Optimizing Monte Carlo simulation of the square-well fluid

by
Michael A. Perlin

A PROJECT

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Physics
(Honors Scholar)

Presented 02 June 2015
Commencement June 2015

AN ABSTRACT OF THE THESIS OF

Michael A. Perlin for the degree of Honors Baccalaureate of Science in Physics presented on 02 June 2015.
Title: Optimizing Monte Carlo simulation of the square-well fluid.

Abstract approved:

David Roundy

We identify and develop efficient Monte Carlo methods for determining thermodynamic properties of the square-well fluid in order to test square-well density functional theories near the critical point. Previous works have developed generic so-called histogram methods for collecting statistics on low energy system states, but little or no literature exists on these methods' systematic comparison, as well as their application to the square-well fluid. The square-well fluid in particular introduces application challenges not manifest in traditional models for testing and benchmarking such numerical techniques (e.g. the Ising model).

We implement our own "simple flat" method, the Wang-Landau method, transition matrix Monte Carlo (TMMC), and a modified version of the optimized ensemble. The performance of each method is measured by low energy sampling rates and maximum errors in computed system properties. We find that Wang-Landau potentially performs better than other histogram methods, but fails catastrophically without a predetermined energy range. The simple flat method and TMMC give results comparable to successful Wang-Landau simulations, but simple flat has some anomalous cases with large errors. Finally, our implementation of the optimized ensemble using a transition matrix results in worse performance over the straightforward TMMC.

Key words: Monte Carlo methods, Square-well fluid, Histogram methods

Corresponding e-mail address: mika.perlin@gmail.com

Optimizing Monte Carlo simulation of the square-well fluid

by
Michael A. Perlin

A PROJECT

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Physics
(Honors Scholar)

Presented 02 June 2015
Commencement June 2015

Honors Baccalaureate of Science in Physics project of Michael A. Perlin presented on 02 June 2015.

APPROVED:

David Roundy, Mentor, representing Physics

Henri J. F. Jansen, Committee Member, representing Physics

David H. McIntyre, Committee Member, representing Physics

Toni Doolen, Dean, University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, University Honors College. My signature below authorizes release of my project to any reader upon request.

Michael A. Perlin, Author

Table of Contents

List of Figures

List of Algorithms

1 Introduction

The critical point of a system is the point in phase space at which the distinction between liquid and gaseous phases ceases to be well-defined. As shown in Figure 1, a fluid need not boil or condense to transition between liquid to gaseous phases: it can smoothly transition between the two by circumventing the critical point. Understanding the behavior of fluids near the critical point is a major challenge in the field of classical density functional theory. The square-well fluid is the simplest system with a liquid-vapor phase transition, and is therefore of interest for studying the critical point. For a generic square-well fluid to be observed in a liquid state, it must have a localized distribution of the spheres which make up the fluid. Such a distribution implies a low energy fluid state, as all interactions between spheres are attractive. Conversely, a low energy liquid state of a square-well fluid otherwise observable in both liquid and gaseous phases implies a localized distribution of spheres. The search for liquid states of the square-well fluid is thus equivalent to the search for low energy states.

Monte Carlo simulations are a standard means for studying the equilibrium thermodynamic properties of a system. In the most straightforward implementation of Monte Carlo fluid simulations, however, highly localized distributions of spheres are extremely unlikely to occur, making such simulations impractical for studying the critical point of the square-well fluid. Several generic computational methods, called histogram methods, exist for dealing with such a problem; namely, that interesting regions of a system's state space cannot be sampled via standard Monte Carlo methods in a reasonable amount of time. There is little consensus, however, on the relative efficacy of these methods, and few have applied them to simulation of the square-well fluid.

In this paper, we first review what the square-well fluid is, as well as how to simulate it via standard, unbiased Monte Carlo (Sections 2.1–2.2). We then address the shortcomings of these simulations as a means for studying the square-well fluid, and consider how one can modify simulations to overcome these shortcomings in Section 2.3. We develop a thorough understanding of several histogram methods in Sections 3.1–3.6, and discuss some details pertinent to the actual implementation of these methods in Section 3.7. Finally, in Section 4 we analyze the performance of a few histogram methods in simulations of a particular square-well fluid.

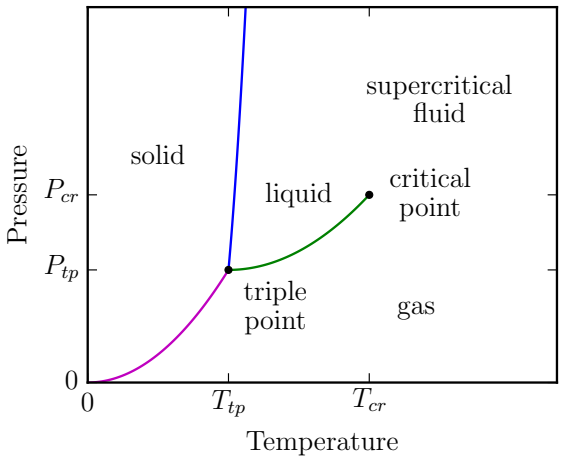

Figure 1: A generic phase diagram showing the boundaries between solid, liquid, and gaseous phases.

2 Background

2.1 The square-well fluid

The square-well (SW) fluid is a simple model used in classical density functional theories to capture low order effects of short-range attractive forces, such as the van der Waals force. The fluid is composed of spheres with diameter σ which have a pair potential

$$v_{sw}(\mathbf{r}) = \begin{cases} \infty & |\mathbf{r}| < \sigma \\ -\epsilon & \sigma < |\mathbf{r}| < \lambda\sigma \\ 0 & |\mathbf{r}| > \lambda\sigma \end{cases}, \tag{1}$$

also shown graphically in Figure 2, where the parameters λ and ϵ are referred to as the well width and depth, respectively. The first ($|\mathbf{r}| < \sigma$) part of this potential forbids spheres from overlapping, whereas the second ($\sigma < |\mathbf{r}| < \lambda\sigma$) associates an energy $-\epsilon$ with each pair of spheres whose centers are within distance of $\lambda\sigma$ of each other (where typically, $\lambda \in (1, 3]$). The net potential energy of the square-well fluid is thus

$$E = \sum_{i<j} v_{sw}(\mathbf{r}_i - \mathbf{r}_j), \tag{2}$$

where $\mathbf{r}_i$ is the position of the i-th sphere. As the potential energy E is the primary form of energy concerning us in this paper, we will refer to E as simply the "energy" of the fluid. An important feature of the square-well fluid is that its energy is always an integer multiple of the well depth. A homogeneous square-well fluid is thus uniquely identified by its well width λ and filling fraction η (i.e. the proportion of space filled by spheres; a dimensionless density), as all other properties can be normalized to the natural energy scale ϵ and length scale σ. In practice, our simulation codes use dimensionless energies E/ϵ, temperatures kT/ϵ, and distances r/σ.

2.2 Monte Carlo fluid simulation

Model systems, such as the square-well fluid, are powerful tools for understanding complex physical systems, but they do not themselves exist in the real world. Consequently, direct experimental

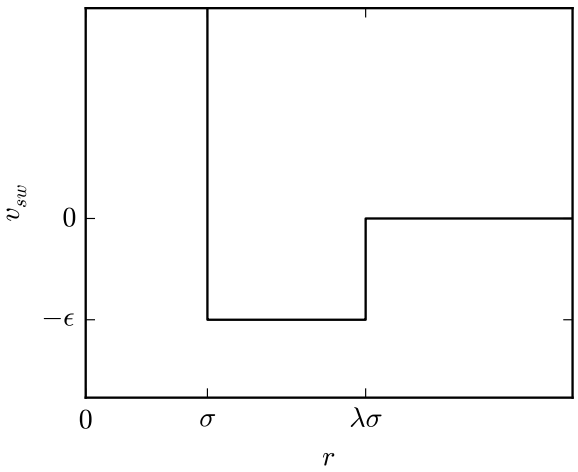

Figure 2: The square-well pair potential can be used to model short range forces to first order. The infinite potential at $r < \sigma$ simply enforces the condition that spheres cannot overlap. At $r > \lambda\sigma$, the potential is null, and makes no contribution to the fluid's internal energy.

tests of theories for model system (e.g. square-well density functional theories) are not possible. For this reason, model system theories are commonly tested against Monte Carlo simulations. Proper implementation of Monte Carlo methods to study completely characterized systems ensures that statistical results from simulations converge on the exact properties of the simulated system in the infinite simulation time limit. Furthermore, uncertainties in quantities computed via Monte Carlo simulations are typically well-defined, monotonically decreasing functions of simulation time, allowing one to run simulations to the desired level of accuracy.

2.2.1 Implementation

Algorithm 1 provides a sketch of unbiased Monte Carlo fluid simulation. Such an algorithm is "unbiased" in the sense that it collects statistics (i.e. data) on all valid system configurations with equal probability. Statistics whose collection time scales as $\mathcal{O}(1)$ with system size, meaning that increasing the number of spheres N in the simulation does not affect collection time, can be collected after every move (defined in the algorithm). Statistics whose collection time scales as $\mathcal{O}(N)$, meaning that doubling N doubles collection time, can be collected after each iteration. In general, collection with $\mathcal{O}[\chi(N)]$ time scaling should not occur more often than once every $\chi(N)$ moves, where $\chi(N)$ may be an arbitrary function of N, e.g. $N \log N$, or 2^N. These collection rules ensure that scaling up simulations does not cause them to asymptotically spend all computation time only collecting statistics, or only simulating the fluid. Collected statistics are used to find thermodynamic properties of the simulated fluid.

In this work, we are concerned with simulating the *homogeneous* square-well fluid. To avoid edge effects resulting from fluid behavior near a wall, we employ periodic boundary conditions. The use of a finite cell with periodic boundary conditions suppresses all density fluctuations on scales larger than the dimensions of the simulated fluid cell, thereby introducing a source of error. Addressing and sequestering this error, however, is outside the scope of this work, and involves considering the limit of numerical results as the number of spheres $N \to \infty$ (keeping the filling fraction η constant).

Algorithm 1: Unbiased Monte Carlo fluid simulation

1. Construct an initial "typical" (i.e. non-ordered) fluid configuration.
2. Randomly attempt to change the position of one sphere (in general, a single fluid "atom" or "molecule") to another location, rejecting the change if it results in a forbidden fluid configuration (e.g. two or more spheres overlap) and accepting the change otherwise. Attempting to move a single sphere is referred to as a move.
3. Repeat step 2 for every other sphere in the fluid. Attempting to move every sphere once is referred to as an iteration of the simulation.
4. Repeat step 3 indefinitely, or until data of sufficient quality has been generated, periodically collecting and dumping statistics on fluid states (e.g. energy, pair distribution histograms, etc.) to data files.

2.2.2 Computing observables

The sort of Monte Carlo simulations described above are only capable of collecting statistics on functions of system microstates, and finding correlations between these functions. For example, in simulation one might periodically compute both the energy $E(s)$ and some system property $X(s)$, both of which are determined by the microstate s, in order to find the mean value of X at any given energy E, i.e. $\langle X \rangle_E$. In the real world, however, information about a system's microstates, and by extension information about functions of microstates (e.g. the energy E), is inaccessible. One therefore cannot measure $\langle X \rangle_E$ directly. Instead, one typically measures the dependence of thermodynamic properties on macroscopic state variables such as temperature, i.e. $\langle X \rangle_T$.

To find $\langle X \rangle_T$, we first need to understand the concept of a density of states. Given an arbitrary system property $Y(s)$ (e.g. energy) determined by the microstate s, one can define a density of states $D(Y)$ such that i) for arbitrary Y_0, the value of $D(Y_0)$ is proportional to the number of microstates s for which $Y(s) = Y_0$, and ii) $\sum_Y D(Y) = 1$. The units of $D(Y)$ are inverse to the units of Y. The temperature dependence of $\langle X \rangle_T$ can be expressed in terms of the expectation value $\langle X \rangle_E$ and the density of states $D(E)$ by

$$\langle X \rangle_T = \frac{1}{Z(T)} \sum_E \langle X \rangle_E D(E) e^{-E/kT}, \tag{3}$$

where the partition function $Z(T)$ is simply a normalization factor, given by

$$Z(T) = \sum_E D(E) e^{-E/kT}. \tag{4}$$

To reduce redundant computations and numerical error in implementation, we will generally use a partition function with an unnormalized density of states $\tilde{D}(E)$, i.e.

$$\tilde{Z}(T) = \sum_E \tilde{D}(E) e^{-E/kT}, \tag{5}$$

in terms of which

$$\langle X \rangle_T = \frac{1}{\tilde{Z}(T)} \sum_E \langle X \rangle_E \tilde{D}(E) e^{-E/kT}. \tag{6}$$

2.3 Histogram methods

Due to the fact that unbiased Monte Carlo simulations sample all of state space randomly, a histogram $H(X)$ of observations of some system property X is directly proportional to the density of states $D(X)$; that is, the number of observations $H(X_0)$ of the system with $X(s) = X_0$ is proportional to the total number of states s for which $X(s) = X_0$. It is sometimes the case, however, that the density of states in some range R of possible X is so low that it is practically impossible (via unbiased Monte Carlo) to sufficiently sample R, that is, to accumulate a statistically significant histogram $H(X \in R)$, in any reasonable amount of time. Given that the entropy $S(X)$ can be expressed in terms of the number of microstates $\Omega(X)$ as $S(X) = k \ln \Omega(X) \propto \ln D(X)$, we may refer to regions with low state densities $D(X)$ as low entropy states.

Figure 3 provides an example of an unnormalized density of states $\tilde{D}(E)$ in energy for a particular square-well fluid. The density of states has been "normalized" to have a maximum value of 1 in order to emphasize the logarithmic span of $D(E)$. This logarithmic span ($\sim 10^{300}$) is proportional to the number of moves which one must simulate via unbiased Monte Carlo in order to observe

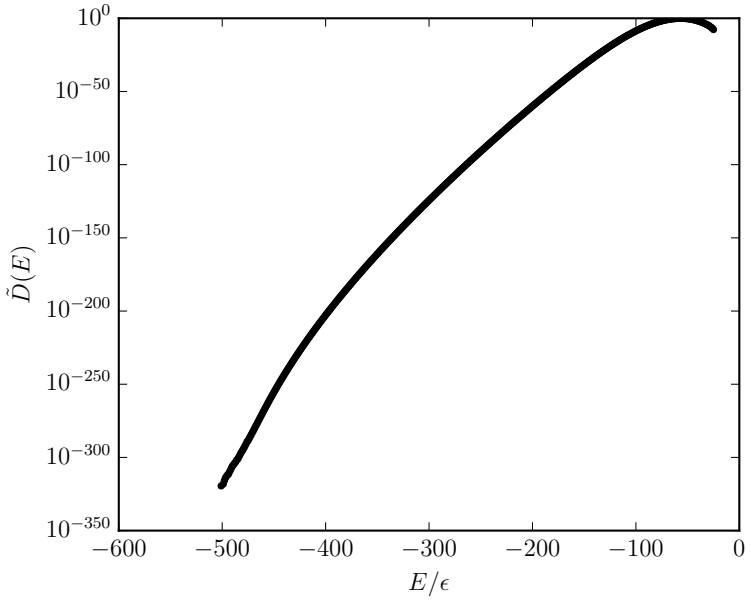

Figure 3: Sample density of states for square-well fluid with a well width $\lambda = 1.3$, filling fraction $\eta = 0.1$, and $N = 100$ spheres, computed via the transition matrix Monte Carlo method. The density of states has been normalized to have a maximum value of 1.

the energies with the lowest state densities. For reference, there are approximately 10^{80} atoms in the observable universe, and the world's current fastest supercomputers can perform about 10^{16} floating-point operations per second.

2.3.1 Biased sampling

We do not have time to count all of the atoms in the observable universe, and we certainly do not have time to wait for unbiased Monte Carlo to sample (and worse, collect decent statistics on) low energy square-well fluid states. Histogram methods provide a means to address unbiased Monte Carlo's inability to sufficiently sample low energy states by introducing a bias into the otherwise random sampling of state space. A weighting function w is introduced, whose domain is the value of some property $X(s)$ which depends on the system state s. An additional condition is then added to step 2 of Algorithm 1 in order to accept an attempted move: the weights $w(X_i)$ and $w(X_f)$ of the initial (pre-move) state s_i and final (post-move) state s_f, respectively, are used to determine the probability $P_m(s_i \to s_f)$ of accepting an otherwise valid move via

$$P_m(s_i \to s_f) = \min\left\{\frac{w(X_f)}{w(X_i)}, 1\right\}. \tag{7}$$

This formula means that when $w(X_f) > w(X_i)$, the move $s_i \to s_f$ is accepted; when $w(X_f) < w(X_i)$, the ratio of these weights determines the probability of accepting the move. Due to the fact that only ratios of weights determine $P_m(s_i \to s_f)$, the weights $w(X)$ are scale-invariant, meaning that their effect on simulations is unchanged by scale factors that are constant with respect to $X(s)$. Furthermore, simulating with flat weights $w(X) = w_0$ for all X is equivalent to simulating without weights, as the probability $P_m(s_i \to s_f)$ will always equal 1.

This sort of biased simulation is called Metropolis-Hastings Monte Carlo sampling. Employing Metropolis-Hastings Monte Carlo simulations, sketched out in Algorithm 2, allows one to construct

Algorithm 2: Metropolis-Hastings Monte Carlo fluid simulation

1. Construct an appropriate weight function $w[X(s)]$, whose argument is a system property $X(s)$ determined by the microstate s.
2. Construct an initial typical fluid configuration.
3. Randomly attempt to change the position of one sphere, rejecting the change if i) it results in a forbidden fluid configuration, or ii) a newly chosen random number on the interval $[0, 1]$ is larger than the probability determined by (7) for the initial and final states s_i and s_f, respectively.
4. Repeat step 3 the simulation produces statistics of sufficient quality.

weight functions that favor some region of state space over others, as transitions to states with higher weights are always accepted, whereas transitions to states with lower weights may be rejected, artificially preventing the simulated system from leaving interesting regions of state space. Crucially, the bias introduced by weights can be reversed when computing system properties from sampling statistics, as the weight of a particular state is directly proportional to the probability bias of that state; that is, a state with a weight of 2 will be sampled twice as often as it would have been with a weight of 1. If we wish to convert a histogram $H(X)$ of observations in a Metropolis-Hastings Monte Carlo simulation into a numerical unnormalized density of states $\tilde{D}(X)$ in X, we therefore divide the histogram by the corresponding weights $w(X)$, i.e.

$$\tilde{D}(X) = \frac{H(X)}{w(X)}. \tag{8}$$

The normalized density of states $D(X)$ in X is then

$$D(X) = \frac{\tilde{D}(X)}{\sum_{X'} \tilde{D}(X')} = \frac{H(X)/w(X)}{\sum_{X'} H(X')/w(X')}. \tag{9}$$

In general, the domain and shape of the weight function will depend on the desired yields (e.g. density of states, heat capacity) of a simulation. A histogram method is simply an algorithm or procedure for determining a weight function appropriate for a particular simulation.

In this paper, we will use a histogram $H(E)$, weights $w(E)$, and density of states $D(E)$ which are functions of the square-well fluid's energy E. Due to the fact that the square-well fluid can only have discrete energies $E = -n\epsilon$, where n is a non-negative integer and ϵ is the well depth, in simulation we store the weight function as an array of values. We will therefore generally refer to the weight function as a "weight array," or simply "weights."

2.3.2 Canonical weights

The most common weight array $w(E)$ used by physicists for what is called a canonical (or Metropolis) Monte Carlo simulation involves choosing a particular temperature T_0, and using weights proportional to the Boltzmann factor at that temperature, i.e.

$$w(E) = e^{-E/kT_0}, \tag{10}$$

where there is no reason to normalize $w(E)$ due to the fact that only ratios of weights, as per (7), are ever used in simulation. The fact that only ratios of weights are used in simulation makes $w(E)$ scale-invariant.

The partition function at $T = T_0$ for simulations with canonical weights is

$$\tilde{Z}(T_0) = \sum_E \tilde{D}(E)\, e^{-E/kT_0} = \sum_E \frac{H(E)}{w(E)}\, e^{-E/kT_0} = \sum_E \frac{H(E)}{e^{-E/kT_0}}\, e^{-E/kT_0} = \sum_E H(E), \tag{11}$$

and the value of a thermodynamic property $\langle X \rangle_{T_0}$

$$\langle X \rangle_{T_0} = \frac{1}{\tilde{Z}(T_0)} \sum_E \langle X \rangle_E \tilde{D}(E)\, e^{-E/kT_0} = \frac{\sum_E \langle X \rangle_E H(E)}{\sum_E H(E)}, \tag{12}$$

where $\langle X \rangle_E$ is the mean value of X at the energy E. The simplifications in in (11) and (12), which have no explicit dependence on T_0, occur because canonical Monte Carlo simulations sample energies in proportion to the distribution (over energy) of microstates at a temperature of T_0. Canonical Monte Carlo simulations can therefore fail to sufficiently sample energies which are important (i.e. energies with a non-negligible state probability density) at different temperatures. As a consequence, such simulations should not be used to determine properties $\langle X \rangle_{T \neq T_0}$, and are thus referred to as "fixed temperature" simulations. Incidentally, unbiased Monte Carlo simulations can be thought of as infinite-temperature simulations, as they are equivalent to simulations with canonical weights

$$w(E) = \lim_{T \to \infty} e^{-E/kT} = 1. \tag{13}$$

This observation will be used in Section 3.7.2 to identify a maximum energy of interest in simulations.

Though canonical Monte Carlo is simple to implement, its inability to investigate a system at more than one temperature at a time is a disadvantage for determining the temperature dependence of system properties. In order to find the behavior of $\langle X \rangle_T$, one must run many simulations at discrete temperature intervals; each such simulation will yield one data point on $\langle X \rangle_T$.

Sampling low energies, however, or understanding system behavior at low temperatures, is even more problematic with canonical weights. Using low temperature canonical weights will indeed force a simulated system down to low energies, but will also likely freeze the system into a local minimum of its energy landscape. Freezing into a state in this manner will limit a simulation to sampling only a small portion of the energy landscape, even though there may (and generally will) be many other states with the same energy.

Due to these problems, we will not use canonical weights alone to study the square-well fluid. We will, however, use canonical weights for portions of all weight arrays, which we discuss in Section 3.7.2.

2.3.3 Broad energy sampling

Given the expression for $\langle X \rangle_T$ in (6), sufficient accumulation of statistics on $\langle X \rangle_E$ at all available energies in principle allows one to determine $\langle X \rangle_T$ for any temperature T. In practice, the density of states can fall off so quickly with energy that some range of allowable energies is practically inaccessible via Monte Carlo simulations, biased or otherwise. In such a case, computing $\langle X \rangle_T$ to a reasonable degree of accuracy requires sufficiently sampling the energies at which $\tilde{D}(E)\, e^{-E/kT}$

dominates the sum in (6). Section 3.7.2 discusses identifying the energies which are important for computing system properties at a given temperature.

We will employ histogram methods in order to sample as broad of an energy range as possible in each simulation. Broad energy sampling will allow us to determine, for various square-well fluids, i) the density of states $D(E)$, and ii) the temperature dependence $\langle X \rangle_T$ of various thermodynamic properties X, particularly at temperatures near the liquid-gas phase boundary.

3 Methods

Having covered the basics of the square-well fluid as well as Monte Carlo fluid simulation, we can now discuss several histogram methods for constructing energy weights. In Section 3.1, we will introduce the flat histogram method, which is not directly applicable to the study of most systems, but which forms the basis of many other methods. We will then introduce, in Section 3.2, the "simple flat" method, which we designed to be the simplest viable histogram method, and against which we can compare more complicated methods. Next, in Sections 3.3–3.5 we introduce three published methods: Wang-Landau, transition matrix Monte Carlo, and the optimized ensemble. The last of these methods we will not actually implement and analyze because, unlike most histogram methods, it was designed to optimize pre-existing weights rather than construct them. We will, however, borrow ideas from the optimized ensemble to design a histogram method of our own, a hybrid of the optimized ensemble and transition matrix Monte Carlo, in Section 3.6. Finally, we will cover some general implementation details and loose ends in Section 3.7

3.1 The flat histogram (multi-canonical) method

The flat histogram method, also called the multi-canonical method, assumes complete knowledge of the density of states $D(E)$ of the system in question, and solves (8) for a weight array $w(E)$ which should yield a flat energy histogram $H(E) = H_0$ to get

$$w(E) = \frac{1}{D(E)}, \tag{14}$$

where we may neglect the constant scale factor H_0 and do not worry about normalization factor distinguishing $D(E)$ from $\tilde{D}(E)$ due to the scale-invariance of $w(E)$.

For all but the most trivial or well-studied systems, however, the density of states $D(E)$ is not known prior to simulation, and is in fact one of the system properties which a simulation is intended to determine. Nonetheless, the this method has pedagogical value by enabling discussion of an "optimal" weight array for generating a flat energy histogram in simulations. All algorithms to construct $w(E)$ with the aim of producing a flat energy histogram should converge on the same weight array, given by (14).

Algorithm 3 provides a naive way to implement the flat histogram method without knowing a priori the density of states $D(E)$. This algorithm involves first running a simulation without weights (or with constant weights $w(E) = 1$) while collecting a histogram $H(E)$ of the observed energies E. After some time (loosely defined), the histogram $H(E)$ may be taken as an approximation of the density of states $\tilde{D}(E)$, and used via (14) to generate a weight array for an actual simulation. Algorithm 3 has one free parameter, which determines the amount of time for which to collect an energy histogram $H(E)$.

The problem with this implementation of the flat histogram method is that histogram methods are generally necessary when some range of energies R is inaccessible via unbiased Monte Carlo simulations. In such a case, the energy histogram $H(E \in R)$ after simulating for any reasonable

amount of time will be statistically insignificant or null, invalidating the proportionality $H(E) \propto D(E)$. This method will therefore only "flatten" the histogram at energies E which have been sufficiently sampled for $H(E)$ to be statistically significant. The end goal in implementing histogram methods, however, is precisely to sample those energies that cannot be sufficiently sampled without employing clever schemes to bias Monte Carlo simulations. Therefore, this method does not directly help us study the square-well fluid. This method does, however, motivate the method in Section 3.2, and provides some of the theory behind the methods in Sections 3.4 and 3.6.

3.2 The simple flat method

The "simple flat" method, developed by ourselves, extends the implementation of the flat histogram method discussed in Section 3.1 to an algorithm intended to be the simplest viable broad energy histogram method. While we do not expect this method to outperform other, more sophisticated methods, the simple flat method should provide a lower bar and a standard of comparison for the performance of other methods. Any nontrivial histogram method should, at a minimum, consistently outperform the simple flat method in order to be considered for any applications.

The idea behind the simple flat method is such: after simulating for some time, the energy histogram $H(E)$ will have a peak, or some other uneven distribution of observations, in a region of energies R for which $H(E \in R)$ statistically significant. One can thus use the existing histogram to construct weights which should flatten $H(R)$. Subsequent simulations should then spend less time at energies with the highest state densities, and more time at energies with relatively low state densities. The energy histogram from such simulations should therefore be statistically significant in a larger range of energies, improving the available estimate for a density of states with which to compute flat histogram weights. One could then use the improved estimate to construct new weights, and repeat this simulation and updating process until simulations satisfy some end condition (in fact, most histogram methods will require end conditions, which is discussed in Section 3.7.1).

Algorithm 4 provides an implementation of the simple flat method. This algorithm has two free parameters: the initial number of iterations n_0 for which to simulate and the factor u. Due to the exponential growth of $n_k = u^k n_0$, this algorithm should not be sensitive to the value of n_0.

The most interesting part of Algorithm 4 is step 4c, which updates the weights as $w(E) \leftarrow w(E)/H(E)$. The reasoning behind this step is such: say that for two energies E_0 and E_b we have that $H(E_0) = \langle H \rangle_E$ and $H(E_b) = b \langle H \rangle_E$, where $\langle H \rangle_E$ is the mean histogram value over all energies. In this case, we wish to decrease the current simulation bias on E_b by a factor of b while keeping the bias on E_0 the same. As the weights $w(E)$ are proportional to the simulation bias on each respective energy E, dividing the weights $w(E)$ by the current histogram $H(E)$ achieves the desired changes to the simulation biases. The factor $\langle H \rangle_E$ in this weight update has no effect on

Algorithm 3: A naive flat histogram method

1. Construct an initial typical fluid configuration.
2. Simulate (without weights) for some time τ, which may be real time, computer time, or a number of simulation iterations, collecting a histogram $H(E)$ of energy observations after every move.
3. Set weights $w(E) = 1/H(E)$. At energies with $H(E) = 0$, set $w(E) = 1$.

Algorithm 4: The simple flat method

1. Construct an initial typical fluid configuration.
2. Set $k = 0$, choose an initial number of iterations n for which to simulate, and initialize weights $w(E) = 1$ for all E.
3. Simulate for n iterations (with weights), collecting a histogram $H(E)$ of energy observations after every move.
4. If some predetermined initialization end condition has *not* been met, then
   (a) increment $k \leftarrow k + 1$
   (b) set $n_k = un_{k-1}$ with a predetermined factor $u \geq 1$,
   (c) update weights as $w(E) \leftarrow w(E)/H(E)$ for all E at which $H(E) \neq 0$,
   (d) return to step 3.

the function of the weights, and is in practice immediately scaled out of the weight array.

In our implementation of the simple flat method, we used $u = 2$ and $n_0 = L\exp(\epsilon/kT_{\min})$, where L is the number of energy levels of the given square-well fluid and $T_{\min}$ is a minimum temperature of interest for the current simulation. The factor of L appears because initialization time should scale with the range of energies of the simulated system. If we have twice the number of energies, we should explore energy space for twice as long. To estimate L, we multiplied the maximum number of spheres M which fit within a radius of $\lambda\sigma$ of a single sphere on a face-centered cubic lattice (i.e. the maximum number of spheres with which a given sphere could interact) by the total number of spheres N, and divided the result by two so as to not double-count interactions between spheres (so $L = MN/2$). The exponential $\exp(\epsilon/kT_{\min})$, a Boltzmann factor for two adjacent energies, appears because lower minimum temperatures of interest warrant sampling lower energies, which in turn requires simulating for longer (in step 3 of Algorithm 4) because these energies have lower state densities. The introduction of a minimum temperature may seem ad-hoc, but all simulations in this paper require a choice of minimum temperature anyways (discussed in Section 3.7.2).

3.3 The Wang-Landau method

The Wang-Landau method is the first published histogram method we discuss[1–4]. This method has been used to study a wide variety of systems, and is the standard method to compute the density of states of a system via Monte Carlo simulations[4–7].

Unlike the simple flat method, the Wang-Landau method modifies the weight array on the fly. After every move during initialization Wang-Landau decreases the weight $w(E)$ on the current energy E by some factor of f, so as to decrease weights on the most commonly observed energies relative to those of the least commonly observed energies. Eventually, such a simulation with constant modification of the weights $w(E)$ should yield a flat histogram $H(E) \approx \langle H \rangle_E$ for all E, as the modifications to $w(E)$ increasingly pushes the simulated system away from the most sampled energies, and toward the least sampled energies. When the energy histogram is sufficiently flat, $H(E)$ is reset (i.e. set to $H(E) = 0$ for all E), the factor f is decreased (geometrically approaching unity), and the entire process is repeated until $f - 1$ falls below some cutoff $c \ll 1$.

Algorithm 5: Wang-Landau initialization of weights

1. Construct an initial typical fluid configuration
2. Set $k = 0$, choose some factor $f_0 > 1$, and initialize $w(E) = 1$ for all E.
3. Simulate for n iterations. After each move, increment the histogram $H(E)$ of energy observations and update $w(E) \leftarrow w(E)/f_k$, where E is the current energy of the system.
4. If the histogram $H(E)$ is not sufficiently flat, i.e. if it fails a "flatness" condition $C[H]$, return to step 3, otherwise
   (a) reset $H(E) = 0$ for all E,
   (b) increment $k \leftarrow k + 1$,
   (c) set $f_k = \sqrt[u]{f_{k-1}}$ for some predetermined $u > 1$,
   (d) if $f_k \geq c$ for some predetermined c, return to step 3.

Resetting and repeating this initialization process is necessary to perform fine tuning and refinement of the weights. A flat histogram after a simulation in which $w(E)$ has changed does not guarantee that simulating with the final weights will likewise result in a flat histogram. Eventually, when $f - 1 < c \ll 1$, i.e. $f \approx 1$, the initialization process no longer makes any appreciable modifications to the weights $w(E)$, so that upon achieving a flat histogram one can be confident that a regular simulation which fixes the current weights will yield similar results.

Algorithm 5 walks through the Wang-Landau initialization procedure. This algorithm includes four free parameters which affect initialization process: the number of iterations n to run in step 3, the initial factor f_0, the update factor u, and the flatness condition $C[H]$. The cutoff c determines the condition for Algorithm 5 to terminate.

The first of these parameters, n, controls how often to check the flatness condition $C[H]$ and run updates, if necessary. The value of n is not too important, so long as it is reasonable for the system at hand. Too small value of n will waste computation time determining $C[H]$, while too large of a value will waste time simulating when $C[H]$ has already been satisfied. In general, n should be proportional to the computation time of $C[H]$, which scales with energy range of the simulated system. We therefore use $n = L$, where L is, as described in Section 3.2, the number of energy levels for the given square-well fluid. Other works typically neglect providing any heuristic or formula for n, reporting a system-independent $n = 10^3$[2] or 10^5[1].

The next parameter, f_0, controls the initial amount by which to adjust the weight of the current energy after each move. As mentioned in Algorithm 5, one should always have $f_0 > 1$, but the appropriate value of f_0 will generally depend on the system at hand. Too large a value of f_0 will magnify stochastic error in initialization, while too small a value of f_0 will cause simulations to take an exceedingly long time to converge. Lacking any heuristics for assigning f_0, we use $f_0 = e$ (more precisely, $\ln f_0 = 1$), a value commonly suggested by literature[1].

The factor u appearing in step 4c of Algorithm 5 controls the amount by which to adjust f when the histogram $H(E)$ is sufficiently flat. In principle, one need not take a root of f at each reset: the general idea behind Algorithm 5 requires only $f_k < f_{k-1}$ to converge (assuming that $C[H]$ can be satisfied) and $\lim_{k\to\infty} f_k = 1$ to terminate. Taking the u-th root of f at each increment of k is a natural and convenient iterative means to satisfy these requirements. Previous works have reported

using values such as $u = 2$ [1, 2] and $u = 10$ [3]; we use the former value.

One might guess why the flatness condition $C[H]$ for updating f and restarting the Wang-Landau initialization process should not be made too lax: doing so would prompt the initialization process to i) reset and decrease f too early, i.e. when the weight array could still use some heavy handed adjustment, and ii) quit before one has any reason to think that simulations might yield a flat histogram. Less obvious, however, is the fact that the flatness condition $C[H]$ should not be made too stringent: if the histogram is completely flat, say $H(E) = H_0$ for all E, then the weights $w(E)$ have all been modified by the same factor of f^{H_0}. Remembering the scale-invariance of $w(E)$, this result would mean that the weights have not actually changed at all!

The success of Wang-Landau thus relies on some degree of leniency in the flatness condition, which makes tuning $C[H]$ crucial to the efficient and effective implementation of this method. Some previous works[1] suggest that $C[H]$ should be made as stringent as possible, cautioning only that some simulations might not satisfy too strict of a flatness condition in any reasonable amount of time. These works fail to recognize the reliance of Wang-Landau on the *failure* of $C[H]$ to enforce a perfectly flat histogram. A commonly suggested flatness condition is that the minimum histogram value in some specified range of energies be at least some fixed proportion of the mean histogram value, i.e. $C[H] : \min[H(E)] \geq x \langle H \rangle_E$, and several papers report using $x = 0.95$. We use this condition with the same value of x. Determining the range of energies over which to evaluate $C[H]$ is the subject of Section 3.7.2.

The final free parameter in the Wang-Landau method is the cutoff c, which determines how small f can get before the algorithm stops modifying the weight array. Too large of a value for c will make Wang-Landau quit prematurely, whereas too small a value will cause the algorithm to waste time simulating after the weights $w(E)$ have essentially converged. One might imagine using an alternate end condition for Wang-Landau, which involves comparing the current weights $w_k(E)$ to those at the end of the previous cycle, $w_{k-1}(E)$, to check whether the initialization process is still making appreciable modifications to the weights, but such a check would require both a metric and a free parameter to define what an appreciable change is. For simplicity, we stuck with the standard end condition given in Algorithm 5. As with the other free parameters in Wang-Landau, we are not aware of any heuristics for determining an appropriate value of c, but literature cites typical values of $\ln c = 10^{-8}$. In order to compare different histogram methods fairly, we used an alternate end condition in step 4d of Algorithm 5 (discussed in Section 3.7.1), and therefore did not need to choose a value of c.

3.4 The transition matrix Monte Carlo (TMMC) method

Unlike the previous histogram methods, the transition matrix Monte Carlo method does not use weights, and merely computes the density of states $D(E)$ of a system[8]. The density of states can in turn be used to determine flat histogram weights via (14). TMMC introduces a new object: the energy transition matrix, T, whose components T_{ij} are the probabilities that a system will transition *to* a state with energy E_i *from* a given state with energy E_j, i.e. $T_{ij} = P(E_j \to E_i)$.

The transition matrix is trivially square and positive-definite, but it is not symmetric. Given that unbiased Monte Carlo samples all of state-space randomly, meaning all possible system states are equally likely to occur during simulation, a transition $E_i \to E_f$ from an energy with a low state density to one with a high state density, $D(E_i) < D(E_j)$, is more probable than the inverse transition $E_j \to E_i$ simply due to the fact that there are more states with energy E_i than those with energy E_j. Thus $D(E_i) < D(E_j)$ implies $P(E_i \to E_j) > P(E_j \to E_i)$, which means $T_{ji} > T_{ij}$. The asymmetry of the transition matrix is precisely the origin of the second law of thermodynamics: systems are exceedingly likely to evolve towards macroscopic states with higher accessible microstate

densities.

Knowing the transition matrix of a system determines, among other properties, the system's density of states. Consider an ensemble of unbiased Monte Carlo simulations with a distribution of energies at a time t given by $\hat{D}^t(E)$, such that the probability of any given simulation to have an energy E_i at a time t is $\hat{D}_i^t = \hat{D}^t(E_i)$. The probability $\hat{D}_i^t$ can be expressed in terms of the distribution $\hat{D}^{t-1}(E)$ one time step (i.e. one Monte Carlo move) prior by

$$\hat{D}_i^t = \sum_j P(E_i \to E_j)\, \hat{D}_j^{t-1} = \sum_j T_{ij} \hat{D}_j^{t-1}. \tag{15}$$

At equilibrium, the probability distributions do not vary with time, meaning $\hat{D}^t = \hat{D}^{t-1}$. Furthermore, the distribution of states in an equilibrium ensemble of unbiased Monte Carlo simulations is precisely the density of states $D(E)$, which means $D_i = \sum_j T_{ij} D_j$. Expressing this condition for all energies with a state density vector D and transition matrix T,

$$D = TD. \tag{16}$$

Finding the density of states of a system can thus be reduced to finding the eigenvector of the transition matrix which has a corresponding unit eigenvalue. Once one has determined the density of states of a system, one can initialize weights via the flat histogram method, i.e. (14).

Computationally determining the transition matrix of a system is straightforward: one need only simulate the system and collect a histogram $\tilde{T}_{ij} = \tilde{T}(E_i, E_j)$ of the energy transitions after each unbiased Monte Carlo move attempt, i.e. after step 2 of Algorithm 1. A histogram of transitions $\tilde{T}\left(E_i, \bar{E}_j\right)$ from a fixed energy $\bar{E}_j$ to energies E_i is then proportional to the probability distribution of transitions from $\bar{E}_j$, i.e. the transition matrix $T\left(E_i, \bar{E}_j\right)$. Crucially, this proportionality does not depend on the history of a simulation, or on how the simulation got to be in the energy $\bar{E}_j$, so long as samples of the energy $\bar{E}_j$ are themselves void of systemic bias. Furthermore, one does not actually need to transition from E_j to E_i in order to collect statistics on $T(E_i, E_j)$: after deciding whether to accept a move based on whether it results in a valid system state, one can increment $\tilde{T}(E_i, E_j)$ before deciding to reject the move for some other reason, e.g. because $w(E_j) \gg w(E_i)$, or even because one wishes to collect more statistics on $T\left(E_i, \bar{E}_j\right)$.

The proportionality $\tilde{T}_{ij} \propto T_{ij}$ can be made explicit by enforcing that the probabilities of all transitions from an energy E_j sum to unity, i.e.

$$\sum_i P(E_j \to E_i) = \sum_i T_{ij} = 1, \tag{17}$$

which implies

$$T_{ij} = \frac{\tilde{T}_{ij}}{\sum_i \tilde{T}_{ij}}. \tag{18}$$

Computing T from $\tilde{T}$ therefore requires normalizing each row of $\tilde{T}$ independently.

If a system has L energy levels, the transition histogram $\tilde{T}(E_i, E_j)$ is an $L \times L$ matrix. Given that L is proportional to the system size N, the memory footprint of $\tilde{T}(E_i, E_j)$ grows as N^2. For most systems, however, there will be a maximum energy M independent of system size by which a single move can change the energy of the system. The transition histogram for such systems is therefore band-diagonal, which one can exploit to save memory. In particular, it is natural to store the transition histogram in the form $\tilde{T}_d(E, \Delta E)$, which gives the probability that a system will

Algorithm 6: Transition matrix Monte Carlo initialization

1. Construct an initial typical fluid configuration.
2. Initialize a transition histogram $\tilde{T}_d(E, \Delta E) = 0$ for all E and $\Delta E \in [-M, M]$, where M is the possible maximum energy transition of a single move.
3. Randomly attempt to change the position of one sphere, tentatively accepting the transition from state s_i to state s_f if s_f is not a forbidden fluid configuration.
4. Let $\Delta E = E_f - E_i$ and increment $\tilde{T}_d(E_i, \Delta E)$, where E_k is the energy of state s_k.
5. If $\Delta E < 0$, accept the move. Otherwise
   (a) compute the normalization factor
   $$n_k = \sum_{\Delta E'} \tilde{T}_d\left(E_k, \Delta E'\right)$$
   for $k \in \{i, f\}$, and
   (b) accept the move $s_i \to s_f$ with probability
   $$P_m(s_i \to s_f) = \max\left\{ \frac{\left(\tilde{T}_d(E_i, \Delta E) + c\right)/(n_i + c)}{\left(\tilde{T}_d(E_f, -\Delta E) + c\right)/(n_f + c)}, \exp\left(-\frac{\Delta E}{kT_{\min}}\right) \right\}$$
   for a predetermined number c and temperature $T_{\min}$.
6. If some predetermined initialization end condition has *not* been met, return to step 3. Otherwise, compute the density of states $D(E)$ by solving (16), and set weights $w(E) = 1/D(E)$.

transition from the energy E to the energy $E + \Delta E$ in a single move. One can convert between $\tilde{T}_d(E, \Delta E)$ and $\tilde{T}(E_i, E_j)$ using the relation

$$\tilde{T}(E_i, E_j) = \tilde{T}_d(E_j, E_i - E_i), \tag{19}$$

which means, from (18),

$$T_{ij} = \frac{\tilde{T}_d(E_j, E_i - E_j)}{\sum_i \tilde{T}_d(E_j, E_i - E_j)} = \frac{\tilde{T}_d(E_j, E_i - E_j)}{\sum_{\Delta E} \tilde{T}_d(E_j, \Delta E)}. \tag{20}$$

Expressed in this form, the memory footprint of the transition histogram grows linearly with system size N and the maximal energy difference M.

Figure 4 provides an example of a transition matrix $T_d(E, \Delta E)$ for a particular square-well fluid. The band at $\Delta E = 0$ indicates that most moves transition from an energy to itself. Low energies are more likely to transition up in energy, with $\Delta E > 0$, than down in energy, with $\Delta E < 0$. The high energies which are more likely to transition down in energy than up are those above the state of maximum entropy, i.e. the state at which $D(E)$ is maximal.

While one could imagine many initialization routines to determine the transition matrix T, we use Algorithm 6, which is a modified version of a routine provided in [8]. The primary motivation

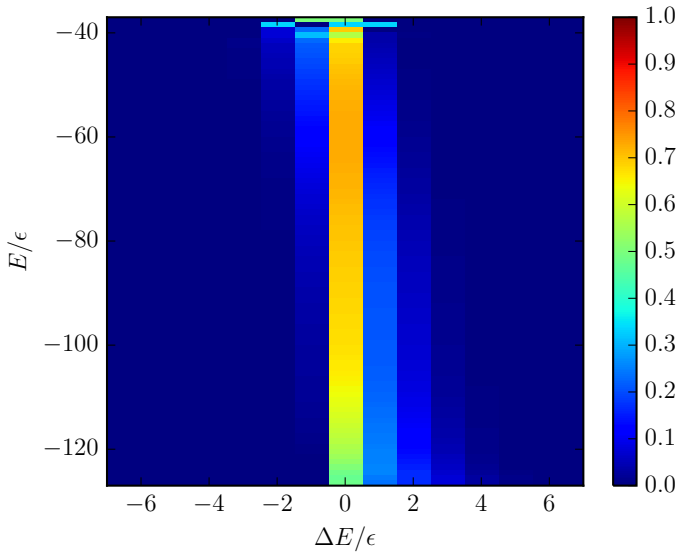

Figure 4: Sample transition matrix $T_d(E, \Delta E)$ for square-well fluid with a well width $\lambda = 1.3$, filling fraction $\eta = 0.3$, and $N = 25$ spheres. The band at $\Delta E = 0$ indicates that any given state is most likely to transition into another state with the same energy.

behind this algorithm is such: as lower energies are always more difficult to sample than higher energies, transitions to lower energies should always be accepted, while transitions to higher energies should pass a probabilistic acceptance test. To understand the probability in step 5b of Algorithm 6 of accepting a move $s_i \to s_f$ for which $E_f > E_i$, first consider the case of $c = 0$ and $T_{\min} \to 0$, in which case the probability becomes

$$P_m(s_i \to s_f) = \frac{\left(\tilde{T}_d(E_i, \Delta E)\right) / \left(\sum_{\Delta E'} \tilde{T}_d(E_i, \Delta E')\right)}{\left(\tilde{T}_d(E_f, -\Delta E)\right) / \left(\sum_{\Delta E'} \tilde{T}_d(E_f, \Delta E')\right)} = \frac{T_d(E_i, \Delta E)}{T_d(E_f, -\Delta E)} = \frac{P(E_i \to E_f)}{P(E_f \to E_i)}, \quad (21)$$

where $\Delta E = E_f - E_i$ and $P(E_m \to E_n)$ is the unbiased probability of a transition from an energy E_m to an energy E_n. The result in (21) is the probability of accepting moves $s_i \to s_f$ required for simulation via Algorithm 6 to yield a flat energy histogram[8]. Unlike the previous histogram methods, we desire a flat histogram during initialization via Algorithm 6 in order to broadly sample the transition histogram $\tilde{T}_d(E, \Delta E)$, rather than to construct weights $w(E)$ directly.

Aside from a minimum temperature $T_{\min}$ and an end condition, Algorithm 6 contains one free parameter, c, used in step 5b to modify (21). This parameter, which is our own addition to the algorithm provided in [8], makes the rejection of transitions more conservative when there are too few statistics in the histogram $\tilde{T}_d$ to be confident in the current estimate of the transition matrix. Without this parameter, the TMMC initialization routine can get stuck at energies which, due to poor collection of statistics in $\tilde{T}_d$, appear to be more important than they actually are. In order to make the effects of c negligible as $\tilde{T}_d$ collects more statistics throughout initialization, c should be of order unity, though its optimal value is generally system-dependent. In our implementation of Algorithm 6, we use $c = 16$.

The last interesting part of the probability in step 5b of Algorithm 6 is the limit on $P_m(s_i \to s_f)$ to a minimum value of $\exp(-\Delta E / kT_{\min})$. This limit caps the bias on energies to that introduced by canonical (fixed-temperature) weights at a temperature of $T_{\min}$. As a result, Algorithm 6 will

not waste time oversampling energies which are unimportant for determining system properties at temperatures $T \geq T_{\min}$.

3.5 The optimized ensemble (OE) method

All histogram methods introduced in this paper thus far have focused on determining a weight array $w(E)$ for a flat energy histogram $H(E)$, in order to sample and collect statistics on all energies equally. This motivation considers the quantity of statistics, but not their quality. Metropolis-Hastings Monte Carlo simulations collect statistics on low energy states by first getting into these states, and then rejecting moves which move the system into higher energy states. Statistics on low energy states will therefore generally be highly correlated, as they are based on many samples of only a few low energy regions of a system's energy landscape. The optimized ensemble[9] attempts to address the autocorrelation between low energy samples by finding weights to maximize the rate at which a simulation makes round trips between low and high energy states. Maximizing the round trip rate, in turn, maximizes the rate at which a simulation makes independent, uncorrelated samples of low energies.

The optimized ensemble works by considering an ensemble of simulations, each of which defines the position of a "walker" in energy space. Given that we want walkers to move back and forth between two extrema E_+ and E_- of some specified energy range, we label each walker by which extrema it has visited most recently. We can then identify "down-going" ("up-going") walkers by those which more recently visited E_+ (E_-), which means that to make a round trip they first needed to get to E_- (E_+). Denoting the total walker and down-going density at an energy E respectively by $n(E)$ and $n_+(E)$, the ratio $f(E) = n_+(E)/n(E)$ gives the proportion of walkers at E which are down-going. We also denote the walker diffusivity at an energy E by $\alpha(E)$.

At equilibrium with a flat histogram, meaning n is constant, the down-going walker current $j(E)$ can be expressed in terms of the diffusivity $\alpha(E)$ by

$$j = -\alpha \frac{dn_+}{dE}. \tag{22}$$

This identity can be taken as the definition of the walker diffusivity α, which is the independent of the label we have assigned any given walker. Substituting $n_+ = fn$,

$$j = -\alpha \left(n\frac{df}{dE} + f\frac{dn}{dE} \right) = -\alpha n \frac{df}{dE}, \tag{23}$$

where a constant n implies $dn/dE = 0$. We can rearrange (23) and integrate over the energy range as

$$\int_{f=f(E_-)}^{f(E_+)} \frac{df}{j} = -\int_{E=E_-}^{E_+} \frac{dE}{\alpha n}. \tag{24}$$

At equilibrium, the number of down-going walkers $n_+(E)$ at any given energy E is constant with time, which means that the walker flux $j(E)$ must be constant with energy. We can therefore evaluate one of the integrals in (24) to get

$$\frac{1}{|j|} = \int_{E=E_-}^{E_+} \frac{dE}{\alpha n}. \tag{25}$$

The optimized ensemble method minimizes the integral in (25), thereby maximizing the down-going walker current $|j|$, by varying the walker density n with the constraint that n must remain

normalized. With the additional assumption that $\alpha(E)$ does not strongly depend on the weights $w(E)$, the authors of [9] find that the optimal walker density n is

$$n_{\text{opt}} \propto \frac{1}{\sqrt{\alpha}}, \tag{26}$$

where the proportionality is determined by enforcing that n is normalized.

Equating the set of observations in a single simulation with a single observation of an ensemble of simulations, the walker density $n(E)$ is proportional to the energy histogram $H(E) = \tilde{D}(E) w(E)$, which means

$$w_{\text{opt}} = \frac{1}{D\sqrt{\alpha}}, \tag{27}$$

where we neglect the distinction between $\tilde{D}$ and D, which has no effect on the weights w.

The algorithm provided in [9] to construct weights given by (27) finds the diffusivity α by solving for it in (22). This procedure accepts flat histogram weights, or equivalently the density of states, as an input, and is not meant to construct weights from scratch. We therefore will not be comparing the optimized ensemble to other methods in this paper. We do, however, credit [9] for motivating the hybrid OETMMC method in Section 3.6.

3.6 The hybrid OETMMC method

As well as the density of states, the transition matrix $T(E_i, E_j)$ determines the local diffusivity $\alpha(E)$ for any given weights $w(E)$. Denoting by $P_b(E_i \to E_j)$ the actual, biased probability of transitioning from E_i to E_j in a single Monte Carlo move, we can say that

$$P_b(E_i \to E_f) = P(E_i \to E_f) P_m(E_i \to E_f) = T_{ji} \max\left\{\frac{w(E_f)}{w(E_i)}, 1\right\}, \tag{28}$$

where $P(E_i \to E_j)$ is the probability of attempting the move $E_i \to E_j$ and $P_m(E_i \to E_j)$ is the probability of accepting that move, should it occur. In terms of a transition histogram $\tilde{T}_d(E, \Delta E)$ with $E = E_i$ and $\Delta E = E_f - E_i$,

$$P_b(E \to E + \Delta E) = \frac{\tilde{T}_d(E, \Delta E) w(E + \Delta E)/w(E)}{\sum_{\Delta E'} \tilde{T}_d(E, \Delta E') w(E + \Delta E')/w(E)}. \tag{29}$$

We now define

$$\langle \Delta E \rangle_E = \sum_{\Delta E} \Delta E P_b(E \to E + \Delta E) = \frac{\sum_{\Delta E} \Delta E\ \tilde{T}_d(E, \Delta E) \max\{w(E + \Delta E)/w(E), 1\}}{\sum_{\Delta E'} \tilde{T}_d(E, \Delta E') \max\{w(E + \Delta E')/w(E), 1\}} \tag{30}$$

and

$$\left\langle \Delta E^2 \right\rangle_E = \sum_{\Delta E} \Delta E^2 P_b(E \to E + \Delta E) = \frac{\sum_{\Delta E} \Delta E\ \tilde{T}_d(E, \Delta E) \max\{w(E + \Delta E)/w(E), 1\}}{\sum_{\Delta E'} \tilde{T}_d(E, \Delta E') \max\{w(E + \Delta E')/w(E), 1\}}, \tag{31}$$

in terms of which the diffusivity $\alpha(E)$ is

$$\alpha(E) = \left\langle \Delta E^2 \right\rangle_E - \langle \Delta E \rangle_E^2. \tag{32}$$

The diffusivity $\alpha(E)$ is thus the variance of the change in energy ΔE from as single move out of E. Reconciling this definition with that we provide in (22) is considered outside the scope of this paper.

The OETMMC method, a hybrid of TMMC and the optimized ensemble, first runs Algorithm 6 to collect a transition histogram and compute flat histogram weights, then computes the diffusivity $\alpha(E)$ via (30)-(32), and finally computes optimized weights as given by (27). Unlike the optimized ensemble, OETMMC does not require any extra simulation routines, and is therefore straightforward to implement and compare to other histogram methods.

3.7 Loose ends

Throughout the discussion and development of histogram methods, we have thus far neglected expanding on a few important subjects: i) initialization end conditions i.e. the means by which we decide when an algorithm to construct weights is finished, ii) the minimum temperature $T_{\min}$ which we have sometimes mentioned, iii) the energy range over which we are interested in, and iv) what to do with the weight array outside of the energy range of interest, or at energies where it is not defined. The last three of these subjects turn out to be related, as the minimum temperature $T_{\min}$ chosen for any given simulation determines the minimum energy E_- of interest, as well as the weights $w(E)$ at energies $E < E_-$.

3.7.1 End conditions

While the Wang-Landau method comes equipped with an end condition of its own, most histogram methods require some external means of deciding when they are finished initializing. Wang-Landau's "natural" end condition does not advantage it over other methods, however, as the Wang-Landau end condition makes use of a free parameter (the cutoff c), which any end condition will require. In order to compare all methods fairly, we need a method-independent end condition to initialization. The most straightforward of such end conditions is to simply to stop initializing after some fixed number of Monte Carlo iterations, and finish up any other calculations in the algorithm if appropriate. We thus initialize systems of N spheres for $10^5 N^2$ iterations; the choice of $10^5 N^2$ we address in Section 4. In general, however, one might want an end condition which is independent of system size. We came up with a four such end conditions as examples, though we do not use them in simulations analyzed in this paper.

Our end conditions require knowledge of the minimum and maximum energies of interest, which we denote respectively by E_- and E_+. We define the pessimistic sample count $s_p(E_0)$ as the number of times which a system has traveled from E_+ to E_0. The trip from E_+ to E_0 may be made in any number of moves, but cannot include intermediate visits to E_- or E_0. The count $s_p(E_0)$ provides the most pessimistic estimate for the number of independent, uncorrelated observations of the energy E_0, assuming that a visit to E_+ wipes all correlations between samples of any other energy; this assumption will be justified by our definition of E_+ as the energy at which the density of states $D(E)$, and by extension the entropy $S(E)$, is maximal. We similarly define an optimistic sample count $s_o(E_0)$ as the number of times which the system has transitioned into E_0 from any energy $E > E_0$ in a single move. This count represents an optimistic estimate of the number of independent observations of E_0, as the system had merely to go to any energy $E > E_0$, where the density of states may be only marginally larger, between each sample.

In general, any initialization routine should sample the entire range $R = [E_-, E_+]$ of interest for a given simulation before terminating. Furthermore, samples at the end of initialization should be numerous enough to be statistically significant. Given that any samples of E_- require first

sampling all other energies in R on the way down from E_+, this energy (E_-) will generally have the fewest number of samples. As an end condition, therefore, one can simply enforce some minimum number of optimistic or pessimistic samples at E_-, i.e. minimum values of $s_o(E_-)$ or $s_p(E_-)$. A more sophisticated end condition would be to choose a minimum temperature of interest $T_{\min}$ and initialize until the mean fractional error $\langle \delta_s \rangle_{E,T_{\min}}$ in one of the sample counts reaches some minimum value. These mean errors are

$$\langle \delta_s \rangle_{E,T} = \left\langle s^{-1/2} \right\rangle_{E,T} = \sum_E s(E)^{-1/2} P(E,T) = \frac{\sum_E s(E)^{-1/2} \tilde{D}(E) e^{-E/kT}}{\sum_{E'} \tilde{D}(E') e^{-E'/kT}}, \tag{33}$$

where we use the fact that the fractional error δ_G in any histogram G collected via random Monte Carlo sampling is $G^{-1/2}$.

3.7.2 The energy range of interest

Using histogram methods generally requires specifying, whether explicitly or implicitly, the energy range of interest for a given simulation. The Wang-Landau method, for example, periodically checks a flatness condition $C[H]$ on the energy histogram $H(E)$. In order for such a condition to be well-defined, it needs have a specified energy range over which to evaluate $C[H]$. While some systems, such as the Ising model, have a well-defined and easily computable energy range, the same is not true of most systems. In the case of the square-well fluid, not only is the minimum energy unknown, but some energies have such an incredibly low state density $D(E)$ that one cannot reasonably expect to ever observe them via Monte Carlo, biased or otherwise. We therefore need to identify an energy range of interest, bounded by a minimum E_- and maximum E_+.

Identifying a maximum energy E_+ of interest is straightforward: this is simply the energy at which the density of states $D(E)$ is maximal. Though high energies become more important for determining system properties $\langle X \rangle_T$ at higher temperatures, the relative importance of energies $E > E_+$ to that of the energy E_+ approaches a maximum as $T \to \infty$. There is therefore a maximum weight $w(E)$ which is appropriate to assign energies $E > E_+$ relative to the weight $w(E_+)$, which is given by the ratio of these weights as $T \to \infty$. As observed in (13), the weight array becomes flat as $T \to \infty$, which means that the ratio of $w(E > E_+)$ to $w(E)$ appropriate for any given temperature T maximally approaches unity. We can therefore always set $w(E > E_+) = w(E_+)$, which will ensure that energies $E > E_+$ are not sampled any more than is necessary for any positive temperature. The maximum of $D(E)$ can be found by simulating without weights for a short while, after which the energy histogram $H(E)$ should be maximal at E_+.

The minimum important energy E_- for a system is less obvious than E_+. Just as determining E_+ involved reference to a "maximal" temperature $T \to \infty$, determining E_- requires a choice of some minimum temperature $T_{\min}$. The minimum important energy E_- is then the energy at which the term $D(E) e^{-E/kT_{\min}}$ in the partition function is maximal. Lower energy terms in the partition function grow increasingly less important as the distance $|E - E_-|$ grows, and are sufficiently sampled with canonical weights $w(E < E_-) = w(E_-) e^{-E/kT_{\min}}$ to determine any property $\langle X \rangle_{T \geq T_{\min}}$. To maximize $D(E) e^{-E/kT_{\min}}$, we need

$$\frac{d}{dE}\left(D e^{-E/kT_{\min}} \right) = \frac{dD}{dE} e^{-E/kT_{\min}} + D e^{-E/kT_{\min}} \left(-\frac{1}{kT_{\min}} \right) = 0, \tag{34}$$

which means

$$\frac{1}{D}\frac{dD}{dE} = \frac{d \ln D}{dE} = \frac{1}{kT_{\min}}. \tag{35}$$

Though we do not always know the density of states $D(E)$, in practice we compute the best available estimate of $D(E)$ every time we need to know E_-. If we do not find any energy satisfying (35), we use the minimum energy the simulation has ever observed in place of E_- to define our energy range of interest.

Once we identify an energy E_- satisfying (35) in an initialization routine for any histogram method, we set weights $w(E < E_-) = e^{-(E-E_-)/kT_{\min}}$ with a temperature $kT_{\min} = 0.2\epsilon$. These weights prevent the initialization routine from wasting time on energies at which we will override the weight array after initialization anyways. We chose this temperature because it is comfortably below the temperature at which the square-well fluid condenses from a gas into a liquid in simulation (see Section 4.3).

We made a special case for Wang-Landau treatment of the energy range of interest, as the design of this method assumes knowledge of the energy range over which to evaluate $C[H]$. In order to more fairly compare Wang-Landau to other methods, as well as having Wang-Landau determine E_- in the manner described above, we also simulated via Wang-Landau with E_- hard-coded for each system size N. We will distinguish the two sets of Wang-Landau simulations by referring to those in which E_- is hard-coded as "vanilla Wang-Landau."

4 Results and Discussion

Having developed and implemented all of the histogram methods discussed in Section 3, we will now look at the results from simulating with these methods. Due to time constraints, we simulated only one system, the square-well fluid with well width $\lambda = 1.3$ and filling fraction $\eta = 0.3$, for sizes (number of spheres) $N \in [5, 25]$. We initialized histogram methods for $10^5 N^2$ iterations ($10^5 N^3$ moves), and simulated for just as long afterwards. To check for consistency in simulation results, we simulated each combination of histogram method and system size with 30 different random number generator seeds.

4.1 Controls

Our analysis will involve comparing errors in system properties computed from simulation results. As we do not know the true values of these system properties, we will compare standard simulation results against those from "gold standard" simulations, which run for significantly longer than the standard $10^5 N^2$ iterations. For our gold standard simulations, we use the TMMC method and initialize until it collects 10^4 optimistic samples of the minimum important energy (i.e. until $s_o(E_-) \geq 10^4$). To increase confidence in low-temperature gold standard results, we initialized them with $kT_{\min} = 0.1\epsilon$. After initialization, the gold standard runs simulate for 10^{10} iterations ($10^{10} N$ moves). While the entire initialization and simulation process takes around two hours for the largest regular simulations, the largest golden simulations initialize for ~ 11 hours, and simulate for a week. To control for iteration number in the results from any given set of simulations, we compare against "converged flat" simulations. Converged flat simulations read in the transition matrix from golden simulations to construct a weight array, and simulate for the same number of iterations as the standard simulations.

4.2 Histograms and arrays

Most of our analysis will use quantities computed from two arrays: the energy histogram $H(E)$ and the weights $w(E)$. To get a sense of what these arrays look like, Figure 5 shows the energy histogram and weights from $N = 25$ simulations, along with the density of states computed via

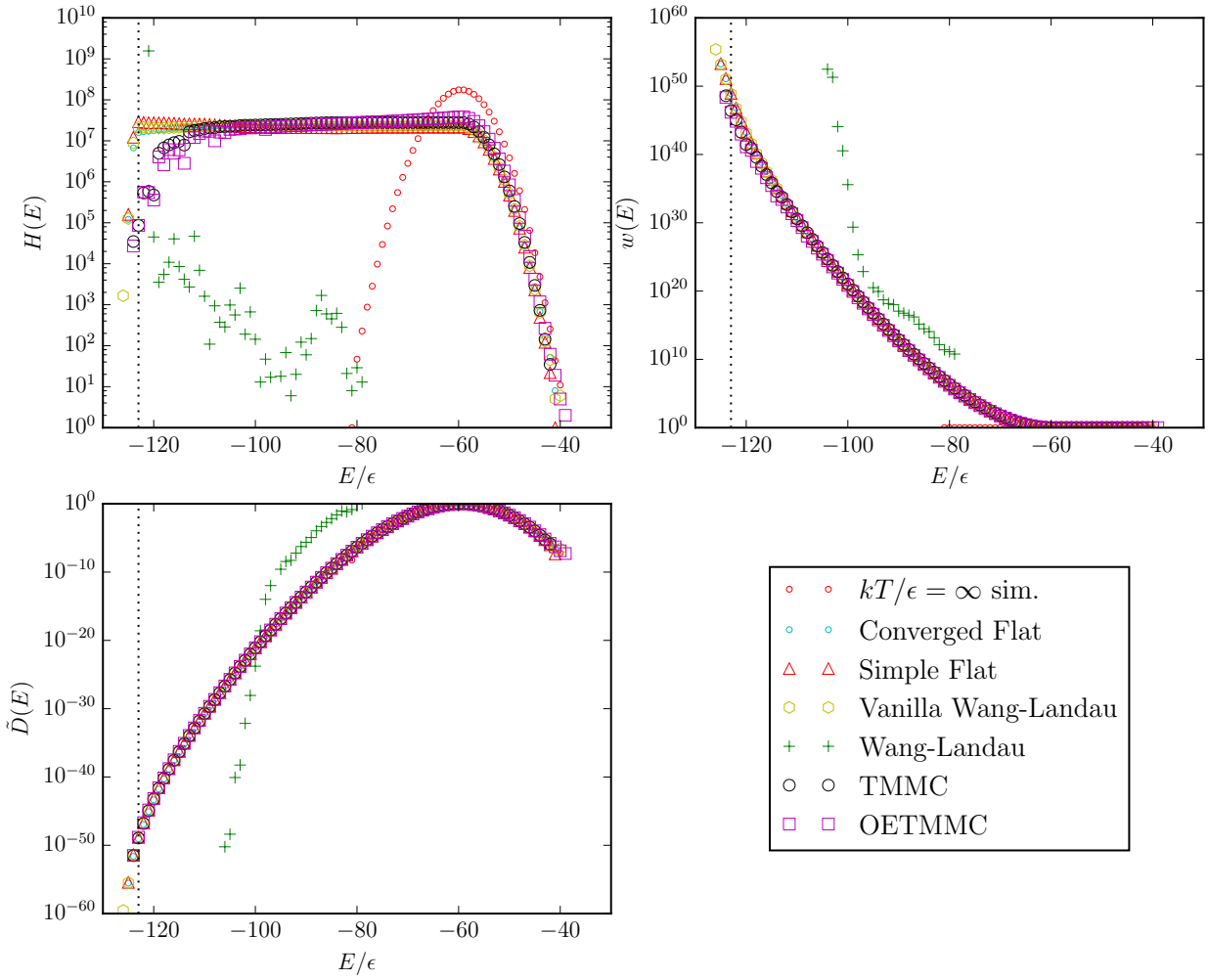

Figure 5: Example arrays computed by various histogram methods for a square-well fluid with a well width $\lambda = 1.3$, filling fraction $\eta = 0.3$, and $N = 25$ spheres after simulating for $10^5 N^2$ iterations: (top left) energy histogram, (top right) weight array, (bottom left) density of states. Vertical reference lines denote the minimum important energy E_{-}.

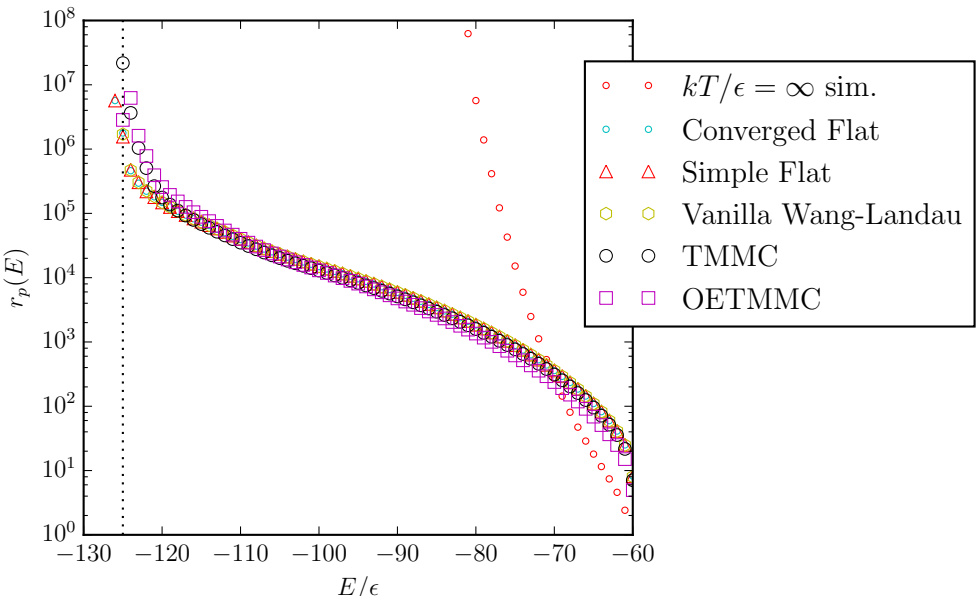

Figure 6: Example pessimistic energy sampling rates computed by various histogram methods. Sample rates for each method are averaged over 30 simulations with different random number seeds. The vertical reference line denotes the minimum important energy E_{-}.

$\tilde{D} = H/w$ and normalized to have a maximum value of 1. The vertical reference line in these plots denotes the minimum important energy E_{-}. For reference, we also provide the histogram produced by an infinite temperature (no weights) simulation.

The most immediate observation in these plots is the failure of Wang-Landau to produce meaningful results without a predefined energy range. This failure occurs because the Wang-Landau algorithm does not have a mechanism for properly setting and adjusting weights on newly observed energies. After several reset cycles of the Wang-Landau algorithm, the weight-modifying factor f_k gets exceedingly small relative to its initial value. When Wang-Landau finds new low energies, therefore, the simulation gets stuck trying to fix the weights at these energies with a factor f_k whose magnitude is appropriate for fine tuning, rather than heavy adjustment. If Wang-Landau knows the energy range of interest to begin with, it will not decrease f_k until it has observed all important energies, and adjusted their weights to some reasonable value. Due to the drastic failure of Wang-Landau, will omit its nonsensical results from some figures, as these results are distracting and unfit for drawing any conclusions.

The energy histogram in Figure 5 shows that TMMC and OETMMC spend considerably less time at important low energies than the simple flat and vanilla Wang-Landau methods. At face value, this behavior of TMMC and OETMMC is undesirable, as these are the most interesting energies for low temperatures, and the most difficult energies to sample.

High energy histogram counts at low energies mean that a simulation has collected many statistics on these energies, but the energy histogram contains no information about how correlated these statistics are. To determine how effective different simulations are at collecting uncorrelated statistics, we can look at the pessimistic sample rate $r_p(E)$, which is the mean number of simulation iterations required to sample a given energy E. This rate is roughly equal to the number of iterations I in a simulation and the pessimistic sample rate $s_p(E)$ by $r_p(E) = I/s_p(E)$; the greater the number of iterations I, the better this approximation. Figure 6 shows pessimistic sample rates averaged over 30 simulations with $N = 25$. Lower pessimistic sample rates mean that a simulation

takes less time to collect statistics which are guaranteed to be uncorrelated. By this metric, TMMC and OETMMC again perform worse than simple flat and vanilla Wang-Landau.

4.3 Thermodynamic properties and errors

Histograms and sampling rates let us speculate about the quality of a simulations, but the most important deliverable of any simulation is accurate physical results. We therefore compute, via the formula in (6), the temperature dependence of the square-well fluid's potential energy

$$U = \langle E \rangle_T \,, \tag{36}$$

the heat capacity

$$C_V(T) = \frac{\partial U}{\partial T} = \left\langle \left(\frac{E}{kT} \right)^2 \right\rangle_T - \left\langle \frac{E}{kT} \right\rangle_T^2 , \tag{37}$$

and the configurational entropy

$$S_{\text{config}}(T) = S(T) - S(T \to \infty) = k \left\langle \frac{E}{kT} + \ln Z \right\rangle_T - k \left\langle \frac{E}{kT} + \ln Z \right\rangle_{T \to \infty} . \tag{38}$$

All of these properties are directly computable from the density of states $D(E)$, and are shown in Figures 7–9 for $N = 25$ simulations, averaged over 30 simulations. When looking at these figures, one should remember that we are only confident in results down to $kT = 0.2\epsilon$ for regular simulations, and $kT = 0.1\epsilon$ for golden ones. With the exception of Wang-Landau, which has a heavy low-energy bias, all simulations agree fairly well on these properties of the square-well fluid.

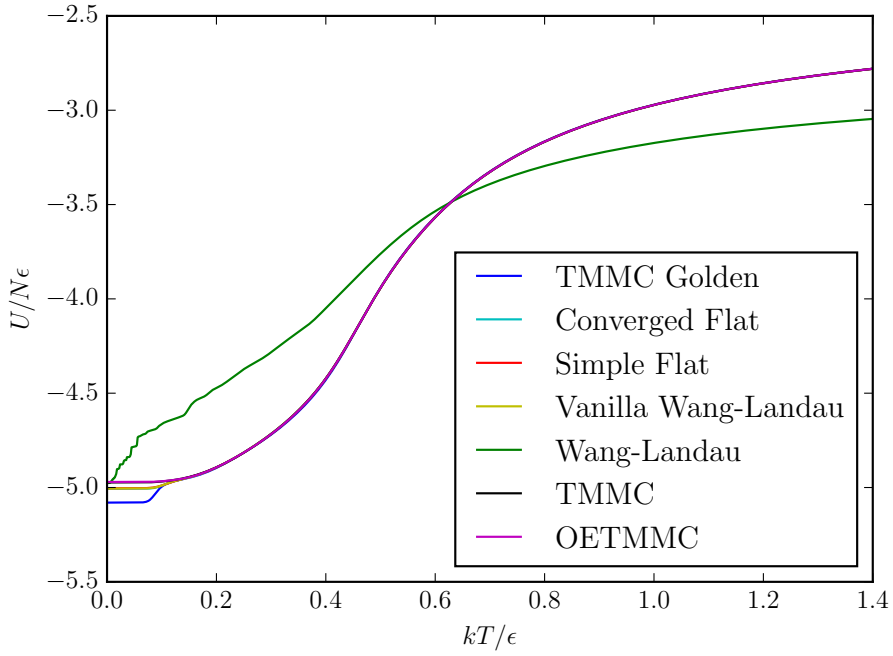

Figure 7: Specific internal energy curves averaged over 30 $N = 25$ simulations.

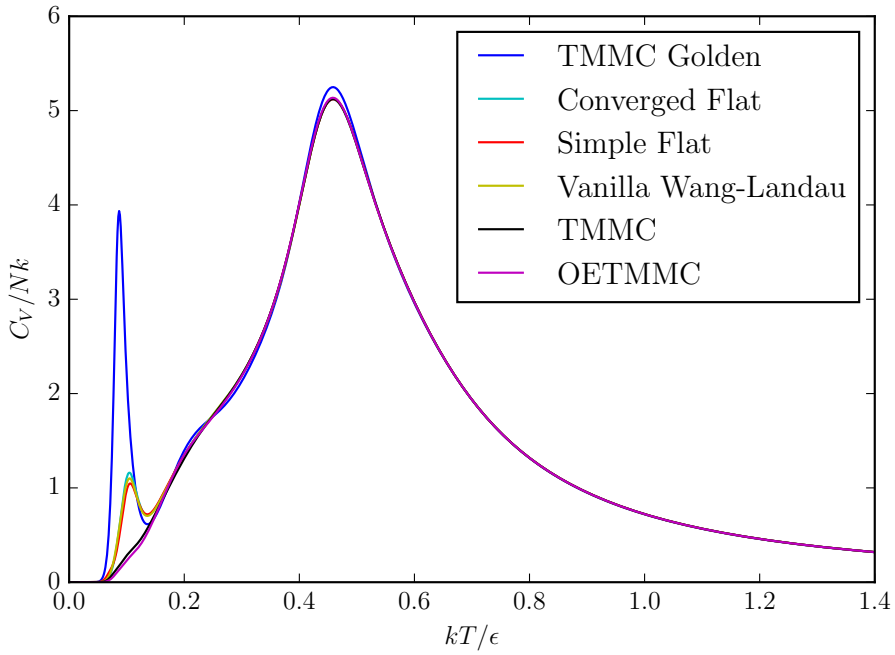

Figure 8: Specific heat capacity curves averaged over 30 $N = 25$ simulations.

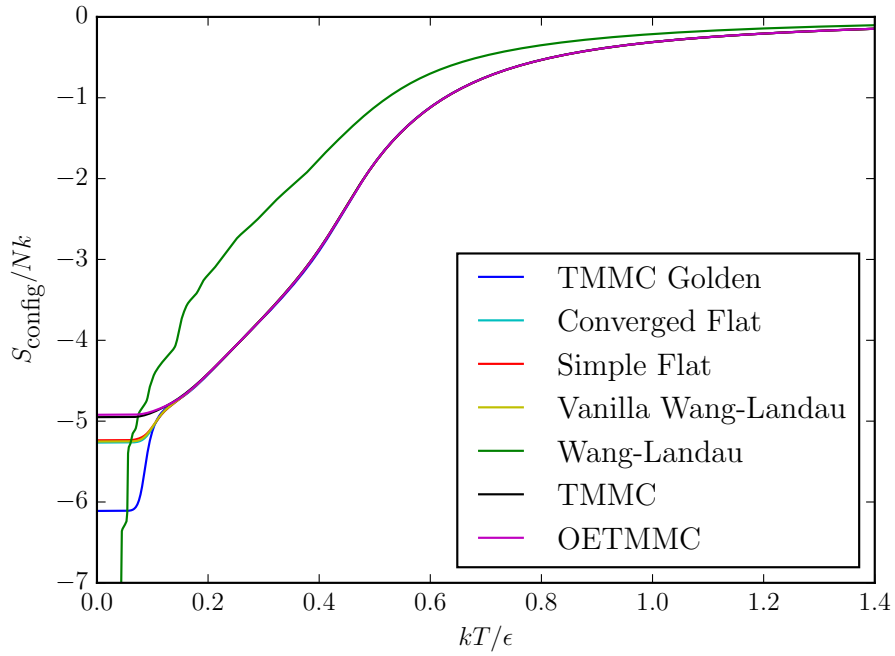

Figure 9: Specific configurational entropy curves averaged over 30 $N = 25$ simulations.

Though Figures 7–9 are interesting for studying the square-well fluid, they do not help us resolve the efficacy of different histogram methods. These figures do, however, show that the square-well fluid has a phase transition around $kT \approx 0.5\epsilon$, which justifies our choice of $kT_{\text{min}} = 0.2\epsilon$ and ensures that the histogram methods are indeed performing the nontrivial task of exploring the liquid-gas phase boundary of the square-well fluid. For the purposes of this paper, we will neglect to prove that this phase transition is indeed between liquid and gas phases, and take this fact as given.

Having a sense of how U, C_V and S_{config} behave with respect to temperature, we can now look at the how the errors ΔU, ΔC_V and ΔS_{config} relative to the TMMC golden calculations, shown in Figures 10–12. Much like the arrays in Figure 5, we provide these figures for a sense of how error curves behave. Errors (at $kT \geq 0.2\epsilon$) typically peak near the phase transition at $kT \approx 0.5\epsilon$, and drop back down.

Figures 13–15 provide a more comprehensive picture of errors from each method. These figures show the average maximum error in U, C_V, and S_{config} for different system sizes. Though errors jump around considerably in these figures, the fact that they are roughly stable (logarithmically) between $N = 15$ to 25 justifies the choice N^2 scaling for our number of initialization iterations. Future studies might use this scaling to estimate appropriate initialization time for larger systems.

Figures 16–18 provide a different visualization of the same information as in Figures 13–15, plotting the errors of all histogram methods against those of converged flat histogram weight simulations. The fact that some points fall below the diagonal reference line is strange, and indicates that some regular simulations are outperforming the converged flat ones. We do not understand why such events occur, as converged flat simulations should be using high quality golden data to construct weight arrays, and should therefore outperform all standard simulations.

As with the arrays, the error comparison figures demonstrate the inability of Wang-Landau to sample a system without a predefined energy range. The simple flat method, vanilla Wang Landau, and TMMC seem to give comparable errors, except for some anomalous cases with $N = 18$ and

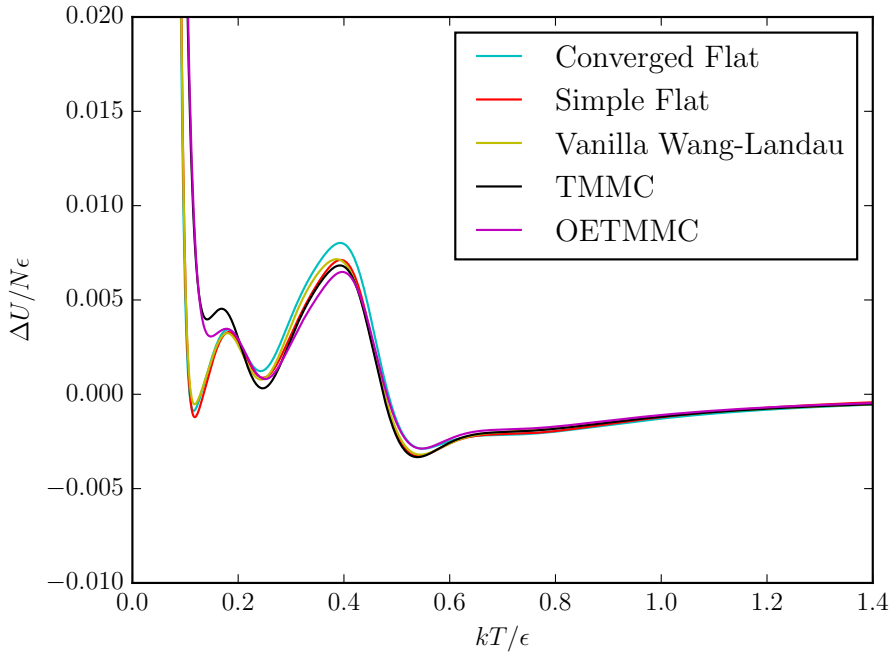

Figure 10: Errors in specific internal energy, averaged over 30 $N = 25$ simulations.

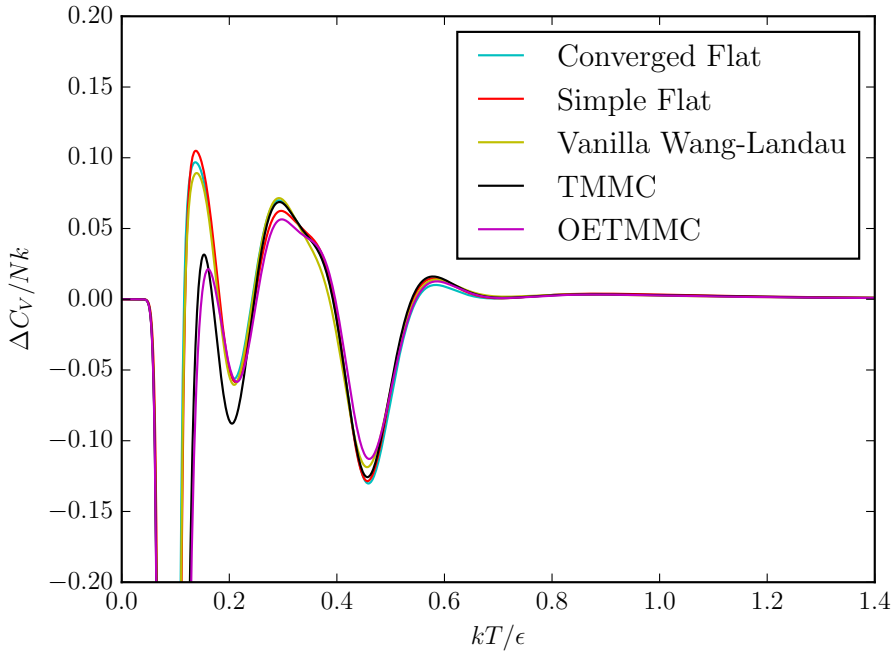

Figure 11: Errors in specific heat capacity, averaged over 30 $N = 25$ simulations.

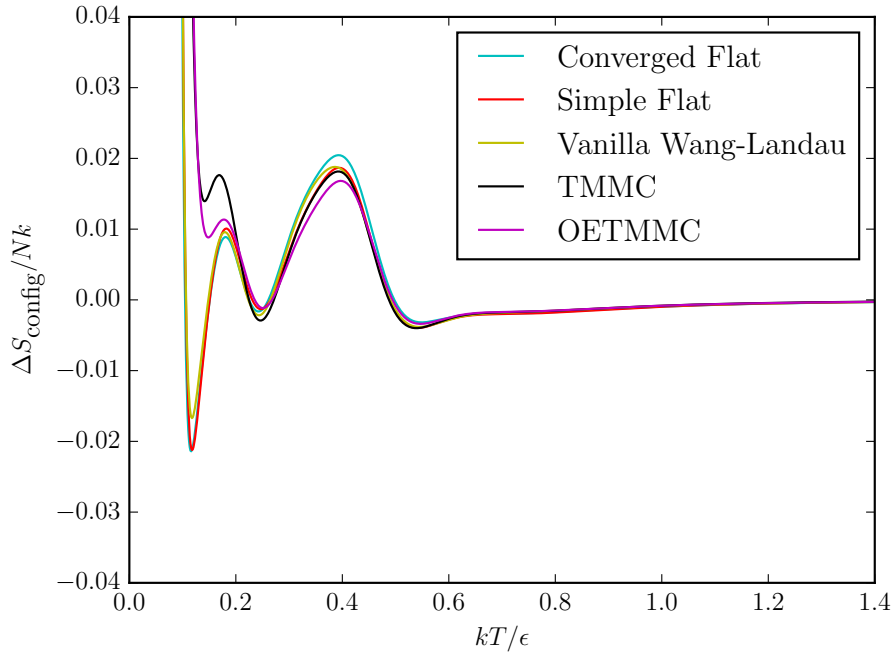

Figure 12: Errors in specific configurational entropy, averaged over 30 $N = 25$ simulations.

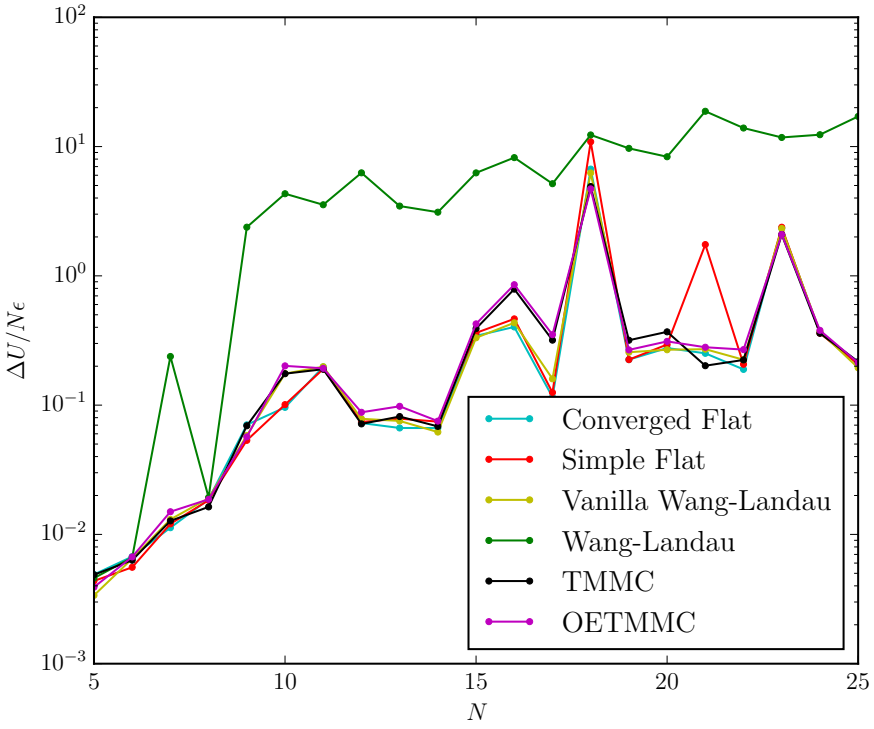

Figure 13: Specific internal energy error scaling.

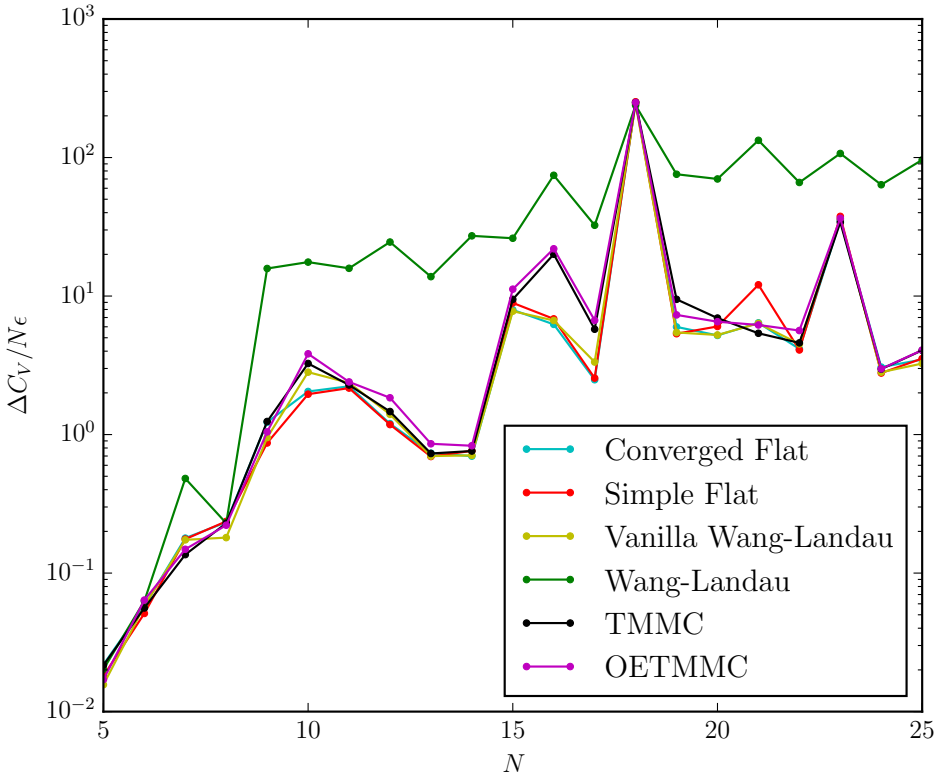

Figure 14: Specific heat capacity error scaling.

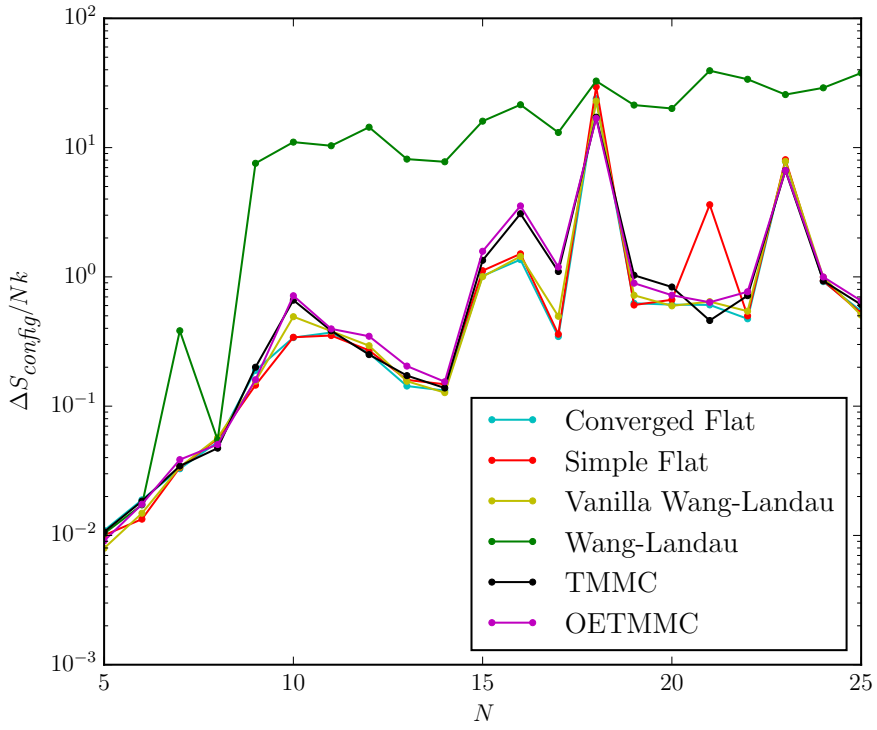

Figure 15: Specific configurational entropy error scaling.

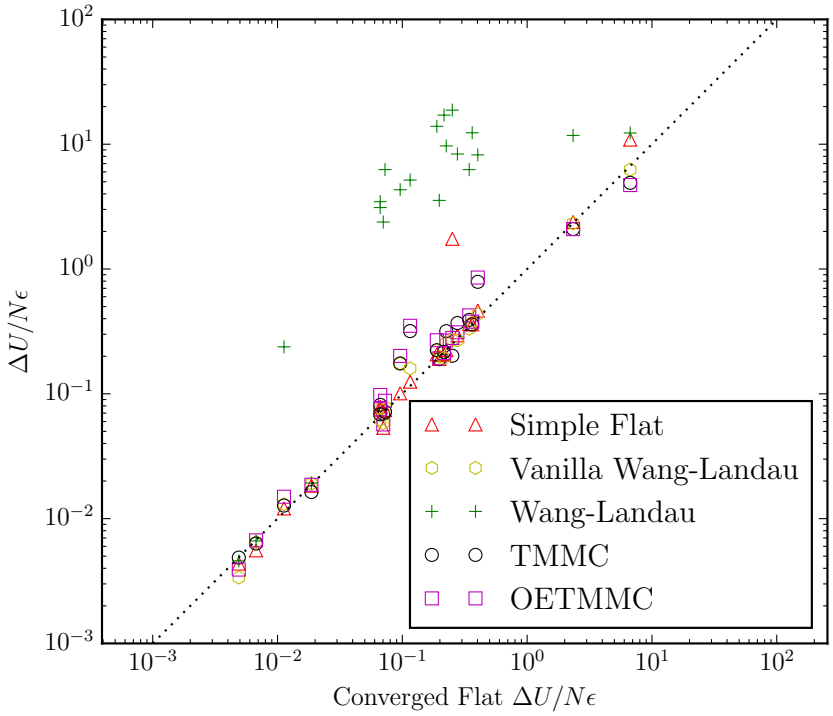

Figure 16: Specific internal energy errors, plotted against those from converged flat histogram weight simulations.

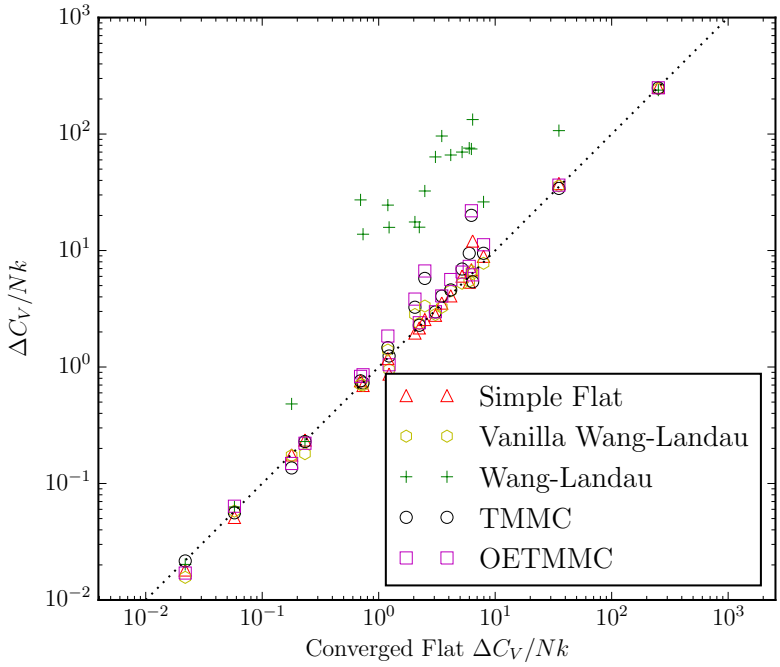

Figure 17: Specific heat capacity errors, plotted against those from converged flat histogram weight simulations.

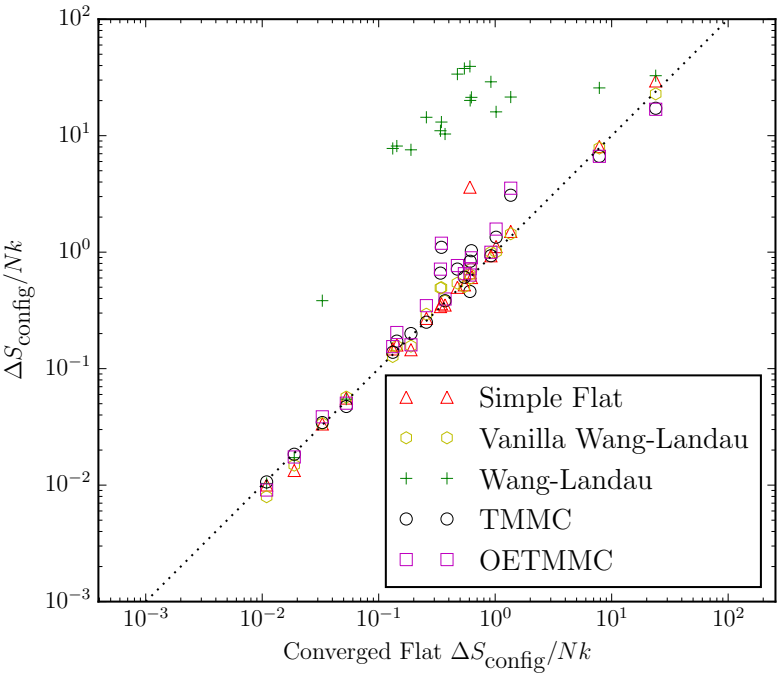

Figure 18: Specific configurational entropy errors, plotted against those from converged flat histogram weight simulations.

$N = 21$ in which simple flat has large errors. We do not know the reason for these anomalies, but they could occur if the method terminates before finding energies near E_-, in which case it would get stuck in simulation at low energies with canonical weights for the minimum temperature $kT_{\min} = 0.2\epsilon$. Finally, these error figures further confirm earlier suspicions of OETMMC's underperformance relative to TMMC, despite the fact that it is supposed to provide an improvement over the TMMC weight array.

5 Conclusions

After implementing several histogram methods and applying them to simulation of the $\lambda = 1.3$, $\eta = 0.3$ square-well fluid, we found that Wang-Landau, if given a predefined energy range, consistently yields some of the smallest errors and highest low-energy sampling rates of all histogram methods, with comparable results from TMMC and the simple flat method. Without prior knowledge of the minimum important energy of a system, however, Wang-Landau can get stuck at low energies, in which case it is incapable of recovering. Wang-Landau is therefore most appropriate for studying systems in which the entire energy range is known a priori, or in which only a given energy range is of particular interest. The simple flat method, while generally performing well, sometimes fails to initialize. If gone unnoticed, such failures can be costly in terms of computation time and resources. Our implementation of the optimized ensemble using the transition matrix, i.e. OETMMC, resulted in lower sampling rates and higher errors than the flat histogram TMMC weights. Though we have not completely ruled out that our implementation of OETMMC contains bugs, we suspect that the optimized ensemble is failing to improve TMMC either because i) the theory behind the optimized ensemble neglects the discretization of energies, or ii) OETMMC is highly sensitive to sampling error in the transition matrix. The first of these suspicions can be tested by simulating larger systems with a denser specific energy spectrum, and for which discretization should become asymptotically negligible; the second suspicion can be tested simply by initializing via TMMC for longer, thereby reducing errors in the transition matrix, before computing diffusion-optimized weights.

6 Acknowledgments

I wish to thank, first and foremost my advisor, David Roundy, for mentoring me and providing me a wealth of knowledge and expertise in various subjects of physics, numerical methods, and programming nuances. I would also like thank Paho Lurie-Gregg, with whom I frequently exchange and discuss codes and ideas, and whose hard polyhedra Monte Carlo simulations I used as the template for my own square-well codes. Lastly, I would like to thank thank Eva Morgun, who prompted me to *act* on my textbook enthusiasm for physics, and get involved in the fun and interesting world of scientific research.

References

[1] F. Wang and D. Landau, "Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram," Physical Review E **64**, 056101 (2001).

[2] R. Belardinelli and V. Pereyra, "Fast algorithm to calculate density of states," Physical Review E **75**, 046701 (2007).

[3] C. Zhou and R. Bhatt, "Understanding and improving the Wang-Landau algorithm," Physical Review E **72**, 025701 (2005).

[4] D. Landau, S.-H. Tsai, and M. Exler, "A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling," American Journal of Physics **72**, 1294–1302 (2004).

[5] C. Yamaguchi and Y. Okabe, "Three-dimensional antiferromagnetic q-state Potts models: application of the Wang-Landau algorithm," Journal of Physics A: Mathematical and General **34**, 8781 (2001).

[6] D. F. Parsons and D. R. Williams, "Globule transitions of a single homopolymer: A Wang-Landau Monte Carlo study," Physical Review E **74**, 041804 (2006).

[7] D. F. Parsons and D. R. Williams, "An off-lattice Wang-Landau study of the coil-globule and melting transitions of a flexible homopolymer," The Journal of chemical physics **124**, 221103 (2006).

[8] J.-S. Wang and R. H. Swendsen, "Transition matrix Monte Carlo method," Journal of statistical physics **106**, 245–285 (2002).

[9] S. Trebst, D. A. Huse, and M. Troyer, "Optimizing the ensemble for equilibration in broad-histogram Monte Carlo simulations," Physical Review E **70**, 046701 (2004).