^D \left(\frac{n_c}{N_{test}}\right) \frac{z_i}{P_{\omega n}(N_i + 1)} e^{-\beta\Delta U}. \tag{50}$$

This equation takes the form of Eq. 4 of Ref. [102] because the fraction of the points in the cavity is $P_c = n_c/N_{test}$ and the total volume considered for insertions is $v = N_l\delta^D$.

With deletion as the forward transition, the position of the chosen particle to be deleted is $\mathbf{r}_o$, which is within δ in each dimension of a point within the predefined list, $\mathbf{r}_{ol}$. If not, the deletion is rejected to obey detailed balance. The transition probabilities are summarized in Table 16.

Using Table 16 and substituting Eq. 27 into Eq. 4, the acceptance probability for cavity bias particle deletions with N_{test} points is

$$\chi = \frac{1}{N_l\delta^D} \left(\frac{N_{test}}{n_c}\right) \frac{N_i P_{\omega o}}{z_i} e^{-\beta\Delta U}. \tag{51}$$

Similar to Eq. 49, n_c is computed after the particle is deleted and includes $\mathbf{r}_{ol}$, which is the closest grid point to the old macrostate position of the deleted particle, $\mathbf{r}_o$.

10.4 Energy bias

Particle insertions may also be biased in specific regions according to their energy of interaction with a fixed adsorbent material [102], as illustrated in Fig. 11. The simulation volume is divided into a number of small voxels with volume v. Energy bias particle insertions and deletions proceed as follows.

Forward	$\alpha_{o\to n}$
Choose delete	1/2
Choose particle of type i	$1/N_i$
Reverse	$\alpha_{n\to o}$
Choose insert	1/2
Choose N_{test} from N_l points. Probability $\mathbf{r}_{ol}$ is in N_{test}	N_{test}/N_l
Choose $\mathbf{r}_{ol}$ from n_c	$1/n_c$
Choose $\mathbf{r}_o$ about $\mathbf{r}_{ol}$	$\mathrm{d}\mathbf{r}/\delta^D$
Choose orientation	$P_{\omega o}\,\mathrm{d}\boldsymbol{\omega}$

Table 16. Cavity bias particle deletion transition probabilities with a predefined list.

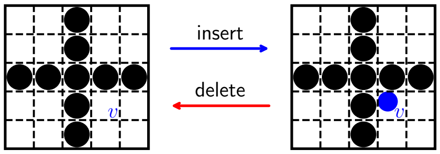

Figure 11. Energy bias insertion and deletion [102] divides the simulation domain into voxels, with 25 shown here by the dashed lines. One voxel, v, is chosen for the available insertion volume according to a probability Boltzmann-weighted by the energy of interaction with a rigid adsorbent frame, shown by the black circles. The reverse move is when the newly inserted fluid particle, shown in blue, is randomly chosen for deletion among all the fluid particles.

With equal probability, an insertion or deletion is attempted. For an insertion, a voxel is randomly chosen with probability P_c. A new particle of type i is then randomly placed inside of the chosen voxel. For a deletion, an existing particle of type i is chosen randomly and removed from the system. The trial probabilities are summarized with insertion as the forward transition in Table 17.

Substituting Eq. 27 into the first ratio in Eq. 4, and using Table 17 for the second ratio in Eq. 4, results in

$$\chi = \frac{vz_i}{P_{\omega n}(N_i + 1)P_c} e^{-\beta\Delta U} \tag{52}$$

for energy bias insertions. Deletions as the forward trial is summarized in Table 18.

Substituting Eq. 27 into the first ratio in Eq. 4, and using Table 18 for the second ratio in Eq. 4, results in

$$\chi = \frac{N_i P_c P_{\omega o}}{vz_i} e^{-\beta\Delta U} \tag{53}$$

for deletions as the reverse of energy bias insertions.

For both insertions and deletions, a suitable P_c must be utilized that is ideally inexpensive to compute and increases sampling in the desired location. As originally formulated, a

Forward	$\alpha_{o\to n}$
Choose insert	1/2
Choose voxel	P_c
Choose position in v	$\mathrm{d}\mathbf{r}/v$
Choose orientation	$P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose delete	1/2
Choose particle of type i	$1/(N_i+1)$

Table 17. Energy bias particle insertion transition probabilities.

Forward	$\alpha_{o\to n}$
Choose delete	1/2
Choose particle of type i	$1/N_i$
Reverse	$\alpha_{n\to o}$
Choose insert	1/2
Choose voxel	P_c
Choose position in v	$\mathrm{d}\mathbf{r}/v$
Choose orientation	$P_{\omega o}\,\mathrm{d}\boldsymbol{\omega}$

Table 18. Energy bias particle deletion transition probabilities.

good choice for P_c is to weight the probability by the Boltzmann factor of the energy a particle in the center of the chosen target volume, v, would have with an adsorbent material, U_t^a,

$$P_c = \frac{e^{-\beta U_t^a}}{\sum_i e^{-\beta U_i^a}}. \tag{54}$$

10.5 Insertion and deletion in a targeted region

Particle insertions and deletions may also be performed in a targeted region, v, shown in Fig. 12 with a rigid adsorbent framework. The trial proceeds as follows. First, a particle insertion or deletion trial is chosen randomly with equal probability. If a particle insertion trial is chosen, then a particle of type i is placed at a random position selected uniformly in a targeted region, v, with random orientation. If a particle deletion trial is chosen, then a random particle of type i within v is chosen for removal. Here, n_i is the number of particles of type i inside v at the beginning of the trial. For particle deletions, if $n_i = 0$, then reject the trial. To obey detailed balance, v cannot change during the trial. The transition probability of an insertion trial as the forward transition is then summarized in Table 19.

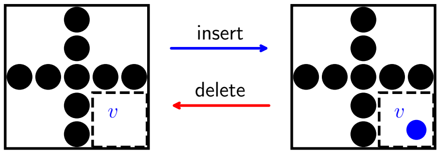

Figure 12. Insertion and deletion trials in a targeted region, v. Similar to Fig. 11, the targeted region could be carefully chosen to avoid excluded volume overlap with a rigid adsorbent framework.

Forward	$\alpha_{o\to n}$
Choose insert	1/2
Choose position in v	$\mathrm{d}\mathbf{r}/v$
Choose orientation	$P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose delete	1/2
Choose particle of type i in v	$1/(n_i+1)$

Table 19. Particle insertion transition probabilities in a targeted region.

Substituting Eq. 27 into the first ratio in Eq. 4, and using Table 19 for the second ratio in Eq. 4, results in

$$\chi = \frac{v z_i}{P_{\omega n}(n_i+1)} e^{-\beta \Delta U}, \tag{55}$$

where n_i is defined as the number of particles of type i inside v at the beginning of the insertion trial (i.e., before insertion). The transition probability of a deletion trial as the forward transition is then summarized in Table 20.

Substituting Eq. 27 into the first ratio in Eq. 4, and using Table 20 for the second ratio in Eq. 4, results in

$$\chi = \frac{n_i P_{\omega o}}{v z_i} e^{-\beta \Delta U}, \tag{56}$$

where n_i is defined as the number of particles of type i inside v at the beginning of the deletion trial (i.e., before deletion).

Forward	$\alpha_{o\to n}$
Choose delete	1/2
Choose particle of type i in v	$1/n_i$
Reverse	$\alpha_{n\to o}$
Choose insert	1/2
Choose position in v	$\mathrm{d}\mathbf{r}/v$
Choose orientation	$P_{\omega o}\,\mathrm{d}\boldsymbol{\omega}$

Table 20. Particle deletion transition probabilities in a targeted region.

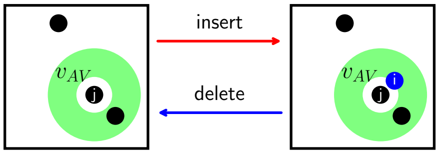

Figure 13. Insertion and deletion of a blue particle of type *i* within the aggregation volume (AV), shown in green, of an existing particle of type *j*.

10.6 Aggregation volume bias (AVB)

AVB insertions and deletions specifically target the positions near other molecules [107] in order to improve sampling for aggregated clusters of molecules, as shown in Fig. 13. In these derivations, aggregation volumes (AVs) are defined uniquely by one specific interaction site on a target particle and one specific site on a mobile particle. A single site should be specifically defined, not a type of site when there are multiple sites of the same type, or else detailed balance may be broken. Instead, if multiple AVs are desired in a single target particle, then simply treat each as a separate trial. For an isotropic AV, the mobile particle is inside the AV of the target particle if the distance between the two sites is between r_{inner} and r_{outer}.

An attempted AVB insertion and deletion proceeds as follows. First, an insertion or deletion trial is chosen randomly with equal probability. If a particle insertion is chosen, then the new position, $\mathbf{r}_n$, of an interaction site of the new particle of type i is placed at a random position selected uniformly within the AV, v_{AV}, of a different particle of type j that is randomly chosen from among N_j particles. The AV, in this example, is defined using the spherical shell between r_{inner} and r_{outer}, although it is possible to define an orientation-specific AV [108]. See Section 12.2.3 for how to generate randomly uniform points in a spherical shell. The remaining sites in the new particle may be oriented randomly about $\mathbf{r}_n$ or regrown with CB. If a particle deletion is chosen, then a particle of type j is randomly chosen. A random particle of type i within the AV of the chosen particle of type j is removed from among the N_i^{AV} particles of type i within the AV. The transition probabilities with insertion as the forward transition is summarized in Table 21.

If particle type i and j are identical, then $\delta_{ij} = 1$. Otherwise, $\delta_{ij} = 0$. Substituting Eq. 27 into the first ratio in Eq. 4, and using Table 21 for the second ratio in Eq. 4, results in

$$\chi = \frac{N_j}{N_j + \delta_{ij}} \left[\frac{v_{AV} z_i}{P_{\omega n}(N_i^{AV} + 1)} \right] e^{-\beta \Delta U}. \quad (57)$$

Forward	$\alpha_{o \to n}$
Choose insert	1/2
Choose particle of type *j*	$1/N_j$
Choose $\mathbf{r}_n$ in AV	$d\mathbf{r}/v_{AV}$
Choose orientation	$P_{\omega n}\, d\boldsymbol{\omega}$
Reverse	$\alpha_{n \to o}$
Choose delete	1/2
Choose particle of type *j*	$1/(N_j + \delta_{ij})$
Choose interaction site in AV	$1/(N_i^{AV} + 1)$

Table 21. AVB particle insertion transition probabilities.

Forward	$\alpha_{o \to n}$
Choose delete	1/2
Choose particle of type *j*	$1/N_j$
Choose interaction site in AV	$1/N_i^{AV}$
Reverse	$\alpha_{n \to o}$
Choose insert	1/2
Choose particle of type *j*	$1/(N_j - \delta_{ij})$
Choose $\mathbf{r}_o$ in AV	$d\mathbf{r}/v_{AV}$
Choose orientation	$P_{\omega o}\, d\boldsymbol{\omega}$

Table 22. AVB particle deletion transition probabilities.

Deletions are derived in an identical fashion, except that N_i^{AV} is evaluated from the perspective of the state before the deletion (e.g., after insertions), and the trial is summarized in Table 22. The deletion acceptance is obtained by substituting Eq. 27 into the first ratio in Eq. 4, and using Table 22 for the second ratio in Eq. 4, resulting in

$$\chi = \frac{N_j}{N_j - \delta_{ij}} \left(\frac{N_i^{AV} P_{\omega o}}{v_{AV} z_i} \right) e^{-\beta \Delta U}. \quad (58)$$

11 Biased canonical ensemble (*NVT*) trials

In this section, biased *NVT* trials are derived. First, we consider AVB displacement trials. Displacement trials include both linear and angular displacement. There are a few variants of AVB to consider. Although inefficiencies were found in the original version [74, 109, 110], the second variant, AVBMC2, efficiently moves particles in and out of the AV [107]. A third variant, AVBMC3, allows for particles to move from the AV of one particle into the AV of another particle (denoted as "in → in") [107]. First, we introduce AVBMC2. Then, we derive the "in → in" part of AVBMC3, and then introduce a relatively new variant of AVBMC3 that we call AVBMC4 [111], which is very similar to AVBMC3 but

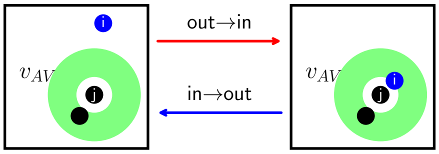

Figure 14. AVBMC2 displacement of a blue particle of type *i* from outside to inside the AV, v_{AV} shown in green, of an existing target particle of type *j*, and the reverse trial.

allows for overlapping AVs. After AVB, we also derive CB and DCCB displacements in the *NVT* ensemble.

11.1 Displacement with aggregation volume bias (AVBMC2)

In AVBMC2, a target particle of type *j* is chosen randomly from N_j particles. The AV is then defined about one specific site in the target particle. The mobile particle with one specific site then attempts a move into or out of the AV, with N_i particles of this type. The target and mobile particles may be the same. In this case, $\delta_{ij} = 1$, otherwise, $\delta_{ij} = 0$. At the beginning of the trial, there are N_i^{AV} particles of type *i* inside the AV of the specific site in the chosen particle of type *j*. AVBMC2 is an *NVT* trial because the total number of particles does not change.

In the AVB algorithm, it is possible for a particle to be inserted into an AV such that it happens to fall in the AV of two other particles, so that there would be more than one way to transition to the same microstate. The transition probabilities do not need to account for this circumstance, because super-detailed balance is satisfied for each path. The unbonding bonding (UB) algorithm [109] presents a variation of the AVB algorithm that more explicitly accounts for these cases.

The AVBMC2 trial proceeds as follows. A target particle of type *j* is chosen randomly from N_j and the AV is defined about the specific site in the chosen particle. With probability P_{bias}, an "out → in" trial is performed. Otherwise, an "in → out" trial is performed. For "out → in," one of N_i particles of type *i* is chosen randomly that has the specific site in *i* that is not in the AV. The chosen mobile site of the particle of type *i* is then placed at a random position selected uniformly in the AV. If the particle has other sites, they may then be regrown with CB or randomly rotated. For "in → out," one of the N_i^{AV} sites in the AV is placed randomly in the entire simulation domain that is not in the AV. Again, the remainder of the particle may then be oriented randomly or regrown with CB.

The transition probabilities for AVBMC2 are summarized in Table 23 for a forward transition of "out → in." The reverse move always has $N_i^{AV}+1$ in the AV, where N_i^{AV} is the number of mobile sites in particles of type *m* in the AV at the beginning of the trial.

Forward	$\alpha_{o\to n}$
Choose particle of type *j*	$1/N_j$
Choose "out → in"	P_{bias}
Choose mobile site not in AV	$1/(N_i - N_i^{AV} - \delta_{ij})$
Choose $\mathbf{r}_n$ in AV	$\mathrm{d}\mathbf{r}/v_{AV}$
Choose orientation	$P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose particle of type *j*	$1/N_j$
Choose "in → out"	$1 - P_{bias}$
Choose mobile site in AV	$1/(N_i^{AV} + 1)$
Choose $\mathbf{r}_o$ outside AV	$\mathrm{d}\mathbf{r}/(V - v_{AV})$
Choose orientation	$P_{\omega o}\,\mathrm{d}\boldsymbol{\omega}$

Table 23. Particle displacement AVBMC2 "out → in" transition probabilities.

Forward	$\alpha_{o\to n}$
Choose particle of type *j*	$1/N_j$
Choose "in → out"	$1 - P_{bias}$
Choose mobile site in AV	$1/N_i^{AV}$
Choose $\mathbf{r}_n$ outside AV	$\mathrm{d}\mathbf{r}/(V - v_{AV})$
Choose orientation	$P_{\omega n}\,\mathrm{d}\boldsymbol{\omega}$
Reverse	$\alpha_{n\to o}$
Choose particle of type *j*	$1/N_j$
Choose "out → in"	P_{bias}
Choose mobile site not in AV	$1/(N_i - N_i^{AV} - \delta_{ij} + 1)$
Choose $\mathbf{r}_o$ in AV	$\mathrm{d}\mathbf{r}/v_{AV}$
Choose orientation	$P_{\omega o}\,\mathrm{d}\boldsymbol{\omega}$

Table 24. Particle displacement AVBMC2 "in → out" transition probabilities.

Using Table 23 and substituting Eq. 6 into Eq. 4, the acceptance probability is

$$\chi = \frac{P_{\omega o}(1 - P_{bias})(N_i - N_i^{AV} - \delta_{ij})v_{AV}}{P_{\omega n}P_{bias}(N_i^{AV} + 1)(V - v_{AV})}e^{-\beta\Delta U}. \tag{59}$$

For an example of combining AVB with CB, see the Appendix of Ref. [112].

The transition probabilities for "in → out" as the forward transition are summarized in Table 24. Using Table 24 and substituting Eq. 6 into Eq. 4, the acceptance probability is

$$\chi = \frac{P_{\omega o}P_{bias}N_i^{AV}(V/v_{AV} - 1)e^{-\beta\Delta U}}{P_{\omega n}(1 - P_{bias})(N_i - N_i^{AV} - \delta_{ij} + 1)}. \tag{60}$$

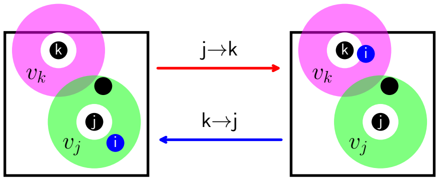

Figure 15. AVBMC4 displacement of a blue particle of type *m* from inside the AV, v_j of an existing target particle of type *j*, into the AV of an existing target particle *k*, v_k, and the reverse trial. AVBMC3 does not allow overlapping AV as shown here [107].

11.2 Displacement from one aggregation volume to another (AVBMC3/4)

In addition to moves inside and outside of AVs, such as AVBMC2 as described in Section 11.1, moves from one AV to another are also possible, as done in AVBMC3 [107]. In this case, two target particles are randomly chosen, and a third mobile particle is moved from the AV of the first target into the AV of the second target. In the original formulation, AVBMC2 type moves were also considered. To simplify the acceptance criteria, the AV of the two target particles also could not overlap. In this section, we describe a variation of AVBMC3, here called AVBMC4 [111], with two subtle differences from AVBMC3. First, AVBMC4 does not contain AVBMC2 “in → out" or "out → in" moves. Second, AVBMC4 allows the AV of the target particles to overlap, as long as the two target particles are not inside the AV of the other.

The AVBMC4 trial proceeds as follows. Two target particles are chosen randomly from all particles of type *j* and *k*, respectively. If types *j* and *k* are the same, then $\delta_{jk} = 1$ and particle *k* is chosen randomly from $N_k - 1$ particles because it cannot be the identical particle *j*. The AV of the *j* particle, v_j, is defined about one specific site in the *j* particle. Similarly, the AV of the *k* particle, v_k, is defined about a specific site in the *k* particle. Choose a $k \to j$ attempt with probability P_{bias}, otherwise attempt a $j \to k$ trial. For a $j \to k$ attempt, a mobile particle of type *i* that has a specific site inside v_j, but is not the same particle as the first two chosen *j* and *k* particles, is then moved inside v_k. If the *i* particle has multiple sites or orientation, then it should be randomly oriented or randomly regrown with CB. For a $k \to j$ attempt, a mobile particle of type *i* that has a specific site inside v_k, but is not the same particle as the first two chosen *j* and *k* particles, is then moved inside v_j.

Because v_j and v_k may overlap in AVBMC4, there are a few extra caveats to consider. If the interaction site in particle *i* is already inside v_k at the beginning of the trial, then $\Delta_{ik} = 1$ and there are $N_i^{in,k} - \Delta_{ik} + 1$ available sites to choose the mobile particle in the reverse move from v_k to v_j. Furthermore, the two targets, *j* and *k*, cannot be inside the AV of each other, or the trial is rejected. This trial is summarized in Table 25.

Forward	$\alpha_{o \to n}$
Choose $j \to k$	$1 - P_{bias}$
Choose particle of type *j*	$1/N_j$
Choose particle of type *k*	$1/(N_k - \delta_{jk})$
Choose mobile site in v_j	$1/N_i^{in,j}$
Choose $\mathbf{r}_n$ in v_k	$d\mathbf{r}/v_k$
Choose orientation	$P_{\omega n}\, d\boldsymbol{\omega}$

Reverse	$\alpha_{n \to o}$
Choose $k \to j$	P_{bias}
Choose particle of type *j*	$1/N_j$
Choose particle of type *k*	$1/(N_k - \delta_{jk})$
Choose mobile site in v_k	$1/(N_i^{in,k} - \Delta_{ik} + 1)$
Choose $\mathbf{r}_o$ in v_j	$d\mathbf{r}/v_j$
Choose orientation	$P_{\omega o}\, d\boldsymbol{\omega}$

Table 25. Particle displacement from one AV to another (AVBMC3/4) transition probabilities.

Using Table 25 and substituting Eq. 6 into Eq. 4, the acceptance probability for the $j \to k$ attempt as the forward transition is

$$\chi = \frac{P_{bias}}{1 - P_{bias}} \left(\frac{v_k P_{\omega o}}{v_j P_{\omega n}} \right) \frac{N_i^{in,j}}{N_i^{in,k} - \Delta_{ik} + 1} e^{-\beta \Delta U}. \tag{61}$$

The acceptance probability for the $k \to j$ attempt as the forward transition is obtained by perturbing the *j* and *k* in Eq. 61 and inverting the P_{bias} term to yield

$$\chi = \frac{1 - P_{bias}}{P_{bias}} \left(\frac{v_j P_{\omega o}}{v_k P_{\omega n}} \right) \frac{N_i^{in,k}}{N_i^{in,j} - \Delta_{ij} + 1} e^{-\beta \Delta U}. \tag{62}$$

11.3 ~~Displacement~~ Translation with configurational bias (CB)

While configurational bias (CB) is often used to grow complex molecules, CB may also be used to ~~displace~~ translate particles. Although this may not be the most efficient choice for many applications, it is instructive to consider how CB is derived. In addition, dual-cut configurational (DCCB) [72] may make this an efficient trial. To our knowledge, CB on single site ~~displacements~~ translation has not been published before.

A CB ~~displacement~~ translation proceeds as follows. A particle is randomly chosen among all mobile, non-fixed particles in the old microstate, $N_m = \sum_i^{mobile} N_i$, where *i* is the particle type. The position of a specific site in the chosen particle is $\mathbf{r}_o$ at the beginning of the trial. Then, *c* new positions, $\mathbf{r}_n^i$, are placed at random positions selected uniformly within $\pm\delta/2$

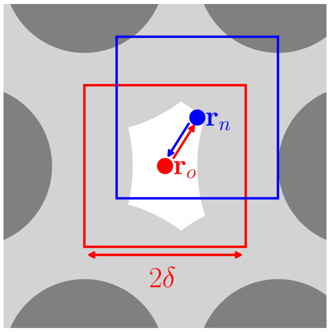

Figure 16. An illustration of the old and new Rosenbluth terms for hard circles as described in Fig. 2. The red box shows the positions available to π_{cn} while the blue box shows the positions available to π_{co}. Although the entire free volume is available to π_{cn}, it is not all available to π_{co}. Thus, $\pi_{cn} \neq \pi_{co}$.

Forward	$\alpha_{o\to n}$
Choose from N_m	$1/N_m$
Choose c positions about $\mathbf{r}_o$. Probability that $\mathbf{r}_n$ is in c.	$c\,\mathrm{d}\mathbf{r}/\delta^D$
Choose $\mathbf{r}_n$ in c positions	π_{cn}
~~Choose orientation $P_{\omega n}$ dω~~ height**Reverse**	$\alpha_{n\to o}$
Choose from N_m	$1/N_m$
Choose c positions about $\mathbf{r}_n$. Probability that $\mathbf{r}_o$ is in c.	$c\,\mathrm{d}\mathbf{r}/\delta^D$
Choose $\mathbf{r}_o$ in c positions	π_{co}
~~Choose orientation $P_{\omega o}$ dω~~	

Table 26. ~~Displacement~~ Translation with CB transition probabilities.

of $\mathbf{r}_o$ in each of D dimensions. One of the c positions are chosen with probability π_{cn}, and the particle is rigidly ~~displaced~~ translated to that position with fixed orientation. The π_{cn} may be a function of the positions of the particles, and this choice is the namesake of the CB method [9]. The reverse trial is to choose the same particle of type i, and ~~displace~~ translated from $\mathbf{r}_n$ to $\mathbf{r}_o$ after choosing from among c positions with probability π_{co}. The probability π_{co} is used in choosing a new position about $\mathbf{r}_n$, while π_{cn} is used in choosing a new position about $\mathbf{r}_o$. Fig. 16 shows an illustrative example of $\pi_{co} \neq \pi_{cn}$.

The transition probabilities are summarized in Table 26. Using Table 26 and substituting Eq. 6 into Eq. 4, the acceptance probability is

$$\chi = \frac{P_{\omega o}}{P_{\omega n}} \frac{\pi_{co}}{\pi_{cn}} e^{-\beta\Delta U}. \tag{63}$$

There is remarkable flexibility in the choice of c positions, π_{cn}. An often-used functional form is given by

$$\pi_{cn} = \frac{e^{-\beta U(\mathbf{r}_n)}}{W_{cn}}, \tag{64}$$

where $W_{cn} = \sum_{in}^{c} e^{-\beta U(\mathbf{r}_{in})}$ and the subscript in denotes that the positions, $\mathbf{r}_{in}$, were placed at random positions selected uniformly about $\mathbf{r}_o$ and contain $\mathbf{r}_n$. Similarly, π_{co} is given by

$$\pi_{co} = \frac{e^{-\beta U(\mathbf{r}_o)}}{W_{co}} \tag{65}$$

where $W_{co} = \sum_{io}^{c} e^{-\beta U(\mathbf{r}_{io})}$ and the subscript io denotes that the positions, $\mathbf{r}_{io}$, were placed at random positions selected uniformly about $\mathbf{r}_n$ and contain $\mathbf{r}_o$.

Substituting Eqs. 64 and 65 into Eq. 63 gives

$$\chi = \frac{P_{\omega o}}{P_{\omega n}} \frac{W_{cn}}{W_{co}} e^{-\beta[\Delta U - U(\mathbf{r}_n) + U(\mathbf{r}_o)]}. \tag{66}$$

This form of π_{cn} is convenient because $\Delta U - U(\mathbf{r}_n) + U(\mathbf{r}_o)$ contains only interaction energies beyond the center of the first site (e.g., if there were multiple sites).

The efficiency of ~~this method depends on the relative computational cost of the orientation term, $\frac{P_{\omega o}}{P_{\omega n}}$, to the translational term, $\frac{W_{co}}{W_{cn}}$. For~~ translation CB is debatable and depends upon the model system. Standard translations for single-site isotropic particles ~~, $P_{\omega o} = P_{\omega n}$ and standard displacements~~ described in Section ~~??~~ 5.1 may be more optimal for the following reason. Assuming that the calculation of the energy takes the majority of the CPU time, a standard ~~displacement~~ translation requires computing the energy twice (or only once if the old contribution is stored). In comparison, this choice of π_{cn} and π_{co} requires computing the energy $2c$ times (minus one if the old contribution is stored). Thus, in the CPU time it takes to complete one of these CB trials, approximately c standard trials could have been attempted, each with their own acceptance probability. In this case, its possible more than one of the c standard trials could have been accepted, in which case the standard approach would be more efficient.

On the other hand, ~~if the orientation term, $\frac{P_{\omega o}}{P_{\omega n}}$, is comparable or more computationally expensive than the translational term, $\frac{W_{cn}}{W_{co}}$, then this trial may be an efficient choice. In addition,~~ other choices of π_{cn} and π_{co} could make CB ~~useful~~ translation efficient, even for single-site particles, as we will ~~shown~~ show in the next section.

11.4 ~~Displacement~~ Translation with dual-cut configurational bias (DCCB)

A more efficient choice of π_{cn} and π_{co} is the DCCB approach if a suitable reference potential may be utilized [72]. The namesake of DCCB is to apply a shorter cutoff to the energy calculation in π_{cn} and π_{co}, but here we generalize dual-cut to include

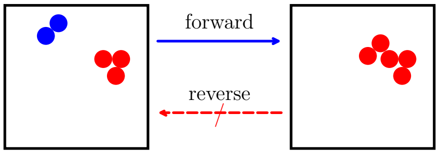

Figure 17. Detailed balance is violated if two separate clusters coalesce during a rigid cluster translation. In the forward transition, the blue cluster is ~~displaced~~ translated near a separate cluster shown in red. If this move were allowed, the reverse cluster move would be impossible because clusters are moved rigidly and cannot be broken apart during the trial. To obey detailed balance, reject rigid cluster moves that result in the coalescence of two originally separate clusters.

any reference potential, U_r, that takes less CPU time than the full potential, as discussed in Section 10.1. For DCCB,

$$\pi_{cn} = \frac{e^{-\beta U_r(\mathbf{r}_n)}}{W_{cnr}}, \qquad (67)$$

where $W_{cnr} = \sum_{in}^{c} e^{-\beta U_r(\mathbf{r}_{in})}$ and the subscript *in* denotes that the positions, $\mathbf{r}_{in}$, were placed at random positions selected uniformly about $\mathbf{r}_o$. The acceptance criteria is then obtained by substituting Eq. 67 for both π_{cn} and π_{co} in Eq. 63,

$$\chi = \frac{P_{\omega o}}{P_{\omega n}} \frac{W_{cnr}}{W_{cor}} e^{-\beta[\Delta U - U_r(\mathbf{r}_n) + U_r(\mathbf{r}_o)]}. \qquad (68)$$

Assuming that U_r results in the choice of likely accepted configurations by approximating the highest-contributing parts of the full potential, the sampling efficiency may be increased. On the other hand, a poor choice of U_r could decrease the simulation efficiency. In the worst case, ergodic sampling is broken by the rejection of valid configurations due to a poor reference potential, and the simulation results could be wrong. Despite this, DCCB is an efficient method that may be tested and benchmarked against simulations without DCCB.

11.5 ~~Displacement~~ Translation and rotation of rigid clusters

The ~~displacement~~ translation of rigid clusters is very similar to that of single particles described in Section ??5.1, except for the key differences that a cluster criteria must be specified, and that particles identified as part of a cluster must stay in that cluster after the trial. If the clusters are allowed to change, then detailed balance is violated, as shown in Fig. 17. Rigid cluster trial moves are biased because the transition probabilities depend on the configuration (i.e., the cluster criteria).

A rigid cluster ~~displacement~~ translation or rotation proceeds as follows. First, an individual cluster is identified. This identification could take many forms. For example, one particle could be randomly chosen for a given type, and then a flood-fill algorithm could be used to find all particles within a distance cutoff from all particles in the cluster [112]. Regardless of the metric used to identify clusters, the same metric must be applied again at the end of the trial. In addition, if a cluster has picked up or lost any particles, then the trial must be immediately rejected. Clusters may be rigidly ~~displaced~~ translated or rotated with a parameter that is tunable during equilibration but must stay constant during production simulations, as described in Section ??5.1. The acceptance criteria is then given by Eq. ??9 or 12.

12 Implementation issues in MC molecular simulations

Although the derivation of the acceptance criteria for MC trials is the focus of this article, common errors encountered during the implementation of MC are included here. For a more comprehensive discussion of the inner workings of MC see Ref. [113]. Thus, the scope of this section is for testing MC trial implementations. In this section, we discuss energy tests and machine precision issues for helping to capture common errors, the calculation of standard deviations using the block method [114, 115], the generation of a random position on the surface of a unit sphere and spherical shell, and the generation of random ~~quaternions and~~ rotation matrices. This section then ends with a discussion of the general workflow for testing a new MC trial.

12.1 Energy tests and machine precision

MC often involves the perturbation of a subset of the system instead of the entire system. Only the energy of interaction of the perturbed particles needs to be computed to obtain the change in the energy of the system. The energy of the system after a trial is accepted is the previous energy of the system plus the change. This energy is referred to as the running total energy. However, round off errors at the limit of numerical precision cause the running total energy to be different from the total energy calculated based on the interactions of all the particles. This difference increases with the number of trials. It is recommended to infrequently compute the energy of the entire system and compare the total energy with the running total energy from each energy change. The total energy may also be compared and updated during trials which update all the particles, such as volume changes. If the relative differences are not within an acceptable tolerance based on machine precision and the magnitude of the total energy, then this is a clear indication there is a problem with the way that the change in energy was computed, and an error or warning should be made visible. If the relative difference is within an acceptable tolerance, the running total energy is then reset to the newly calculated value and the simulation continues.

Another recommended test is to not use any optimizations when infrequently computing the total energy of the entire system as part of the energy test described above. Infrequent in this context essentially means much less often than once every *N* trials, where *N* is the total number of particles, assuming each trial moves individual particles, such that the computational cost of the trials is much greater than the computational cost of the infrequent and unoptimized total energy calculations. If this unoptimized calculation of the entire system is done infrequently, it will not slow down the simulation significantly. But it will serve as a test of the optimizations and may reveal issues with neighbor or cell lists, for example.

Machine precision round off errors may cause a number of other problems. One that is frequently encountered in software implementations of MC is the acceptance probabilities. Acceptance probabilities may involve several factors which are both big and small. For this reason, it is best to compute and store acceptance probabilities as the natural logarithm, $\ln\chi$, and not χ [23]. Machine precision issues may also appear when selecting among Rosenbluth weights. In the FEASST [30] implementation, overflow is avoided by shifting by a constant inside the exponential terms based on the maximum value, and later correcting for the shift. Also, the total Rosenbluth factor may reveal if there is a very small chance for the trial to be accepted. In that case, the trial may be outright rejected, because machine precision can make it difficult to select from among the poor choices.

12.2 Additional code

The following three sections contain Python code examples that were used in this article. The first is for computing standard deviations of the mean using block averaging, which was utilized in the ideal gas tests to compare ensemble average properties with theoretical expectations. The second is for randomly generating positions on a unit sphere, which is used for orienting molecules and anisotropic particles. The third is for randomly generating positions in a spherical shell, which is used in AVB algorithms.

12.2.1 Block averages

In the example shown in Code Block 10, the standard deviation is computed using the blocking method [114]. In this case, only blocks of 100 samples are considered, but a careful analysis would show that the resulting standard deviation is independent of the block size (when large enough). Thus, with this implementation, we are assuming that the sampled states are independent after 100 MC trials, which may be reasonable for an ideal gas but most likely not for a dense liquid. See the blocking tutorial in ~~math plugin~~ python interface of FEASST [30] for an example with computing the block standard deviation from the average position of a particle randomly walking in one dimension with periodic boundary conditions. If the maximum ~~trial displacement~~ translation is smaller than the periodic boundary, correlations in the position may persist over many trials.

Code Block 10. The statistics module computes the mean and the standard deviation of the mean using blocks.

```
import numpy as np
def block(values, block_size=100, equil=1000):
    blocks = [np.average(block) for block in
              np.array_split(values[equil:], block_size)]
    return str(np.average(values[equil:])) + ' stdev ' + \
           str(np.std(blocks)/np.sqrt(len(blocks)))
```

12.2.2 Random position on a unit sphere surface and random orientation

~~In the example shown in Code Block 11, positions are generated~~ Situations arise where one needs to generate a point at random uniformly on the surface of a sphere in 3D. This is ~~an important case to consider, because MC trials often assume that particles may be randomly oriented . However, one must be careful in how an orientation is generated. Uniformly selected values of Euler angles, for example, will result in a bias along the poles.~~ equivalent to defining an isotropically random direction or a randomly oriented unit vector. There are several ways to approach this: (1) sample the spherical coordinates θ and ϕ, appropriately weighted; (2) use a rejection method to generate a point uniformly inside a sphere, i.e., repeatedly sample in a cube that circumscribes the sphere until the sample is also in the sphere, and then normalize the sampled point to a unit vector; (3) sample the *x*, *y* and *z* coordinates on independent Gaussian distributions, which yields a point sampled from a spherically symmetric 3D probability density, and normalize to a unit vector. The first method listed here is illustrated in Code Block 11.

A similar problem is sampling a quaternion, which may be interpreted as a point on the surface of a unit 4D-hypersphere. Accordingly, Code Block 12 ~~also~~ shows how quaternions may be randomly generated ~~to represent a uniformly random orientation of a point~~ by sampling uniformly on the surface of a ~~sphere~~ 4D hypersphere using an efficient rejection method [79]. Code Block 13 uses these randomly generated quaternions to compute rotation matrices~~. A~~, which could be used to generate a fully random orientation by rotating from an arbitrary reference orientation. Finally, a randomly generated perturbation of

Code Block 11. The unit_sphere module generates randomly uniform positions on the surface of a unit sphere in three dimensions.

```
import random
import numpy as np
import matplotlib.pyplot as plt

def uniform_surface(radius=1):
    theta = 2*np.pi*random.uniform(0, 1)
    phi = np.arccos(random.uniform(-1, 1))
    return [radius*np.sin(phi)*np.cos(theta),
            radius*np.sin(phi)*np.sin(theta),
            radius*np.cos(phi)]

def plot(ax):
    for i in range(int(1e4)):
        x = uniform_surface()
        ax.plot([0, x[0]], [0, x[1]], [0, x[2]],
                color='blue', alpha=0.025)
```

Code Block 12. The quaternion module generates a random quaternion by sampling on the surface of a 4-dimensional hypersphere.

```
import random
import numpy as np
def rand():
    q = np.empty(4)
    s1 = 2
    while s1 > 1:
      q[0] = random.uniform(-1, 1)
      q[1] = random.uniform(-1, 1)
      s1 = q[0]**2 + q[1]**2
    s2 = 2
    while s2 > 1:
      q[2] = random.uniform(-1, 1)
      q[3] = random.uniform(-1, 1)
      s2 = q[2]**2 + q[3]**2
    s1 = np.sqrt((1 - s1)/s2)
    q[2] *= s1
    q[3] *= s1
    return q
```

Code Block 13. The rotation module generates a rotation matrix from an axis and angle or a quaternion. The first index is the row and the second index is the column, each starting with an index of zero.

```
import numpy as np
def axis_angle(axis, angle):
    cos = np.cos(angle)
    sin = np.sin(angle)
    rtmx = np.empty(shape=(3,3))
    rtmx[0][0] = cos + (1-cos)*axis[0]**2
    rtmx[1][0] = (1-cos)*axis[0]*axis[1] + sin*axis[2]
    rtmx[2][0] = (1-cos)*axis[0]*axis[2] - sin*axis[1]
    rtmx[0][1] = (1-cos)*axis[0]*axis[1] - sin*axis[2]
    rtmx[1][1] = cos + (1-cos)*axis[1]**2
    rtmx[2][1] = (1-cos)*axis[1]*axis[2] + sin*axis[0]
    rtmx[0][2] = (1-cos)*axis[0]*axis[2] + sin*axis[1]
    rtmx[1][2] = (1-cos)*axis[1]*axis[2] - sin*axis[0]
    rtmx[2][2] = cos + (1-cos)*axis[2]**2
    return rtmx

def quaternion(q):
    rtmx = np.empty(shape=(3,3))
    rtmx[0][0] = q[0]*q[0] - q[1]*q[1] - q[2]*q[2] +q[3]*q[3]
    rtmx[1][0] = 2*(q[0]*q[1] + q[2]*q[3])
    rtmx[2][0] = 2*(q[2]*q[0] - q[1]*q[3])
    rtmx[0][1] = 2*(q[0]*q[1] - q[2]*q[3])
    rtmx[1][1] = q[1]*q[1] - q[2]*q[2] - q[0]*q[0] +q[3]*q[3]
    rtmx[2][1] = 2*(q[1]*q[2] + q[0]*q[3])
    rtmx[0][2] = 2*(q[2]*q[0] + q[1]*q[3])
    rtmx[1][2] = 2*(q[1]*q[2] - q[0]*q[3])
    rtmx[2][2] = q[2]*q[2] - q[0]*q[0] - q[1]*q[1] +q[3]*q[3]
    return rtmx
```

Code Block 14. Visualize randomly uniform perturbations in three dimensional particle orientations.

```
import numpy as np
import matplotlib.pyplot as plt
import unit_sphere
import random
import rotation
random.seed(43279857)
ax = plt.figure().add_subplot(projection='3d')
unit_sphere.plot(ax)
for i in range(int(1e4)):
    angle = random.uniform(-np.pi/2, np.pi/2)
    axis = unit_sphere.uniform_surface()
    x = np.matmul(rotation.axis_angle(axis, angle), [0, 0, 1])
    ax.plot([0, x[0]], [0, x[1]], [0, x[2]],
            color='orange', alpha=0.025)
ax.plot([0,0], [0,0], [0,1], color='black')
plt.show()
```

orientation is ~~then shown in~~ shown using the axis-angle method in Code Block 14 and Fig. ~~??~~18.

12.2.3 Uniform position in a spherical shell

In the example shown in Code Block 15, positions are generated uniformly within a spherical shell. This is used to generate new trial positions with the AV of target isotropic sites, as shown in Fig. 19.

12.3 Workflow for testing a new MC trial

Another recommendation during testing of new trials is to start with a result that was obtained using an established and well-tested set of trials and known to high confidence. One source of such results is the National Institute of Standards and Technology (NIST) standard reference simulation website [116]. Another choice is to generate results with one

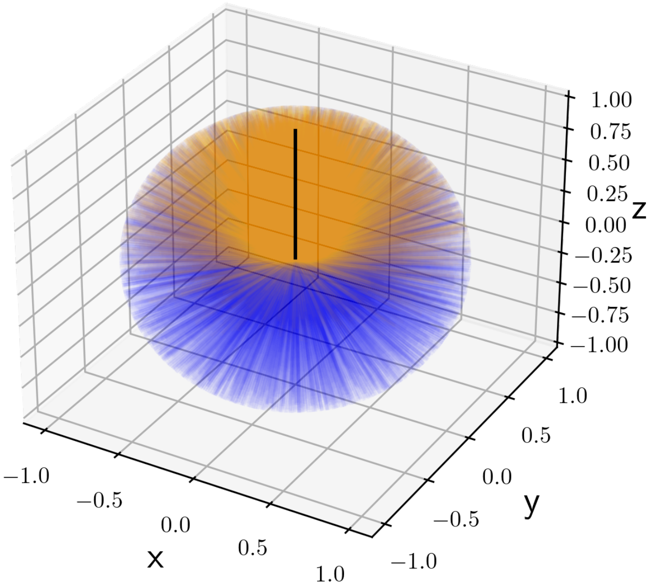

Figure 18. ~~Example~~ The output from Code Block 14 shows 10^3 uniformly random ~~points on a sphere in~~ blue ~~, and~~ unit vectors emanating from the origin. The other 10^3 ~~points in~~ orange unit vectors are confined to the top hemisphere because they were randomly generated ~~about~~ with the axis-angle method by perturbing the orientation of an upwards z-axis ~~using an orientational tunable~~ vector (shown in black) with a maximum angle parameter, ~~$\delta = 0.3$~~ $\delta\theta_{max} = \pi$ = 180 degrees.

Code Block 15. Visualize randomly uniform positions in a spherical shell in three dimensions.

```
import random
import numpy as np
import matplotlib.pyplot as plt
import unit_sphere
r_upper = 3
r_lower = 2
x = np.zeros(shape=(int(2e5), 3))
for i,_ in enumerate(x):
    radius = (random.uniform(0, 1)*(r_upper**3 − r_lower**3) +
              r_lower**3)**(1./3.)
    x[i] = unit_sphere.uniform_surface(radius)
ax = plt.figure().add_subplot(projection='3d')
x = x[x[:, 2] < 0]
ax.scatter(x[:, 0], x[:, 1], x[:, 2], s=1, alpha=0.05)
plt.show()
```

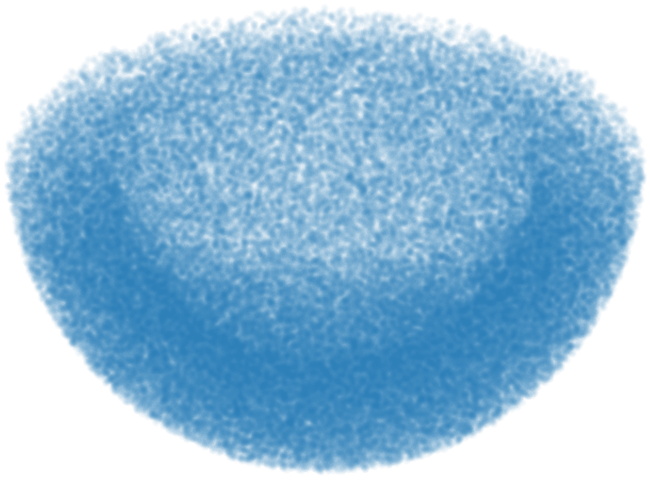

Figure 19. Example output from Code Block 15 shows approximately 10^5 small blue points randomly generated inside a spherical shell of outer radius 3 and inner radius 2. To see the inside cavity, points above z = 0 were removed.

or more well-tested, open-source simulation codes. Then, when computing the same result with the new trial, any differences may be further investigated. Start with the simplest model and thermodynamic conditions that may reveal differences, such as an ideal gas simulation. Also, if possible, completely remove as many other trials from the simulation with the new trial, thereby allowing one to better determine if the new trial is incorrect. Otherwise, the difference in the results due to the new trial may be difficult to detect, or the impact of the new trial difficult to isolate, because the other trials are contributing greatly. For example, AVB insertions and deletions can be tested without standard insertions and deletions, and without even *NVT* ~~displacement moves~~translation and rotation trials.

13 Conclusions

We provided a framework for deriving the Metropolis MC acceptance probabilities in the *NVT*, *NPT*, (semi)-μVT and Gibbs ensembles with a detailed checklist for generating unbiased trials in each of these ensembles. Discussions of how detailed balance may be violated, as well as code examples to compare ideal gas MC results with theoretical predictions, were provided. The checklist was applied with a table format for generating the transition probabilities to help reduce errors. We considered complex trials with bias, which include the illustrations and tables but not code examples. Finally, a discussion of common implementation issues in MC were included, along with code examples. It is our hope that this article will make MC molecular simulation methodology development more accessible in general to researchers. When implementing complex MC trials, we recommend first testing with a number of models and thermodynamic conditions, starting with the simplest and fastest ideal gas simulation.

We encourage readers to comment upon this guide using the issue tracker of the GitHub repository [https://github.com/usnistgov/best-practices-mc]. There are many more methods we would like to add to this guide. Future versions may include the following MC methods: orientation bias, Mayer sampling, expanded ensembles, geometric cluster algorithm, shell particles, bond, angle and branch algorithms and tests.

14 Acknowledgements and Disclaimer

This article was funded by NIST and is not subject to U.S. Copyright. Certain commercial firms and trade names are identified in order to specify the usage procedures adequately for reproducibility. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that related products are necessarily the best available for the purpose. Support to D.A.K. from the National Science Foundation (CBET-2152946) and the Schmid Family Foundation is gratefully acknowledged.

Author Information

ORCID:

Harold W. Hatch: 0000-0003-2926-9145

David S. Corti: 0000-0003-3289-3926

David A. Kofke: 0000-0002-2530-8816

Vincent K. Shen: none

References

[1] **Metropolis N**, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of State Calculations by Fast Computing Machines. J Chem Phys. 1953; 21(6):1087–1092. https://doi.org/10.1063/1.1699114.

[2] **Rosenbluth MN**, Rosenbluth AW. Further Results on Monte Carlo Equations of State. J Chem Phys. 1954; 22(5):881–884. https://doi.org/10.1063/1.1740207.

[3] **Rosenbluth MN**, Rosenbluth AW. Monte Carlo Calculation of the Average Extension of Molecular Chains. J Chem Phys. 1955; 23(2):356–359. https://doi.org/10.1063/1.1741967.

[4] **Wood WW**. Monte Carlo Calculations for Hard Disks in the Isothermal-Isobaric Ensemble. J Chem Phys. 1968; 48(1):415–434. https://doi.org/10.1063/1.1667938.

[5] **Norman GE**, Filinov VS. Investigation of Phase Transitions by a Monte-Carlo Method. High Temp. 1969; 7(10):216–222.

[6] **Adams DJ**. Grand Canonical Ensemble Monte Carlo for a Lennard-Jones Fluid. Mol Phys. 1975; 29(1):307–311. https://doi.org/10.1080/00268977500100221.

[7] **Kofke DA**, Glandt ED. Monte Carlo Simulation of Multicomponent Equilibria in a Semigrand Canonical Ensemble. Mol Phys. 1988; 64(6):1105–1131. https://doi.org/10.1080/00268978800100743.

[8] **Panagiotopoulos AZ**. Direct Determination of Phase Coexistence Properties of Fluids by Monte Carlo Simulation in a New Ensemble. Mol Phys. 1987; 61(4):813–826. https://doi.org/10.1080/00268978700101491.

[9] **Siepmann JI**, Frenkel D. Configurational Bias Monte Carlo: a New Sampling Scheme for Flexible Chains. Mol Phys. 1992; 75(1):59–70. https://doi.org/10.1080/00268979200100061.

[10] **Lyubartsev AP**, Martsinovski AA, Shevkunov SV, Vorontsov-Velyaminov PN. New Approach to Monte Carlo Calculation of the Free Energy: Method of Expanded Ensembles. J Chem Phys. 1992; 96(3):1776–1783. https://doi.org/10.1063/1.462133.

[11] **Escobedo FA**, Pablo JJd. Expanded Grand Canonical and Gibbs Ensemble Monte Carlo Simulation of Polymers. J Chem Phys. 1996; 105(10):4391–4394. https://doi.org/10.1063/1.472257.

[12] **Kofke DA**. Gibbs-Duhem Integration: a New Method for Direct Evaluation of Phase Coexistence by Molecular Simulation. Mol Phys. 1993; 78(6):1331–1336. https://doi.org/10.1080/00268979300100881.

[13] **Wang F**, Landau DP. Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States. Phys Rev Lett. 2001; 86(10):2050–2053. https://doi.org/10.1103/PhysRevLett.86.2050.

[14] **Fitzgerald M**, Picard RR, Silver RN. Canonical Transition Probabilities for Adaptive Metropolis Simulation. Europhys Lett. 1999; 46(3):282. https://doi.org/10.1209/epl/i1999-00257-1.

[15] **Errington JR**, Shen VK. Direct Evaluation of Multicomponent Phase Equilibria Using Flat-histogram Methods. J Chem Phys. 2005; 123(16):164103–164103–9. https://doi.org/doi:10.1063/1.2064628.

[16] **Singh JK**, Kofke DA. Mayer Sampling: Calculation of Cluster Integrals Using Free-Energy Perturbation Methods. Phys Rev Lett. 2004; 92(22):220601. https://doi.org/10.1103/PhysRevLett.92.220601.

[17] **Shi W**, Maginn EJ. Continuous Fractional Component Monte Carlo: An Adaptive Biasing Method for Open System Atomistic Simulations. J Chem Theory Comput. 2007; 3(4):1451–1463. https://doi.org/10.1021/ct7000039.

[18] **Martin MG**. MCCCS Towhee: a Tool for Monte Carlo Molecular Simulation. Mol Simul. 2013; 39(14-15):1212–1222. https://doi.org/10.1080/08927022.2013.828208.

[19] **Jorgensen WL**, Tirado–Rives J. Molecular Modeling of Organic and Biomolecular Systems Using BOSS and MCPRO. J Comput Chem. 2005; 26(16):1689–1700. https://doi.org/10.1002/jcc.20297.

[20] **Schultz AJ**, Kofke DA. Etomica: An Object-oriented Framework for Molecular Simulation. J Comput Chem. 2015; 36(8):573–583. https://doi.org/10.1002/jcc.23823.

[21] **Gupta A**, Chempath S, Sanborn MJ, Clark LA, Snurr RQ. Object-oriented Programming Paradigms for Molecular Modeling. Mol Simul. 2003; 29(1):29–46. https://doi.org/10.1080/0892702031000065719.

[22] **Brukhno AV**, Grant J, Underwood TL, Stratford K, Parker SC, Purton JA, Wilding NB. DL_MONTE: a Multipurpose Code for Monte Carlo Simulation. Mol Simul. 2021; 47(2-3):131–151. https://doi.org/10.1080/08927022.2019.1569760.

[23] **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(19):1727–1739. https://doi.org/10.1002/jcc.24807.

[24] **Lund M**, Trulsson M, Persson B. Faunus: An Object Oriented Framework for Molecular Simulation. Source Code Biol Med. 2008; 3(1):1. https://doi.org/10.1186/1751-0473-3-1.

[25] **Dubbeldam D**, Calero S, Ellis DE, Snurr RQ. RASPA: Molecular Simulation Software for Adsorption and Diffusion in Flexible Nanoporous Materials. Mol Simul. 2016; 42(2):81–101. https://doi.org/10.1080/08927022.2015.1010082.

[26] **Nejahi Y**, Soroush Barhaghi M, Mick J, Jackman B, Rushaidat K, Li Y, Schwiebert L, Potoff J. GOMC: GPU Optimized Monte Carlo for the Simulation of Phase Equilibria and Physical Properties of Complex Fluids. SoftwareX. 2019; 9:20–27. https://doi.org/10.1016/j.softx.2018.11.005.

[27] **Penaloza-Amion M**, Sedghamiz E, Kozlowska M, Degitz C, Possel C, Wenzel W. Monte-Carlo Simulations of Soft Matter Using SIMONA: A Review of Recent Applications. Front Phys. 2021; 9. https://doi.org/10.3389/fphy.2021.635959.

[28] **Limbach HJ**, Arnold A, Mann BA, Holm C. ESPResSo—an Extensible Simulation Package for Research on Soft Matter Systems. Comput Phys Commun. 2006; 174(9):704–727. https://doi.org/10.1016/j.cpc.2005.10.005.

[29] **Sun Y**, DeJaco RF, Siepmann JI. Deep Neural Network Learning of Complex Binary Sorption Equilibria from Molecular Simulation Data. Chem Sci. 2019; 10(16):4377–4388. https://doi.org/10.1039/C8SC05340E.

[30] **Hatch HW**, Siderius DW, Shen VK. Monte Carlo Molecular Simulations with FEASST Version 0.25.1. J Chem Phys. 2024; 161(9):092501. https://doi.org/10.1063/5.0224283.

[31] **Alder BJ**, Wainwright TE. Phase Transition for a Hard Sphere System. J Chem Phys. 1957; 27(5):1208–1209. https://doi.org/10.1063/1.1743957.

[32] **Talirz L**, Ghiringhelli LM, Smit B. Trends in Atomistic Simulation Software Usage [Article V1.0]. Living J Comput Mol Sci. 2021; 3(1):1483–1483. https://doi.org/10.33011/livecoms.3.1.1483.

[33] **Allen MP**, Tildesley DJ. Computer Simulation of Liquids. Clarendon Press; 1989.

[34] **Frenkel D**, Smit B. Understanding Molecular Simulation: From Algorithms to Applications. Academic Press; 2002.

[35] **Barhaghi MS**, Crawford B, Schwing G, Hardy DJ, Stone JE, Schwiebert L, Potoff J, Tajkhorshid E. Py-MCMD: Python Software for Performing Hybrid Monte Carlo/Molecular Dynamics Simulations with GOMC and NAMD. J Chem Theory Comput. 2022; 18(8):4983–4994. https://doi.org/10.1021/acs.jctc.1c00911.

[36] **Hénin J**, Lelièvre T, Shirts MR, Valsson O, Delemotte L. Enhanced Sampling Methods for Molecular Dynamics Simulations [Article V1.0]. Living J Comput Mol Sci. 2022; 4(1):1583–1583. https://doi.org/10.33011/livecoms.4.1.1583.

[37] **Ryckaert JP**, Ciccotti G, Berendsen HJC. Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of N-alkanes. J Comput Phys. 1977; 23(3):327–341. https://doi.org/10.1016/0021-9991(77)90098-5.

[38] **Andersen HC**. Rattle: A "velocity" Version of the Shake Algorithm for Molecular Dynamics Calculations. J Comput Phys. 1983; 52(1):24–34. https://doi.org/10.1016/0021-9991(83)90014-1.

[39] **Vakser IA**, Grudinin S, Jenkins NW, Kundrotas PJ, Deeds EJ. Docking-based Long Timescale Simulation of Cell-size Protein Systems at Atomic Resolution. Proc Natl Acad Sci USA. 2022; 119(41):e2210249119. https://doi.org/10.1073/pnas.2210249119.

[40] **Hatch HW**, Bergonzo C, Blanco MA, Yuan G, Grudinin S, Lund M, Curtis JE, Grishaev AV, Liu Y, Shen VK. Anisotropic Coarse-grain Monte Carlo Simulations of Lysozyme, Lactoferrin, and NISTmAb by Precomputing Atomistic Models. J Chem Phys. 2024; 161(9):094113. https://doi.org/10.1063/5.0224809.

[41] **Posch HA**, Hoover WG. Lyapunov Instability of Dense Lennard-Jones Fluids. Phys Rev A. 1988; 38(1):473–482. https://doi.org/10.1103/PhysRevA.38.473.

[42] **Duane S**, Kennedy AD, Pendleton BJ, Roweth D. Hybrid Monte Carlo. Phys Lett B. 1987; 195(2):216–222. https://doi.org/10.1016/0370-2693(87)91197-X.

[43] **Hartmann C**. An Ergodic Sampling Scheme for Constrained Hamiltonian Systems with Applications to Molecular Dynamics. J Stat Phys. 2008; 130(4):687–711. https://doi.org/10.1007/s10955-007-9470-2.

[44] **Ghoufi A**, Maurin G. Hybrid Monte Carlo Simulations Combined with a Phase Mixture Model to Predict the Structural Transitions of a Porous Metal-Organic Framework Material Upon Adsorption of Guest Molecules. J Phys Chem C. 2010; 114(14):6496–6502. https://doi.org/10.1021/jp911484g.

[45] **Spiridon L**, Minh DDL. Hamiltonian Monte Carlo with Constrained Molecular Dynamics As Gibbs Sampling. J Chem Theory Comput. 2017; 13(10):4649–4659. https://doi.org/10.1021/acs.jctc.7b00570.

[46] **Guo J**, Haji-Akbari A, Palmer JC. Hybrid Monte Carlo with LAMMPS. J Theor Comput Chem. 2018; 17(03):1840002. https://doi.org/10.1142/S0219633618400023.

[47] **Fass J**, Sivak DA, Crooks GE, Beauchamp KA, Leimkuhler B, Chodera JD. Quantifying Configuration-Sampling Error in Langevin Simulations of Complex Molecular Systems. Entropy. 2018; 20(5):318. https://doi.org/10.3390/e20050318.

[48] **Swendsen RH**, Wang JS. Replica Monte Carlo Simulation of Spin-Glasses. Phys Rev Lett. 1986; 57(21):2607–2609. https://doi.org/10.1103/PhysRevLett.57.2607.

[49] **Berg BA**, Neuhaus T. Multicanonical Algorithms for First Order Phase Transitions. Phys Lett B. 1991; 267(2):249–253. https://doi.org/10.1016/0370-2693(91)91256-U.

[50] **Hukushima K**, Nemoto K. Exchange Monte Carlo Method and Application to Spin Glass Simulations. J Phys Soc Jpn. 1996; 65(6):1604–1608. https://doi.org/10.1143/JPSJ.65.1604.

[51] **Sugita Y**, Okamoto Y. Replica-Exchange Molecular Dynamics Method for Protein Folding. Chem Phys Lett. 1999; 314(1–2):141–151. https://doi.org/10.1016/S0009-2614(99)01123-9.

[52] **Andersen HC**. Molecular Dynamics Simulations at Constant Pressure And/or Temperature. J Chem Phys. 1980; 72(4):2384–2393. https://doi.org/10.1063/1.439486.

[53] **Laio A**, Parrinello M. Escaping Free-energy Minima. Proc Natl Acad Sci USA. 2002; 99(20):12562–12566. https://doi.org/10.1073/pnas.202427399.

[54] **Basconi JE**, Shirts MR. Effects of Temperature Control Algorithms on Transport Properties and Kinetics in Molecular Dynamics Simulations. J Chem Theory Comput. 2013; 9(7):2887–2899. https://doi.org/10.1021/ct400109a.

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

[56] **Rao M**, Berne BJ. On the Force Bias Monte Carlo Simulation of Simple Liquids. J Chem Phys. 1979; 71(1):129–132. https://doi.org/10.1063/1.438111.

[57] **Liu J**, Luijten E. Rejection-Free Geometric Cluster Algorithm for Complex Fluids. Phys Rev Lett. 2004; 92(3):035504. https://doi.org/10.1103/PhysRevLett.92.035504.

[58] **Liu J**, Luijten E. Generalized Geometric Cluster Algorithm for Fluid Simulation. Phys Rev E. 2005; 71(6):066701. https://doi.org/10.1103/PhysRevE.71.066701.

[59] **Whitelam S**, Geissler PL. Avoiding Unphysical Kinetic Traps in Monte Carlo Simulations of Strongly Attractive Particles. J Chem Phys. 2007; 127(15):154101. https://doi.org/10.1063/1.2790421.

[60] **Ren R**, Orkoulas G. Acceleration of Markov Chain Monte Carlo Simulations Through Sequential Updating. J Chem Phys. 2006; 124(6):064109. https://doi.org/10.1063/1.2168455.

[61] **Anderson JA**, Eric Irrgang M, Glotzer SC. Scalable Metropolis Monte Carlo for Simulation of Hard Shapes. Comput Phys Commun. 2016; 204:21–30. https://doi.org/10.1016/j.cpc.2016.02.024.

[62] **Esselink K**, Loyens LDJC, Smit B. Parallel Monte Carlo Simulations. Phys Rev E. 1995; 51(2):1560–1568. https://doi.org/10.1103/PhysRevE.51.1560.

[63] **Vlugt TJH**. Efficiency of Parallel CBMC Simulations. Mol Simul. 1999; 23(1):63–78. https://doi.org/10.1080/08927029908022112.

[64] **Frenkel D**. Speed-up of Monte Carlo Simulations by Sampling of Rejected States. Proc Natl Acad Sci USA. 2004; 101(51):17571–17575. https://doi.org/10.1073/pnas.0407950101.

[65] **Michel M**, Kapfer SC, Krauth W. Generalized Event-chain Monte Carlo: Constructing Rejection-free Global-balance Algorithms from Infinitesimal Steps. J Chem Phys. 2014; 140(5):054116. https://doi.org/10.1063/1.4863991.

[66] **Brockwell AE**. Parallel Markov Chain Monte Carlo Simulation by Pre-Fetching. J Comput Graph Statist. 2006; 15(1):246–261. https://doi.org/10.1198/106186006X100579.

[67] **Calderhead B**. A General Construction for Parallelizing Metropolis-Hastings Algorithms. Proc Natl Acad Sci USA. 2014; 111(49):17408–17413. https://doi.org/10.1073/pnas.1408184111.

[68] **Hatch HW**. Parallel Prefetching for Canonical Ensemble Monte Carlo Simulations. J Phys Chem A. 2020; 124(35):7191–7198. https://doi.org/10.1021/acs.jpca.0c05242.

[69] **Floriano MA**, Caponetti E, Panagiotopoulos AZ. Micellization in Model Surfactant Systems. Langmuir. 1999; 15(9):3143–3151. https://doi.org/10.1021/la9810206.

[70] **Hatch HW**, Mittal J, Shen VK. Computational Study of Trimer Self-assembly and Fluid Phase Behavior. J Chem Phys. 2015; 142(16):164901. https://doi.org/10.1063/1.4918557.

[71] **Hatch HW**, Siderius DW, Errington JR, Shen VK. Efficiency Comparison of Single- and Multiple-Macrostate Grand Canonical Ensemble Transition-Matrix Monte Carlo Simulations. J Phys Chem B. 2023; 127(13):3041–3051. https://doi.org/10.1021/acs.jpcb.3c00613.

[72] **Vlugt TJH**, Martin MG, Smit B, Siepmann JI, Krishna R. Improving the Efficiency of the Configurational-bias Monte Carlo Algorithm. Mol Phys. 1998; 94(4):727–733. https://doi.org/10.1080/002689798167881.

[73] **Hatch HW**, Shen VK, Corti DS. Theory and Monte Carlo Simulation of the Ideal Gas with Shell Particles in the Canonical, Isothermal–isobaric, Grand Canonical, and Gibbs Ensembles. J Chem Phys. 2024; 161(8):084106. https://doi.org/10.1063/5.0224305.

[74] **Chen B**, Siepmann JI, Oh KJ, Klein ML. Aggregation-volume-bias Monte Carlo Simulations of Vapor-liquid Nucleation Barriers for Lennard-Jonesium. J Chem Phys. 2001; 115(23):10903–10913. https://doi.org/10.1063/1.1417536.

[75] **Hastings WK**. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika. 1970; 57(1):97–109. https://doi.org/10.1093/biomet/57.1.97.

[76] **Manousiouthakis VI**, Deem MW. Strict Detailed Balance Is Unnecessary in Monte Carlo Simulation. J Chem Phys. 1999; 110(6):2753–2756. https://doi.org/10.1063/1.477973.

[77] **Barker AA**. Monte Carlo Calculations of the Radial Distribution Functions for a Proton-Electron Plasma. Aust J Phys. 1965; 18(2):119–134. https://doi.org/10.1071/ph650119.

[78] **Kofke DA**, CE 530 Mol. Simul. Lecture 9: Monte Carlo Simulation; 2024. https://www.eng.buffalo.edu/~kofke/ce530/Lectures/lectures.html.

[79] **Vesely FJ**. Angular Monte Carlo Integration Using Quaternion Parameters: a Spherical Reference Potential for CCl4. J Comput Phys. 1982; 47(2):291–296. https://doi.org/10.1016/0021-9991(82)90080-8.

[80] **Karney CFF**. Quaternions in molecular modeling. J Mol Graph Model. 2007; 25(5):595–604. https://doi.org/10.1016/j.jmgm.2006.04.002.

[81] **Kolafa J**. On Optimization of Monte Carlo Simulations. Mol Phys. 1987; 63(4):559–579. https://doi.org/10.1080/00268978800100381.

[82] **Jacucci G**, Rahman A. Comparing the Efficiency of Metropolis Monte Carlo and Molecular-Dynamics Methods for Configuration Space Sampling. Il Nuovo Cimento D. 1984; 4(4):341–356. https://doi.org/10.1007/BF02451293.

[83] **Chapman W**, Quirke N. Metropolis Monte Carlo Simulation of Fluids with Multiparticle Moves. Physica B+C. 1985; 131(1):34–40. https://doi.org/10.1016/0378-4363(85)90137-8.

[84] **Mountain RD**, Thirumalai D. Quantative Measure of Efficiency of Monte Carlo Simulations. Physica A. 1994; 210(3):453–460. https://doi.org/10.1016/0378-4371(94)90092-2.

[85] **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):4508–4517. https://doi.org/10.1021/jp984742e.

[86] **Martin MG**, Frischknecht AL. Using Arbitrary Trial Distributions to Improve Intramolecular Sampling in Configurational-bias Monte Carlo. Mol Phys. 2006; 104(15):2439–2456. https://doi.org/10.1080/00268970600751078.

[87] **Sepehri A**, Loeffler TD, Chen B. Improving the Efficiency of Configurational-Bias Monte Carlo: Extension of the Jacobian–Gaussian Scheme to Interior Sections of Cyclic and Polymeric Molecules. J Chem Theory Comput. 2017; 13(9):4043–4053. https://doi.org/10.1021/acs.jctc.7b00478.

[88] **Sepehri A**, Loeffler TD, Chen B. Improving the Efficiency of Configurational-Bias Monte Carlo: A Jacobian–Gaussian Scheme for Generating Bending Angle Trials for Linear and Branched Molecules. J Chem Theory Comput. 2017; 13(4):1577–1583. https://doi.org/10.1021/acs.jctc.7b00173.

[89] **Koper GJM**, Reiss H. Length Scale for the Constant Pressure Ensemble: Application to Small Systems and Relation to Einstein Fluctuation Theory. J Phys Chem. 1996; 100(1):422–432. https://doi.org/10.1021/jp951819f.

[90] **Corti DS**, Soto-Campos G. Deriving the Isothermal–isobaric Ensemble: The Requirement of a "Shell" Molecule and Applicability to Small Systems. J Chem Phys. 1998; 108(19):7959–7966. https://doi.org/10.1063/1.476236.

[91] **Corti DS**. Monte Carlo Simulations in the Isothermal—isobaric Ensemble: the Requirement of a 'shell' Molecule and Simulations of Small Systems. Mol Phys. 2002; 100(12):1887–1904. https://doi.org/10.1080/00268970210126619.

[92] **Attard P**. On the Density of Volume States in the Isobaric Ensemble. J Chem Phys. 1995; 103(22):9884–9885. https://doi.org/10.1063/1.469956.

[93] **Han KK**, Son HS. On the Isothermal-isobaric Ensemble Partition Function. J Chem Phys. 2001; 115(16):7793–7794. https://doi.org/10.1063/1.1407295.

[94] **Corti DS**. Isothermal-isobaric Ensemble for Small Systems. Phys Rev E. 2001; 64(1):016128. https://doi.org/10.1103/PhysRevE.64.016128.

[95] **Ströker P**, Hellmann R, Meier K. Systematic Formulation of Thermodynamic Properties in the NpT Ensemble. Phys Rev E. 2021; 103(2):023305. https://doi.org/10.1103/PhysRevE.103.023305.

[96] **Panagiotopoulos AZ**. Adsorption and Capillary Condensation of Fluids in Cylindrical Pores by Monte Carlo Simulation in the Gibbs Ensemble. Mol Phys. 1987; 62(3):701–719. https://doi.org/10.1080/00268978700102501.

[97] **Panagiotopoulos AZ**, Q N, S M, , Tildesley DJ. Phase Equilibria by Simulation in the Gibbs Ensemble: Alternative Derivation, Generalization and Application to Mixture and Membrane Equilibria. Mol Phys. 1988; 63(4):527–545. https://doi.org/10.1080/00268978800100361.

[98] **Mooij GCAM**, Frenkel D, Smit B. Direct Simulation of Phase Equilibria of Chain Molecules. J Condens Matter Phys. 1992; 4(16):L255. https://doi.org/10.1088/0953-8984/4/16/001.

[99] **de Pablo JJ**, Laso M, Suter UW. Estimation of the Chemical Potential of Chain Molecules by Simulation. J Chem Phys. 1992; 96(8):6157–6162. https://doi.org/10.1063/1.462658.

[100] **Mezei M**. A Cavity-biased (T, V, μ) Monte Carlo Method for the Computer Simulation of Fluids. Mol Phys. 1980; 40(4):901–906. https://doi.org/10.1080/00268978000101971.

[101] **Mezei M**. Grand-canonical Ensemble Monte Carlo Study of Dense Liquid. Mol Phys. 1987; 61(3):565–582. https://doi.org/10.1080/00268978700101321.

[102] **Snurr RQ**, Bell AT, Theodorou DN. Prediction of Adsorption of Aromatic Hydrocarbons in Silicalite from Grand Canonical Monte Carlo Simulations with Biased Insertions. J Phys Chem. 1993; 97(51):13742–13752. https://doi.org/10.1021/j100153a051.

[103] **Ikeda T**, Nakayama A. Generalization of Biasing Particle Insertion and Deletion for Grand Canonical Monte Carlo Simulation. J Chem Theory and Comput. 2024; https://doi.org/10.1021/acs.jctc.4c00755.

[104] **Zhang H**, Snurr RQ. Computational Study of Water Adsorption in the Hydrophobic Metal-Organic Framework ZIF-8: Adsorption Mechanism and Acceleration of the Simulations. J Phys Chem C. 2017; 121(43):24000–24010. https://doi.org/10.1021/acs.jpcc.7b06405.

[105] **Sastry S**, Corti DS, Debenedetti PG, Stillinger FH. Statistical Geometry of Particle Packings. I. Algorithm for Exact Determination of Connectivity, Volume, and Surface Areas of Void Space in Monodisperse and Polydisperse Sphere Packings. Phys Rev E. 1997; 56(5):5524–5532. https://doi.org/10.1103/PhysRevE.56.5524.

[106] **Barnes BC**, Siderius DW, Gelb LD. Structure, Thermodynamics, and Solubility in Tetromino Fluids. Langmuir. 2009; 25(12):6702–6716. https://doi.org/10.1021/la900196b.

[107] **Chen B**, Siepmann JI. Improving the Efficiency of the Aggregation-Volume-Bias Monte Carlo Algorithm. J Phys Chem B. 2001; 105(45):11275–11282. https://doi.org/10.1021/jp012209k.

[108] **Rovigatti L**, Russo J, Romano F. How to Simulate Patchy Particles. Eur Phys J E. 2018; 41(5):59. https://doi.org/10.1140/epje/i2018-11667-x.

[109] **Wierzchowski S**, Kofke DA. A General-purpose Biasing Scheme for Monte Carlo Simulation of Associating Fluids. J Chem Phys. 2001; 114(20):8752–8762. https://doi.org/10.1063/1.1369131.

[110] **Wierzchowski S**, Kofke DA. UB Association Bias Algorithm Applied to the Simulation of Hydrogen Fluoride. Fluid Ph Equilib. 2002; 194-197:249–256. https://doi.org/10.1016/S0378-3812(01)00671-9.

[111] **Siderius DW**, Hatch HW, Shen VK. Flat-Histogram Monte Carlo Simulation of Water Adsorption in Metal–Organic Frameworks. J Phys Chem B. 2024; https://doi.org/10.1021/acs.jpcb.4c00753.

[112] **Hatch HW**, Yang SY, Mittal J, Shen VK. Self-assembly of Trimer Colloids: Effect of Shape and Interaction Range. Soft Matter. 2016; 12(18):4170–4179. https://doi.org/10.1039/C6SM00473C.

[113] **Dubbeldam D**, Torres-Knoop A, Walton KS. On the Inner Workings of Monte Carlo Codes. Mol Simul. 2013; 39(14-15):1253–1292. https://doi.org/10.1080/08927022.2013.819102.

[114] **Flyvbjerg H**, Petersen HG. Error Estimates on Averages of Correlated Data. J Chem Phys. 1989; 91(1):461–466. https://doi.org/doi:10.1063/1.457480.

[115] **Grossfield A**, Patrone PN, Roe DR, Schultz AJ, Siderius DW, Zuckerman DM. Best Practices for Quantification of Uncertainty and Sampling Quality in Molecular Simulations [Article V1.0]. Living J Comput Mol Sci. 2018; 1(1). https://doi.org/10.33011/livecoms.1.1.5067.

[116] **Shen VK**, Siderius DW, Krekelberg WP, Hatch HW. NIST Standard Reference Simulation Website. NIST Standard Reference Database Number 173, National Institute of Standards and Technology; 2017. http://doi.org/10.18434/T4M88Q.