Absolute free energies and equilibrium ensembles of dense fluids computed from a nondynamic growth method

Divesh Bhatt and Daniel M. Zuckerman^{a)}

Department of Computational Biology, University of Pittsburgh, Pittsburgh, Pennsylvania 15260, USA

(Received 24 July 2009; accepted 9 November 2009; published online 4 December 2009)

We demonstrate a nondynamical Monte Carlo method to compute free energies and generate equilibrium ensembles of dense fluids. In this method, based on step-by-step polymer growth algorithms, an ensemble of $n+1$ particles is obtained from an ensemble of n particles by generating configurations of the $n+1$ st particle. A statistically rigorous resampling scheme is utilized to remove configurations with low weights and to avoid a combinatorial explosion; the free energy is obtained from the sum of the weights. In addition to the free energy, the method generates an equilibrium ensemble of the full system. We consider two different system sizes for a Lennard-Jones fluid and compare the results with conventional Monte Carlo methods. © 2009 American Institute of Physics. [doi:10.1063/1.3269674]

I. INTRODUCTION

The generation of an equilibrium ensemble and calculation of the free energy of a system are two problems of fundamental importance in statistical mechanics and its applications to a wide variety of physical and biological systems. This manuscript describes the attempt to use polymer-growth ideas for sampling dense fluids and estimating the associated free energies. This is an initial proof of principle study, where we demonstrate for the first time the applicability of polymer-growth ideas to simple, dense fluids. The long term goal is to develop methods capable of growing explicit solvent around biomolecular configurations initially sampled using implicit solvent.

The ability to grow explicit solvent molecules about a solute, in a statistically correct way, could prove to be of great importance. To see why, note that time scales associated with solvent (water) molecules in molecular simulations are significantly faster than time scales of large biomolecules. However, dynamics simulations spend a considerable amount of CPU effort in simulating solvent molecules and calculating energies associated with solvent-solvent and solvent-biomolecule interactions. If successful, nondynamic growth methods could permit rapid addition of solvent molecules around biomolecules while maintaining the correct ensemble—thus, converting an implicit solvent ensemble of a biomolecule to an explicit one and, at the same time, computing the free energy associated with the process. As discussed below, polymer-growth methods^{1–3} intrinsically yield absolute free energies.

Conventional simulation techniques rely on molecular dynamics (MD) or Markov chain Monte Carlo (MCMC) to generate the equilibrium configurations and additional techniques such as thermodynamic integration to compute the free energies—in essence, addressing the two problems (free energy computation and generation of equilibrium ensemble)

separately. More sophisticated sampling algorithms have been developed⁴ but typically free energies are computed separately. In principle, we note that the absolute free energy can be computed via a single exhaustive equilibrium simulation.⁵

Methods based on thermodynamic integration⁴ are frequently used to determine the free energy difference between two states, say A and B, and rely on calculations along a path from state A to state B. The states can be characterized either by different Hamiltonians/potential energy functions or by different structures. Systems where states A and B are far apart (for example, structurally) could require a very complicated path between the states. The weighted histogram analysis method,^{6,7} umbrella sampling approaches,⁸ and nonequilibrium techniques⁹ all require extensive sampling along the A to B path.

In an effort to overcome some of the challenges of thermodynamic integration, Meirovitch and co-workers^{10–12} developed methods that compute the *absolute* free energies of a system. The relative free energies of states A and B can then be computed from their respective absolute free energies. Although several different implementations were adopted, the main idea is to compute the Boltzmann probability of a configuration by scanning through the volume of the simulation box sequentially and calculating the (approximate) “transition probabilities” in each scanned subvolume for the given configuration. The transition probabilities lead to the absolute entropies. This method was applied to computing the free energies of fluids such as argon and water.¹⁰

Polymer-growth approaches, also known as sequential MC, are central to this manuscript and were developed in 1950s. The idea behind these methods is to generate configurations of a polymer by sequentially adding segments, without dynamics. At each stage (i.e., for each added segment), an ensemble is generated. For a random trial generation of the new segment, this procedure is equivalent to a brute-force MC integration of the partition function, one degree of freedom at a time (or, more accurately, for all the degrees of

^{a)}Electronic mail: ddmzz@pitt.edu.

freedom associated with a segment). However, the trial positions of the new segment need not to be random, and this leads nonuniform weight factors. Rosenbluth weights^{1,4} are an example of such weights, where the generation of the trial position is proportional to the Boltzmann factor of the interaction energy change due to adding the new segment. Subsequent developments to enhance the quality of the ensemble at each stage (for example, by removing configurations that contribute negligibly to the ensemble), while still maintaining statistical rigor, led to ideas of pruning and enrichment,^{13,14} and resampling.³ In contrast to MCMC, growth algorithms do not depend on perturbations from an existing configuration, and, hence, do not get trapped in local energy minima. In addition, a step-by-step MC integration leads to immediate estimation of the partition function during the growth procedure.¹ In essence, the method gives the absolute free energy (or, more precisely, the free energy of the full system with respect to an ideal, noninteracting system).

This class of nondynamic growth algorithms appears to hold promise for systems that are characterized by slow dynamics and biomolecules in explicit solvent are prime examples of such systems. One of the authors with other co-workers recently applied the growth algorithm to generating ensembles and calculating free energies of small polypeptides in implicit solvent,¹⁵ and current work is underway to extend the procedure to larger polypeptides. Complementary to the generation of equilibrium ensembles of polypeptides in implicit solvent is the need to generate ensembles of solvent molecules themselves by a similar growth procedure—thereby allowing a more seamless extension to sampling polypeptides in explicit solvent. In contrast to polypeptides and polymers generally, fluid molecules are not connected by covalent bonds—and, thus, require a different treatment.

In this manuscript, apparently for the first time, we apply a growth strategy for determining the equilibrium configurations and the absolute free energy of dense Lennard-Jones (LJ) fluids. The manuscript is organized as follows. First we describe the method we use, providing theoretical background on growth and reweighting approaches. Then, we present results for two different system sizes. We calculate equilibrium density profiles and energy distributions, as well as free energies obtained via growth. We compare our results with conventional MCMC simulations to validate our method. This is followed by a discussion in Sec. V and conclusions in Sec. VI.

II. METHOD

A. Partition function and free energy

The free energy difference between a physical and a reference system is

$$F^{\text{phys}} - F^{\text{ref}} = -kT \ln \frac{Z^{\text{phys}}}{Z^{\text{ref}}}, \quad (1)$$

where Z^{phys} is the partition function of the physical system (and similarly for Z^{ref}). For a system of N distinguishable particles interacting based on a potential U^{phys} , the configurational partition function is

$$Z^{\text{phys}} = \int_V d\mathbf{r}^N e^{-\beta U^{\text{phys}}}, \quad (2)$$

where V is the volume of the system and $\mathbf{r}^N \equiv \{\mathbf{r}_1, \dots, \mathbf{r}_N\}$. A similar expression holds for Z^{ref} with N distinguishable particles with a total potential energy of U^{ref} . Although we specifically consider distinguishable particles in this manuscript (for ease of notation with respect to the growth algorithm we discuss below), indistinguishability in both the physical and the reference systems merely includes a multiplicative factor $(1/N!)$ in Eq. (2), and cancels out from the free energy difference desired in Eq. (1). We have also omitted momentum prefactors for clarity of notation. The LJ systems we study are fully specified in Sec. III.

B. Staging and reweighting

When the reference system consists of noninteracting particles, the partition function, Z^{ref} , is trivial to compute and the use of Eq. (1) can give the absolute free energy of the physical system. Such an ideal reference system has uniform density in the full $3N$ dimensional configurational space. On the other hand, most physical systems occupy only a tiny fraction of the configurational space, rendering any one-step free energy difference method [as implied in Eq. (1)] impractical. With the use of intermediate stages, the ratio of partition function in Eq. (1) can be written as

$$\frac{Z^{\text{phys}}}{Z^{\text{ref}}} = \frac{Z^{\text{phys}}}{Z_M} \left(\prod_{n=0}^{M-1} \frac{Z_{n+1}}{Z_n} \right) \frac{Z_0}{Z^{\text{ref}}}, \quad (3)$$

where $M+1$ intermediate stages are utilized and Z_n is the configurational partition function of stage n . Although the applicability of Eq. (3) is independent of the choice of the set $\{Z_n\}$, some choices are more efficient than others.

A natural (although not necessarily the most efficient) choice for the number of intermediate stages is $N+1$, with each intermediate stage, n , corresponding to $N-n$ ideal particles and n interacting particles. This corresponds to $M=N$, and we use these intermediate stages throughout this manuscript. At every stage one particle is converted from its reference state representation (ideal particle) to a representation where it interacts with the other, already converted particles. We combine this choice of intermediate stages with the use of external potentials denoted by U_i^{ext} for particle i . For an intermediate stage where n particles are already interacting, the full canonical ensemble partition function is then

$$Z_n = \int_V d\mathbf{r}_N e^{-\beta U_N^{\text{ext}}} \cdots \int_V d\mathbf{r}_{n+1} e^{-\beta U_{n+1}^{\text{ext}}} \int_V d\mathbf{r}^n e^{-\beta(U^n + \sum_i^N U_i^{\text{ext}})}, \quad (4)$$

where U^n is the potential energy of the n interacting particles. In this manuscript, $U^N \equiv U^{\text{phys}}$. The use of external potentials in Eq. (4) is elaborated later.

Before discussing the use of external potentials in Eq. (4), we consider the case when $U_i^{\text{ext}}=0$ for all i . The notation Z_n^0 is used to denote the partition functions in this scenario.

FIG. 1. A schematic one-dimensional projection of the configuration space distribution for different values of n . The target distribution is shown with a solid line, whereas intermediate distributions for two successive values of n are shown with solid lines.

Clearly, $Z_N^0 \equiv Z^{\text{phys}}$ and $Z_0^0 \equiv Z^{\text{ref}}$. With this notation, the ratio of partition functions appearing in parentheses in Eq. (3) can be written as

$$\frac{Z_N^0}{Z_0^0} = \langle e^{-\beta \Delta U_N} \rangle_{Z_{N-1}^0} \cdots \langle e^{-\beta \Delta U_1} \rangle_{Z_0^0}, \quad (5)$$

where $\Delta U_n \equiv U^n - U^{n-1}$ is the change in potential energy of the system upon converting the n th ideal particle to a real particle. In other words, a uniform distribution of an ideal particle that is converted to a real particle is reweighted by the change in the Boltzmann factor resulting from that change.

In general, stage-by-stage growth algorithms compute the ratios appearing in Eq. (3) for stage $n+1$ by “turning on” interactions for a previously ideal $n+1$ st particle (uniformly distributed) in the canonical ensemble for stage n . Any generated configuration, i , of the $n+1$ st particle is reweighted into the $n+1$ ensemble according to $\exp(-\beta \Delta U_{n+1}(i))$. Thus, starting from the first stage ($n=0$, which is an ideal gas), a properly distributed ensemble at every stage is generated before the next stage is sampled; this is called “breadth-first” sampling. To avoid a combinatorial explosion, low weight configurations can be weeded out at any stage by a statistically rigorous resampling scheme that is discussed later.

Such an unbiased staging strategy (i.e., without any external potential) works well for a few initial stages in the growth of fluid configurations but fails for generating a full ensemble (at stage N) for liquid-like densities. Figure 1 schematically illustrates the difficulties resulting from the narrowing and shifting of the configuration-space distribution as n increases. As is suggested by Fig. 1, three issues affect the efficiency of a growth algorithm. First, there is successive narrowing of the distribution with each stage—due to the newly added interactions of $n+1$ st particle. Although it is

advantageous for the accuracy of the algorithm to reweight from a broader to a narrower distribution, narrowing also leads to attrition of the ensemble (an issue that is discussed later in relation to resampling). Second, the distribution itself may be shifted for each successive stage: It is not necessary that the part of configuration space that contributes significantly to $\exp(-\beta U^n)$ also contributes significantly to $\exp(-\beta U^{n+1})$. Finally, a somewhat related issue is the possibility that reweighting during initial stages may increasingly favor parts of configuration space that are not relevant for the final, N , ensemble. One physical mechanism for this is that during initial unbiased growth, the system does not contain information about the final desired density.

It is pertinent to note that using a nonuniform distribution of the $n+1$ st ideal particle without any external potential does not ameliorate the issues raised above. The main problem is the unbiased stages themselves—which fail to overlap well with the full physical ensemble—and not the ability to sample them. Solely changing the generating distribution of ideal particles does not have any effect on the stages. We therefore employ external potentials.

C. Staging via external potentials

As mentioned above in relation to Fig. 1, one of the problems associated with growth using Eq. (5) is that the partially grown system does not contain the information about the final desired density. The use of external potentials on each added particle to maintain the density of the growing ensemble is a possible solution. Staging via external potentials leads to determining the ratio Z_{n+1}/Z_n at each stage, and instead of Eq. (5), we have

$$\frac{Z_N}{Z_0} = \langle e^{-\beta \Delta U_N} \rangle_{Z_{N-1}} \langle e^{-\beta \Delta U_{N-1}} \rangle_{Z_{N-2}} \cdots \langle e^{-\beta \Delta U_1} \rangle_{Z_0}, \quad (6)$$

where the lack of “0” superscripts indicates that the generating distributions of ideal-gas particles also use external potentials.

Density information can be used to guide the construction of external potentials. For homogeneous systems, the knowledge of density in any region of the space (simulation box) is trivial. On the other hand, an inhomogeneous system (for example, a small system with hard-wall boundary conditions) could require a short dynamic simulation to estimate the spatially varying density. Figure 2 illustrates how such information can be used to determine the external potentials for staging. The top panel shows (as a schematic) the cumulative number density profile obtained from a dynamic simulation of fluid particles in a hard-walled spherical box. The abscissa is the radial distance from the center of such a spherical box. This geometry is chosen in this schematic in view of the systems (introduced later) considered in this work. If the space is divided into bins (three are shown in Fig. 2), then the cumulative density profile gives the number of particles in each bin (e.g., bin 1 contains n_1 particles on average). A growth procedure that places the first n_1 particles in bin 1, the next $n_2 - n_1$ particles in bin 2, and finally $N - n_2$ particles in bin 3 maintain the desired density at intermediate stages.

FIG. 2. A schematic of the fluid density profile and external potentials that can aid sampling. (a) Cumulative density profile. (b) Strict external potential. (c) External potentials that allow density fluctuations. In the schematic, three different external potential bins, $i=1,2,3$, are used. Any particle n , such that $n_1 < n < n_2$, is “nominally” in bin $i=2$, and the external potentials on that particle are shown in panels (b) and (c) for strict potential and one that allows density fluctuations, respectively. Here, r is the radial distance from the origin.

The strict procedure just described is equivalent to adding an external potential U_n^{ext} on particle n as depicted in the middle panel of Fig. 2, where particle n , with $n_1 < n < n_2$, resides exclusively in bin 2. However, such an infinite potential well does not allow for density fluctuations among bins that might occur in the system and is inherently incorrect. To overcome this problem, we use an external potential that remains finite throughout the space, as illustrated in the bottom panel of Fig. 2. Although each particle is more likely to reside in its own bin, it is allowed to “escape” to other bins—thus permitting density fluctuations that are intrinsic to the targeted physical ensemble (without any external potential). Staging via external potentials also has an approximate intuitive physical interpretation as a mean field potential applied by the particles not yet grown.

Another advantage of staging via external potentials is apparent when generating configurations of liquids below the critical temperature. In absence of an external potential, the initial growth of the system will correspond to two-phase coexistence. Intermediate ensembles will contain liquidlike and vaporlike regions. Presumably, such a two-phase growth would exacerbate the above-mentioned problems associated with unbiased staging.

We now summarize the growth procedure from n to $n+1$ real particles—i.e., from the Z_n to Z_{n+1} ensemble. In the Z_n ensemble, the $n+1$ st (ideal) particle is distributed according to U_{n+1}^{ext} , a spatially dependent external potential. This ideal particle is then converted into a real particle that keeps

the same external potential, but now interacts with the n previously grown particles. Thus, the weight of each generated configuration, i , of the $n+1$ st real particle is given by $\exp(-\beta\Delta U_{n+1}(i))$, as discussed above. We also note that there is no restriction on the form or dimensionality of the external potential. Indeed, as discussed later, for more complex systems, more complicated form of the external potential may be required. We note that homogeneous, periodic systems should also require an external potential to maintain the density of the growing ensemble.

D. Free energy

The final free energy F^{phys} is obtained from Eqs. (1) and (3). Staging via external potentials does not give the required ratio $Z^{\text{phys}}/Z^{\text{ref}}$ in Eq. (3) directly. Rather, it gives the product in parentheses on the right side of Eq. (3). The initial and final ratios must be estimated separately. The last fraction in Eq. (3) factorizes for N ideal particles and is simply

$$\frac{Z_0}{Z^{\text{ref}}} = \prod_{i=1}^N \langle e^{-\beta U_i^{\text{ext}}} \rangle_{\text{ideal}}, \quad (7)$$

where the averages are over uniform distributions (Z^{ref} ensemble). The individual factors are easily obtained analytically or by numerical integration.

It is advantageous to rewrite the first ratio on the right of Eq. (3) because the Z^{phys} distribution is “broader” than the Z_N distribution. The reciprocal can be expressed as

$$\frac{Z_N}{Z^{\text{phys}}} = \frac{1}{\langle \prod_{i=1}^N e^{-\beta U_i^{\text{ext}}} \rangle_{Z^{\text{phys}}}}. \quad (8)$$

By rewriting the ratio in this manner, a broader distribution is used to probe a narrower distribution.

Thus the problem of determination of the free energy is completely phrased such that a broader distribution is always used to probe a narrower one (or, an “insertion” strategy).¹⁶ The use of Eq. (8) may be considered a “look back” strategy, which requires a conventional dynamic simulation (either MCMC or MD) in the Z^{phys} ensemble. For simple fluids, such dynamic simulations are straightforward to perform. In this manuscript, we use this look back strategy for estimating the free energy difference for this final stage (staged-growth stages—given in parentheses in Eq. (3)—always utilize look back). However, as mentioned in Sec. I, dynamic simulations for complex systems are not always feasible. Although we do not consider such systems in this manuscript, the “ $N \rightarrow \text{phys}$ ” free energy difference would then need to be determined via the direct ratio Z^{phys}/Z_N , which is

$$\frac{Z^{\text{phys}}}{Z_N} = \left\langle \prod_i e^{+\beta U_i^{\text{ext}}} \right\rangle_{Z_N}. \quad (9)$$

In other words, a reweighting from a narrower to a broader distribution may be required to estimate the free energy in such cases. We return to this issue in Sec. II E. We note that Eqs. (7) and (8) imply that the individual external potentials can be shifted by constant values that cancel out [as they must in the overall ratio of Eq. (3)].

E. Equilibrium configurations from growth

Conventional dynamics simulations often can generate equilibrium configurations efficiently for simple fluids, but the dynamics could be slow for complex systems. Although such systems are not considered in this initial study, nondynamic methods may be useful for generating equilibrium configurations in such cases. Thus, in this initial study, we probe the utility of nondynamic growth to generate equilibrium configurations of dense simple fluids.

As mentioned above, the staged-growth procedure generates the equilibrium Z_N ensemble starting from the broader Z_0 ensemble. Consequently, the remaining issue is generating the equilibrium configurations in the required physical ensemble, Z^{phys} , from the “narrower” Z_N ensemble—i.e., via Eq. (9). Each configuration must be reweighted by the right side of Eq. (9), on top of any weights already present in the Z_N ensemble, to generate the configurations in Z^{phys} ensemble.

It is, therefore, useful to utilize a set of external potentials that perturbs the Z_N ensemble from the Z^{phys} ensemble minimally while still allowing for the growth of the Z_N ensemble using stage-by-stage growth. In addition to employing minimally perturbative staging, it is also possible to ameliorate the effects of a broadening distribution using brief dynamical simulation¹⁷ as an “enrichment” strategy; however, the present study does not exploit this strategy.

F. Resampling

In the particular staging strategy we use, the system is grown from stage n to $n+1$ by generating k trial positions of the $n+1$ st particle distributed according to U_{n+1}^{ext} for each configuration in the full Z_N ensemble; equivalently, k copies of the $n+1$ st particle are converted from ideal to real particles. If the number of configurations already generated in the Z_n ensemble is C_n , this leads to kC_n configurations generated for the Z_{n+1} ensemble. A significant majority of these kC_n configurations contribute to the ensemble with very low weights. Keeping all the generated configurations would not only lead to a combinatorial explosion with each successive stage, but, in addition, most of the generated configurations are not relevant to the required ensemble. We therefore prune configurations with low weights via a statistically rigorous resampling procedure.³ In its basic form, resampling keeps a configuration, i , at stage $n+1$ with probability given by $W_{n+1}(i)/W_{n+1;\text{max}}$, where $W_{n+1}(i)$ is the weight of the generated configuration i in the $n+1$ th ensemble, and the subscript “max” refers to the maximum weight among the kC_n configurations. After such a resampling, all the remaining configurations at stage $n+1$ have the same weight, $W_{n+1;\text{max}}$.³

In the present study, we employ a procedure called “optimal resampling”³ because we found that it is quite likely for low-weight configurations at current stage to prove significant at later stages. (In our experience, this is a distinct possibility despite staging via external potentials.) Optimal resampling aims to keep approximately the same number of configurations in the ensembles across stages. In this scheme, a configuration i is accepted with probability

$$\min\left\{1, \frac{W_{n+1}(i)}{W_{n+1;c}}\right\}, \quad (10)$$

where $W_{n+1;c}$ is a stage-specific constant chosen to maintain approximately a desired number of configurations in the ensemble. Each accepted configuration with weight less than $W_{n+1;c}$ assumes a weight of $W_{n+1;c}$, whereas any configuration with weight greater than $W_{n+1;c}$ keeps its own weight.

Maintaining different weights for configurations in a particular ensemble is qualitatively equivalent to keeping different numbers of copies of the configurations. Thus, care must be taken to carry these weights over to subsequent stages. In particular, W_{n+1} is a product of weight of the configuration in the $n+1$ st stage and the weights (after resampling) of the parent configurations in each of the preceding n stages.

G. Validation

We performed checks on both the free energy estimates and equilibrium sampling produced by the growth procedure. Staged growth gives the ratio Z_{n+1}/Z_n for all n in a single simulation. This ratio can also be obtained via conventional MC simulation with, for example, Widom’s particle insertion method.¹⁸ In contrast to a single growth simulation, conventional MC requires individual simulations for each n . Accordingly, we performed conventional MC to determine Z_{n+1}/Z_n for a few intermediate values of n to validate the free energies obtained via staged growth.

To check the equilibrium sampling of the growth procedure, we performed conventional canonical MC simulations in both the Z^{phys} and Z_N ensembles. The Z_N ensemble is obtained directly from staged growth but the Z^{phys} ensemble must be reweighted from Z_N . Careful examination of both Z_N and Z^{phys} enables isolation of the effects of reweighting from a narrower to a broader distribution, as discussed in Sec. II E.

III. MODEL AND SYSTEM DETAILS

In this manuscript, we focus on developing and validating the growth algorithm for computing absolute free energies and concurrent generation of equilibrium configurations of dense fluids in the canonical ensemble. For this initial study, we keep issues related to the complexity of force fields and the system size to a minimum. We use a truncated LJ fluid such that $U^{\text{phys}} = \sum_{i>j} u_{ij}^{\text{LJ}}$, where

$$u_{ij}^{\text{LJ}} = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right], & r_{ij} < 1.7\sigma \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where ϵ is the energy well depth, σ is the size parameter, and r_{ij} is the distance between atom pair ij . We simulate the LJ fluid at high, liquidlike density: $N\sigma^3/V=0.8$. We study two different system sizes ($N=32$ and 100), both at $kT/\epsilon=1.0$. We note that for large LJ systems, the density of 0.8 is significantly above the coexistence density at $kT/\epsilon=1.0$. For ease of implementation, we use hard-wall boundary conditions and a spherical simulation box. The use of more common periodic-boundary condition and a cubic box does not add any complexity (in contrast to system size and force fields) to the algorithm. Our choice of a relatively small cut-

TABLE I. Generating probability of a particle n in a particular spherical bin (total two bins) to obtain the results shown for a 32-particle system.

	$0 < r < 1.05$	$1.05 < r < 2.12$
$1 \leq n \leq 5$	0.7	0.3
$6 \leq n \leq 32$	0.15	0.85

off stemmed from preliminary studies of small, periodic systems.

Optimal resampling requires parameter selection to govern ensemble sizes. For the 32-particle system, we choose $W_{n,c}$ for each n so that between 49 000 and 50 000 configurations remain after resampling; and we set $k=400$. Thus, up to 20×10^6 configurations are generated at each stage. The error bars for the growth results are twice the standard error of the mean of 20 independent growth simulations. For the 100-particle system, $W_{n,c}$ is chosen such that between 800 and 1000 configurations remain after resampling, $k=20\,000$, and the standard errors are calculated from 30 independent simulations. The above values are chosen to yield good statistics—although optimization is not explicitly performed.

For each system size, we only use two external potentials to direct the growth. The external potentials are specified via the generating probabilities of ideal particles in each bin and the bin radii as described in Table I for the 32-particle system and in Table II for the 100-particle system. As mentioned in Sec. II E, it is advantageous to use the weakest possible external potential to achieve the narrow-to-broad reweighting required for generating equilibrium configurations. However, fully unbiased staging (i.e., no external potential) failed even for the 32-particle system. We found, by trial and error, that the use of two external potentials to direct the growth was sufficiently simple.

IV. RESULTS

We show growth-based results—both equilibrium distributions and free energies—for 32- and 100-particle LJ fluids, as well as a comparison with independent MCMC simulations. All the values reported are made dimensionless with respect to the LJ distance and energy parameters. Thus, $r = r/\sigma$, $E = E/\epsilon$, and $\rho = \rho\sigma^3$. Error bars represent twice the standard error, as described above.

A. 32 Lennard-Jones particles

We first examine ensemble properties. A comparison of the density profiles obtained in the Z_N ensemble via growth (circles) and a conventional MC simulation (solid line) is shown in Fig. 3. The hard wall boundary condition results in

TABLE II. Generating probability of a particle n in a particular spherical bin (total two bins) to obtain the results shown for a 100-particle system.

	$0 < r < 2.28$	$2.28 < r < 3.10$
$1 \leq n \leq 40$	0.6	0.4
$41 \leq n \leq 100$	0.3	0.7

FIG. 3. Number of particles at particular radial locations for a 32-particle system in the Z_N ensemble.

significant structuring of the density profile, and the staged growth method matches the results from the conventional canonical MC simulation. Another key quantity of interest is the energy distribution. Figure 4 shows the distributions obtained via staged growth and conventional MC. Again, there is good agreement. From Figs. 3 and 4, it is clear that the staged growth is able to reproduce the correct and precise equilibrium distribution in the Z_N ensemble.

A more important goal is to generate a correct equilibrium distribution in the Z_{phys} ensemble without any external potential. As mentioned in Sec. II E, this requires a reweighting from a narrower to a broader distribution. Figures 5 and 6 show the corresponding results for the density profile and the energy distribution, respectively, for Z_{phys} ensemble obtained via staged growth and conventional MC simulations. Again, good agreement is obtained between the staged-growth procedure and conventional MC—indicating that the growth procedure is capable of generating a good representation of the equilibrium system in the desired physical ensemble as well.

The ratio of partition functions, Z_{n+1}/Z_n , is obtained for each stage of the growth from the sum of the weights generated for stage $n+1$. This can be done before or after resampling. We calculate the sum after resampling while taking into account the fact that configurations rejected after resampling have zero weights. A plot of the partition function ratios obtained via staged growth for each n is depicted in Fig. 7. The apparent break at $n=4$ is due to the two-level external potential that changes to a different value for $n > 4$. The two different external potentials are given (as generating

FIG. 4. Energy distribution a 32-particle system in the Z_N ensemble.

FIG. 5. Number of particles as a function of radial location for a 32-particle system in the physical ensemble.

probabilities) in Table I. Due to this, the difference in the partition functions for a system with $n=4$ and $n=5$ additionally incorporates the difference in the external potentials—leading to the apparent break. The results are correct for the chosen external potentials. Partition function ratios for a few values of n using Widom's particle insertion method in conventional canonical MC simulations in the Z_n ensemble are also shown in Fig. 7. The agreement between values obtained via staged growth and those obtained by conventional MC simulations is good.

The results depicted in Fig. 7 lead to the free energy difference between the Z_N and Z_0 ensemble, $(-kT/\epsilon)\ln Z_N/Z_0 = -15.82 \pm 0.03$. To obtain the physical free energy, we calculate the product $(Z^{\text{phys}}/Z_N)(Z_0/Z^{\text{ref}})$, which cancels out any constants in the external potentials. We again note that Eq. (8) is used to compute the first of the two ratios so that the broader Z^{phys} ensemble is used to probe the narrower Z_N ensemble. Combining the result with $(-kT/\epsilon)\ln Z_N/Z_0$ leads to $(F^{\text{phys}} - F^{\text{ref}})/\epsilon = -15.01 \pm 0.04$. Again, the error bars are twice the standard error of mean from 20 independent simulations.

B. 100 Lennard-Jones particles

The results for the larger system are similar to those for the smaller system albeit with more statistical noise. The equilibrium density and the energy distributions for the Z^{phys} ensemble are shown in Figs. 8 and 9, respectively. In both figures, the results obtained via MCMC are also shown (as solid lines). A typical error bar for a small r obtained via

FIG. 6. Energy distribution of a 32-particle system in the physical ensemble.

FIG. 7. Ratio of subsequent partition functions as a function of n obtained via growth in the Z_N ensemble for the 32-particle system. Results obtained using conventional MCMC simulations are shown for three values of n for comparison.

MCMC is shown by the dashed lines. At small r , there are very few particles in each bin—leading to larger errors. For example, the average number of particles for $0 < r < 0.5$ is approximately 0.5. The density profile obtained via growth shows good agreement with MCMC. The energy distribution, although significantly noisier than for the smaller system, also agrees within the errors. We also found that a more stringent set of external potentials leads to easier generation of the Z_N ensemble, at the price of even noisier results for the Z^{phys} ensemble due to the resulting narrowing of the Z_N ensemble. In other words, the statistical noise is not just a function of the system size but also of the set of external potentials that are used to generate the Z_N ensemble—since they will, subsequently, have to be reweighted to generate the Z^{phys} ensemble.

Figure 10 shows Z_{n+1}/Z_n as a function of n . The errors for growth are smaller than the symbols. Results from MCMC are also shown for several different values of n in the figure. The agreement of growth MC with MCMC is, again, good. From the results of Fig. 10 and an independent dynamic simulation in the physical ensemble to estimate Z_N/Z^{phys} [again, using Eq. (8)], we obtain $(F^{\text{phys}} - F^{\text{ref}})/\epsilon = -65.92 \pm 0.44$. Again, a precise value of the absolute free energy is obtained.

FIG. 8. Number of particles as a function of radial location for a 100-particle system in the physical ensemble. The error bars decrease with increasing r due to increasingly more particles occupying the bins.

FIG. 9. Energy distribution a 100-particle system in the physical ensemble.

C. Ensemble quality

The efficacy of the growth method depends on the generation of a proper ensemble at each stage. Without a proper ensemble, the free energy estimates for each stage are erroneous. Thus, a large number of statistically independent configurations should be generated at each stage. Although the optimal resampling scheme can be used to accept as many configurations as desired by adjusting the weight $W_{n;c}$ [Eq. (10)] at each stage n , the ensemble quality must be assessed objectively. A better measure of the size of the ensemble is given by the ratio $f_n \equiv \sum_i W_n^i / (kC_n W_n^{\max})$, which estimates the fraction of configurations that influence a typical ensemble average.¹⁷

Figure 11 shows the plot of f_n as a function of n for the 100-particle system of Fig. 3. Clearly, f_n drops off rapidly with increasing n . This necessitates the generation of a large number of trial configurations at each stage. For example, f_n for the latter stages ($n \sim 80$) is approximately 10^{-4} ; hence, only 2000 out of 2×10^7 generated configurations contribute to the ensemble averages. For unbiased staging, we observed that f_n decayed much more drastically and the generation of a proper ensemble even at very small n was not feasible. A strategy that increases f_n for all n makes the generation of the Z_N ensemble more efficient. Conceivably, some sets of U^{ext} can give better f_n values than those shown in Fig. 11. However, if these sets lead to a poorer overlap of the Z_N ensemble with the Z_{phys} ensemble (indicated by the corresponding $f_{N \rightarrow \text{phys}}$), an increase in f_n may not be the most judicious. For

FIG. 10. Ratio of subsequent partition functions as a function of n obtained via growth in the Z_N ensemble for the 100-particle system. Results obtained using conventional MCMC simulations are also shown for four values of n for comparison. FIG. 11. Fraction of particles left after simple resampling for each stage in the 100-particle system growth, with $f_{N \rightarrow \text{phys}}$ shown with a dashed line.

the set of external potentials used to generate Fig. 11, $f_{N \rightarrow \text{phys}} = 0.22$. Compared to f_n for most values of n , $f_{N \rightarrow \text{phys}}$ is rather large due to relatively weak external potentials used, as shown in Table II. Correspondingly, good results for equilibrium distributions in the required physical ensemble are obtained if the set $\{U^{\text{ext}}\}$ is chosen so that it perturbs the system least from the Z^{phys} ensemble while still allowing for the generation of a full Z_N ensemble. The external potentials defined in Tables I and II reflect an attempt to achieve this balance.

D. CPU usage

Each independent simulation to generate the 32-particle Z_N ensemble via growth took approximately 10 min of single CPU time (3.0 GHz, 2.0 Gbyte random access memory). For the 100-particle system, the corresponding time was 90 min. Because the error bars are calculated from the standard error of the mean, they reflect uncertainty obtained in $20 \times 10 \text{ min}^2$ for the 32-particle system and in $30 \times 90 \text{ min}^2$ for the 100-particle system. For the computation of free energy difference using the look back strategy of Eq. (8), additional computational time was required for MCMC simulations in the Z^{phys} ensemble.

In this initial study, we do not attempt to quantify the efficiency of the method since the focus is on presenting a novel algorithm. The times given above reflect our present implementation, and, as we discuss below, several improvements may be possible to improve the efficiency.

V. DISCUSSION

A. Potential applications

Staged growth procedures are nondynamic, and thus do not suffer from dynamic bottlenecks inherent in conventional algorithms. Growth also provides the absolute free energy. One important goal is to grow “explicit” solvent around an “implicitly solvated” biomolecule obtaining the corresponding ensemble and solvation free energy. A few groups have already used polymer-growth algorithms for computing the free energy of polymers.^{1,13,14} One of the authors and co-workers also used growth with resampling to generate configurations as well as free energies of small polypeptides in implicit solvent.

The growth algorithm appears to be well suited for clusters. In fact, the growth of the systems discussed in this work is equivalent to growth of a (finite-temperature) cluster with additional external potentials.

B. Challenges and possible improvements

For the simple and small LJ systems considered in this initial study, some optimization of parameters—such as the ensemble size at each stage, the number of trial moves of the ideal particle for each existing configuration, and the set of external potentials—was required to generate the desired physical ensemble. This is expected to become more crucial for larger systems where, instead of manual optimization, a procedure to perform the optimization in an automated way may be desirable. Furthermore, for molecules with orientation-dependent potentials, such as water, the uniform external potentials that are used in this work may not be efficient.

Fortunately, several improvements can be foreseen. One possibility, applicable to complex systems, would be to use configurations of the full system (perhaps obtained via short dynamic simulations) to determine the external potentials. In addition, we currently generate the equilibrium configurations in Z^{phys} ensemble from the Z_N ensemble in a single stage. For more difficult systems, additional stages could be used, perhaps in conjunction with other growth strategies such as rejection control.³ Brief “annealing” simulations could also be run at intermediate stages.¹⁷

C. Comparison with related methods

Configurational-bias Monte Carlo (CBMC) method^{19–21} is based on polymer growth algorithms adapted to MCMC. Although CBMC has not been applied to generating ensembles of atomic fluids, as described here, it is instructive to compare staged growth with CBMC. In CBMC, a whole polymer is grown on a segment-by-segment basis by generating several trial positions for each segment and choosing a trial position of the segment with a probability proportional to its weight. Due to the subsequent application of the detailed balance in the Markovian framework, generation of the full ensemble for each segment is not required. In practice, a handful of trial positions is generated and only one segment is chosen for the next stage. Such a construct is necessary in the Markovian framework because one new configuration is generated from one old configuration. In contrast, the growth algorithm requires a full ensemble at each stage—leading to a direct computation of the absolute free energies.

There is another related class of methods that grow ensembles of fluid configurations (and calculate related free energies) although in a dynamic framework.^{22–24} These methods utilize several strategies: transition matrices for exchanging between macrostates with different numbers of particles,²² replicas with different numbers of particles,²³ or associated Landau free energies with different numbers of particles.²⁴ These approaches derive from detailed balance criteria for transforming between ensembles with different numbers of particles. Due to the dynamic detailed balance nature of these simulations, the free energy estimates rely on

overlap between probability distribution in both the insertion and the deletion direction. Presumably such approaches could encounter the usual overlap difficulties between the two directions when applied to dense fluids.⁴ We note that free energy calculation for our nondynamic growth scheme is based solely on the insertion direction.

As mentioned in Sec. I, Meirovitch and co-workers^{10–12} developed “scanning” algorithms to compute the absolute free energies of fluids. Scanning methods compute the Boltzmann probability of a configuration by scanning through the volume of the simulation box sequentially and calculating the (approximate) transition probabilities in each scanned subvolume for the given configuration. The required probability density of a configuration of N particles, $d\mathbf{r}^N \phi(\mathbf{r}^N)$, can be written as $d\mathbf{r}_1 \phi(\mathbf{r}_1) d\mathbf{r}_2 \phi(\mathbf{r}_2 | \mathbf{r}_1) d\mathbf{r}_3 \phi(\mathbf{r}_3 | \mathbf{r}_1 \mathbf{r}_2) \cdots$. The absolute entropy (and, thus, the free energy) of a state can be computed from the generated (Boltzmann) probabilities. In contrast to the growth algorithm presented here, scanning methods do not attempt to generate a full ensemble for intermediate values of n since the scanning occurs with respect to subvolumes rather than particles. For calculating the free energy of the full N particle ensemble, this distinction is, perhaps, unimportant. However, intermediate-stage free energies are of interest in nucleation and cluster growth. In addition, the most efficient scanning method to calculate the free energy uses MCMC to generate configurations, and, hence, depends on dynamics to provide proper sampling.

VI. CONCLUSIONS

We have demonstrated a novel nondynamical MC method to generate ensembles of dense fluids and to compute their absolute free energies. The method involves a staged growth of the full ensemble, with each stage including one additional particle. Successful application of the growth strategy required generation of a large number of trial positions of the new particle, and the use of external potentials to maintain the density of the growing system. We applied the method to dense systems of LJ fluids. Comparisons of the results obtained via growth MC and conventional MCMC validate our procedure: Both the equilibrium ensemble and the free energies are obtained correctly. We suggested possible improvements for more complex systems, as well as potential applications of the growth strategy.

ACKNOWLEDGMENTS

The authors acknowledge Artem Mamonov, Ying Ding, Xin Zhang, and Hagai Meirovitch for helpful discussions. This work was supported by the NIH (Grant No. GM076569) and the NSF (Grant No. MCB-0643456).

¹M. N. Rosenbluth and A. W. Rosenbluth, *J. Chem. Phys.* **23**, 356 (1955).

²J. M. Hammersley and K. W. Morton, *J. R. Stat. Soc. Ser. B (Methodol.)* **16**, 23 (1954).

³J. S. Liu, *Monte Carlo Strategies in Scientific Computing* (Springer, New York, 2004).

⁴D. Frenkel and B. Smit, *Understanding Molecular Simulations* (Academic, San Diego, 2002).

⁵M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, Oxford, 1987).

⁶A. M. Ferrenberg and R. H. Swenson, *Phys. Rev. Lett.* **63**, 1195 (1989).

- ⁷S. Kumar, D. Bouzida, R. H. Swenson, P. A. Kollman, and J. M. Rosenberg, *J. Comput. Chem.* **13**, 1011 (1992).
- ⁸G. M. Torrie and J. P. Valleau, *J. Comput. Phys.* **23**, 187 (1977).
- ⁹F. M. Ytreberg and D. M. Zuckerman, *J. Comput. Chem.* **25**, 1749 (2004).
- ¹⁰R. P. White and H. Meirovitch, *Proc. Natl. Acad. Sci. U.S.A.* **101**, 9235 (2004).
- ¹¹H. Meirovitch, *J. Chem. Phys.* **111**, 7215 (1999).
- ¹²H. Meirovitch, *Phys. Rev. A* **32**, 3709 (1985).
- ¹³F. T. Wall and J. J. Erpenbeck, *J. Chem. Phys.* **30**, 634 (1959).
- ¹⁴P. Grassberger, *Phys. Rev. E* **56**, 3682 (1997).
- ¹⁵X. Q. Zhang, A. B. Mamonov, and D. M. Zuckerman, *J. Comput. Chem.* **30**, 1680 (2009).
- ¹⁶D. Wu and D. A. Kofke, *Phys. Rev. E* **70**, 066702 (2004).
- ¹⁷E. Lyman and D. M. Zuckerman, *J. Chem. Phys.* **127**, 065101 (2007).
- ¹⁸B. Widom, *J. Phys. Chem.* **86**, 869 (1982).
- ¹⁹J. I. Siepmann and D. Frenkel, *Mol. Phys.* **75**, 59 (1992).
- ²⁰D. Frenkel, G. C. A. M. Mooij, and B. Smit, *J. Phys.: Condens. Matter* **4**, 3053 (1992).
- ²¹J. J. de Pablo, M. Laso, and U. W. Suter, *J. Chem. Phys.* **96**, 2395 (1992).
- ²²J. R. Errington, *J. Chem. Phys.* **118**, 9915 (2003).
- ²³Q. Yan and J. J. de Pablo, *J. Chem. Phys.* **111**, 9509 (1999).
- ²⁴F. A. Escobedo and F. J. Martinez-Veracoechea, *J. Chem. Phys.* **129**, 154107 (2008).