

Box length search algorithm for molecular simulation of systems containing periodic structures

A. J. Schultz, C. K. Hall, and J. Genzer

Department of Chemical Engineering, North Carolina State University, Raleigh, North Carolina 27695

(Received 18 February 2003; accepted 30 October 2003)

We have developed a box length search algorithm to efficiently find the appropriate box dimensions for constant-volume molecular simulation of periodic structures. The algorithm works by finding the box lengths that equalize the pressure in each direction while maintaining constant total volume. Maintaining the volume at a fixed value ensures that quantitative comparisons can be made between simulation and experimental, theoretical or other simulation results for systems that are incompressible or nearly incompressible. We test the algorithm on a system of phase-separated block copolymers that has a preferred box length in one dimension. We also describe and test a Monte Carlo algorithm that allows the box lengths to change while maintaining constant volume. We find that the box length search algorithm converges at least two orders of magnitude more quickly than the variable box length Monte Carlo method. Although the box length search algorithm is not ergodic, it successfully finds the box length that minimizes the free energy of the system. We verify this by examining the free energy as determined by the Monte Carlo simulation. © 2004 American Institute of Physics. [DOI: 10.1063/1.1636156]

I. INTRODUCTION

In molecular-level simulations it is sometimes necessary to allow the shape and volume of a simulation box to change in order to accommodate structures with periodic spacing. The standard way to accomplish this is to run the simulation at constant pressure or tension¹⁻⁶ as first described by Parrinello and Rahman. In this paper, we present a new way to accommodate periodic structures that involves finding the simulation box lengths that equalize the pressure in each direction, while maintaining the total volume at a constant value.

One example of a system that has periodic spacing is a microphase-separated block copolymer system. In order to quantitatively compare block copolymer simulation results with theoretical predictions, simulations need to be conducted at constant density. However, if a traditional constant pressure algorithm is used, the density cannot be set. Instead, preliminary simulations would be required to find the appropriate pressure to match the density, or a single pressure would have to be used, and the resulting variation in density (and hence the χ parameter) would complicate the analysis of the data.⁷ An alternative that is sometimes used is to run the simulations in a box that is large enough to accommodate multiple orientations of the structure and assume that the system will find the orientation with the correct periodic spacing. To solve this problem, we have developed an algorithm which finds the appropriate box lengths (those corresponding to the minimum in free energy) while maintaining constant density.

The algorithm operates by calculating the pressure in different directions and using that information to change the box lengths to more appropriate values. Additionally, the algorithm suppresses box length fluctuations by responding to the average rather than to the instantaneous pressure. The

end result is a system with box lengths that minimize the free energy and desired density. Although this algorithm does not properly sample phase space (because it responds to the average rather than to the instantaneous pressure) it does find the appropriate box length which can then be used in further NVT simulations at fixed box length.

We also describe a new Monte Carlo method to find the appropriate box lengths. This method essentially expands the standard NVT algorithm to include moves which change the box lengths while maintaining constant volume. Although this method samples phase space properly, it does not converge to the appropriate box lengths as quickly as the box length search algorithm. We use this Monte Carlo method to verify that the box length from the box length search algorithm corresponds to the minimum in the free energy.

II. METHODS

A. Thermodynamic basis for the algorithm

In order for the box length search algorithm to be successful, it must find the box lengths that minimize the free energy of the system. In this section, we derive an expression for the change in free energy when the box lengths are changed at constant volume, and use this expression to show that the algorithm will converge to the box lengths that minimize the free energy.

We start with the expression for the change in free energy at constant number of molecules, n , and temperature, T ,

$$\delta A = -P \delta V,$$

where A is the Helmholtz free energy, V is the volume, and P is the pressure. This equation only applies to a system in which the pressure is isotropic. If we reformulate this expression so that it becomes appropriate for a system undergoing a

volume change in only one direction, say the x direction, with the number of molecules and temperature fixed, we find

$$\delta A = -P_x L_y L_z \delta L_x,$$

where L_i and P_i are the box length and pressure in the i direction. This can be extended to the case in which all three box lengths change

$$\delta A = -P_x L_y L_z \delta L_x - P_y L_x L_z \delta L_y - P_z L_x L_y \delta L_z. \quad (1)$$

Next, consider the case of having all three box lengths vary, but keeping the total volume fixed as well as the number of molecules and the temperature. The box length changes in this case are then restricted by the equation

$$L_x L_y \delta L_z + L_x L_z \delta L_y + L_y L_z \delta L_x = 0$$

which can be solved for δL_z ,

$$\delta L_z = -\frac{L_x L_z \delta L_y + L_y L_z \delta L_x}{L_x L_y}. \quad (2)$$

Substituting Eq. (2) into Eq. (1) yields

$$\begin{aligned} \delta A &= -P_x L_y L_z \delta L_x - P_y L_x L_z \delta L_y + P_z (L_x L_z \delta L_y \\ &\quad + L_y L_z \delta L_x) \\ &= -(P_x - P_z) L_y L_z \delta L_x - (P_y - P_z) L_x L_z \delta L_y. \end{aligned}$$

Reformulating this expression in terms of reduced variables yields

$$\delta A^* = -Z \left[\left(\frac{P_x - P_z}{P} \right) \frac{\delta L_x}{L_x} + \left(\frac{P_y - P_z}{P} \right) \frac{\delta L_y}{L_y} \right], \quad (3)$$

where $A^* = A/nk_B T$, $Z = PV/nk_B T$, and $P = (P_x + P_y + P_z)/3$.

Based on Eq. (3), we can see that if the pressure in the x dimension is greater than the pressure in the z dimension, then increasing the box length in the x direction will decrease the free energy of the system. The same happens if the box length in the y direction is increased when the pressure in the y dimension is greater than the pressure in the z dimension. This is true regardless of the box length change in the z dimension necessary to maintain constant volume [notice that δL_z does not appear in Eq. (3)] so long as the box length changes are small. The ability to predetermine the sign of the free energy change for any given box length change provides the thermodynamic basis for the box length search algorithm.

B. Description of the box length search algorithm

The steps used in the algorithm to determine the box lengths that minimize the free energy of the system are the following. The first step is to run the simulation with fixed box lengths for a set length of time, Δt , and calculate the pressure in each direction (P_x , P_y , and P_z). We can use the virial theorem to calculate these pressures

$$P_k = \rho k_B T + \left\langle \sum_i \sum_{j>i} r_{ijk} f_{ijk} \right\rangle / V, \quad (4)$$

where ρ is the particle density of the system (N_p/V), and r_{ijk} and f_{ijk} are the k component of the distance and force between particle pair i and j .

The second step is to pick one dimension (we will choose z for consistency with the previous section), and to compare the pressure in that dimension, P_z , with the pressure in the other two dimensions. If P_z differs from the pressure in the x or y dimensions, then the box length in that dimension (x or y) is changed (increased or decreased) by ΔL_x or ΔL_y so that the free energy of the system will decrease, according to Eq. (3). Simultaneously, the box length in the z direction is changed to maintain constant volume. The coordinates of all the particles, r , are scaled with the box length change $r_{i,x} = r_{i,x} L'_x / L_x$, $r_{i,y} = r_{i,y} L'_y / L_y$, and $r_{i,z} = r_{i,z} L'_z / L_z$, where L'_k is the new box length and L_k is the old box length. If the particles have steep or discontinuous potentials, it may be necessary to perform the box length changes in small increments so that the system's potential energy does not increase too much. These two steps are performed repeatedly until each box length converges to a particular value.

The behavior of the algorithm can be adjusted by changing the algorithm parameter, Δt . If Δt is small, the algorithm responds quickly to pressure differences. If Δt is large, the algorithm responds more slowly, but the box lengths fluctuate less because the average in Eq. (4) contains more samples. The quickest way to find an appropriate value for Δt is to guess a trial value and run a short simulation. Once the box length search algorithm has determined an approximate value for the optimal box lengths, increase Δt in order to more precisely determine the optimal box lengths.

In order to suppress even more of the fluctuations, the box length search algorithm tracks the pressures from previous iterations and acts only if the pressure in the last iteration and the average pressure in the z direction during the past five iterations are both above or both below the pressure in the x and y directions. Additionally, if the pressure in the z direction from all five iterations is consistently above or below the pressure in the x and y directions, then the algorithm acts with twice the normal box length change ΔL . This increases the rate of convergence when the box length is far away from the optimal value. Although five iterations is, to a certain extent, arbitrary, it turned out to be the best compromise for our block copolymer simulations. The purpose of these iterations is to allow the algorithm to adjust to different noise levels without restarting the simulation. When the box lengths are far away from the optimal values, the algorithm can perform large box length changes at every iteration. When box lengths approach appropriate values, the algorithm falls back to using the average from previous iterations, which prevents the box lengths from wandering too far from the optimal values. The number of iterations over which to track the pressure can be adjusted to balance between these two modes; a small number of iterations favors the large box length change mode, while a large number of iterations favors the small box length change mode.

C. Variable box length Monte Carlo algorithm

In this section, we extend the standard canonical Monte Carlo algorithm so that it allows the box lengths to fluctuate while still sampling the NVT ensemble. We use this Monte Carlo algorithm to verify that the box length search algorithm described in the previous section finds the minimum in the free energy. In the standard NVT algorithm, reversible particle moves are proposed randomly and accepted with the standard Metropolis criterion,

$$P_{\text{acc}} = \min(1, \exp\{-\beta\Delta U\}). \quad (5)$$

In our modified algorithm, we also include reversible box length “moves” and scale the particle coordinates as in the box length search algorithm. For each box length move, the box lengths in two of the dimensions, for instance x and y , are changed by ΔL_x and ΔL_y , each of which is randomly chosen between $-\Delta L_{\text{max}}$ and ΔL_{max} . The box length in the third dimension (z) is changed so that the volume remains constant; these moves are accepted with the standard Metropolis criterion, Eq. (5), which is applicable because the total volume is conserved.

The algorithm has two parameters, the maximum box length change (ΔL_{max}) and the frequency with which box length changes are attempted. Just as in any Monte Carlo algorithm, these parameters can be adjusted to optimize the algorithm’s exploration of box length phase space.

III. RESULTS

A. Validation of the box length search algorithm

In this section, we validate the box length search algorithm by calculating the free energy versus the box length using the variable box length Monte Carlo algorithm. As a test to see if the free energy is a minimum at the box length obtained in the box length search algorithm, we consider a system of symmetric block copolymers that we have studied previously.⁸ This system orders into lamellar sheets, which are one-dimensional periodic structures. Here the only box dimension with a natural periodicity (and hence requiring a particular value for the box length) is the dimension perpendicular to the lamellar interface, L_x . The box lengths in the other directions can take any values that maintain constant volume. Therefore we restrict the box length search algorithm and the variable box length Monte Carlo algorithm so as to maintain the same box length in the two dimensions parallel to the interface, i.e., $L_y = L_x$ and $\Delta L_y = \Delta L_x$. Equation (3) then reduces to

$$\delta A^* = -2Z \left[\left(\frac{(P_x + P_y)/2 - P_z}{P} \right) \frac{\Delta L_x}{L_x} \right].$$

So now in our box length search algorithm, we only need to look at the average of the x and y pressures and compare that with P_z . For the variable box length Monte Carlo algorithm, we restrict the proposed moves such that $\Delta L_x = \Delta L_y$. Use of an algorithm that maintains constant volume while the temperature, volume fraction of each component, or chain length is varied, is critically important to making quantitative comparisons with experiment and/or theory. This is because experimental research on block copolymers focuses on melts or

on copolymers in solution at a fixed concentration, both of which are nearly incompressible.^{9–11} Theoretical calculations generally assume incompressibility^{12–16} and most simulation studies are performed on a lattice where the total volume must be fixed.^{17–20}

We use discontinuous molecular dynamics (DMD) (Ref. 21) to simulate the motion of the individual copolymer chains in both the box length search algorithm and the variable box length Monte Carlo algorithm. In the variable box length Monte Carlo algorithm, we run a standard NVT DMD simulation, but attempt box length changes at random time intervals during the DMD simulation. Such a hybrid Monte Carlo/molecular dynamics simulation is valid because both algorithms properly sample NVT phase space.

The systems used for both simulations contain 125 symmetric ($f=0.5$) block copolymers modeled as a flexible chain of 10 spheres, with component A spheres on one end and component B spheres on the other. All nonbonded copolymer beads interact with a hard sphere interaction when separated by a distance σ . Adjacent beads along the chain have a hard sphere interaction when separated by a distance 0.85σ and bounce back when they reach a separation of 1.15σ .²² Unlike beads interact with an additional square shoulder repulsive interaction of strength ϵ_{AB} when separated by distance 2σ . The simulations are conducted at $\chi N = 49$, where $\chi = \theta \epsilon_{AB} / kT$, θ is the average number of neighbors within the interaction distance, 2σ , and N is the copolymer chain length. The value of χN puts the system well above the order–disorder transition, which occurs at about $\chi N = 25$. Further details on the simulation of this copolymer system can be found in Ref. 8. For the simulations reported here, we always started the system with a lamellar structure oriented perpendicular to the z axis. Although large systems are capable of forming lamellae from disorder in less than 15 CPU hours, the resulting structure is almost always oriented diagonally within the simulation box and often has metastable defects. This diagonal orientation confuses the box length search algorithm and variable box length Monte Carlo because spacing in a diagonally oriented structure is less sensitive to changes in box length than in a perpendicularly oriented structure.

The box length search algorithm simulation was run for two different size systems, $n=125$ and $n=1000$, until they converged to the optimal box length in the z dimension, $L_z = 9.602\sigma$ for the $n=125$ system and $L_z = 19.222\sigma$ for the $n=1000$ system. (The $n=1000$ system contained two lamellar repeat units in the z direction.) The uncertainty in each of these measurements is approximately 0.002σ , which is the amplitude of the box length fluctuations that existed when the algorithm was stopped. The variable box length Monte Carlo algorithm was also run for 300 CPU hours for the $n=125$ system and 390 CPU hours for the $n=1000$ system to collect a histogram of the probability that the box length in the z dimension takes on any value, $L_{z,i}$. The free energy associated with box length $L_{z,i}$ can then be calculated from the Monte Carlo simulation using²³

$$A_i = A_0 - kT \ln \alpha_i, \quad (6)$$

where A_0 is an arbitrary parameter and α_i is the probability

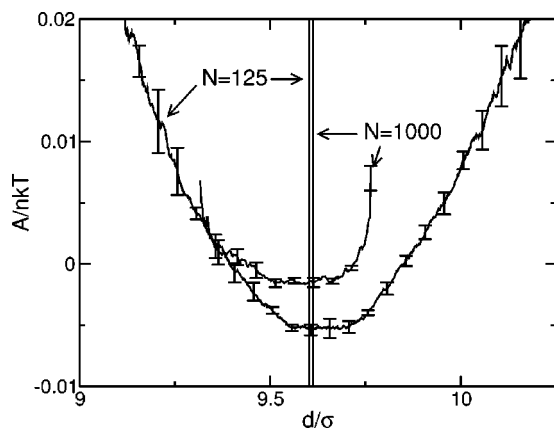


FIG. 1. Free energy as a function of box length from the variable box length Monte Carlo algorithm for system sizes $N=125$ and $N=1000$. Also shown is the free energy minimum as determined by the box length search algorithm.

that the box length is between $L_{z,i}-w/2$ and $L_{z,i}+w/2$, where w is the histogram bin width. We set $A_0=0$. For the system with $n=125$, we set $w/\sigma=0.001$ and for the $n=1000$ system, we set $w/\sigma=0.0025$. The results for A/nkT as a function of d/σ are plotted in Fig. 1, where d is the periodic spacing between lamellae. The error bars on selected points correspond to the standard deviation in the mean from three sections of the simulation. The uncertainty in the minimum location can be estimated by determining the box lengths (adjacent to the free energy minimum) where the free energy error bars both lie above those at the minimum. In this case, $9.50 < d/\sigma < 9.75$ for the small system and $9.45 < d/\sigma < 9.70$ for the large system. The minima in the $n=125$ system and in the $n=1000$ system at approximately $d/\sigma=9.6$ verify (within the error of the variable box length Monte Carlo algorithm) that the box length search algorithm “finds” the optimal box lengths.

B. Performance

It is of interest to compare the performance of the box length search algorithm and the variable box length Monte Carlo algorithm. To do this, we ran both types of simulations starting from various box lengths to test the rate of convergence of both methods to the optimal box length. Since both types of simulations with a small system size ($n=125$) converge to the box length relatively quickly. We focused our attention on the more challenging $n=1000$ system, which contains two lamellar repeat units.

The parameters for the box length search algorithm were chosen (in reduced units) to be $\Delta t^* \equiv \Delta t \sqrt{k_B T} / \sigma = 5$ and $\Delta L/L = 0.0001$. The Monte Carlo algorithm parameter $\Delta L_{\max}/\sigma$, the maximum box length change, was chosen to be equal to 0.002; the average number of reduced time units between box length change attempts was chosen to be 0.01. These Monte Carlo parameters maximize the box length fluctuations and hence the algorithm’s rate of exploration of phase space.

We plot $L_z/2$ versus CPU time for the $n=1000$ system from the box length search algorithm in Fig. 2(a) and from the variable box length Monte Carlo algorithm in Fig. 3.

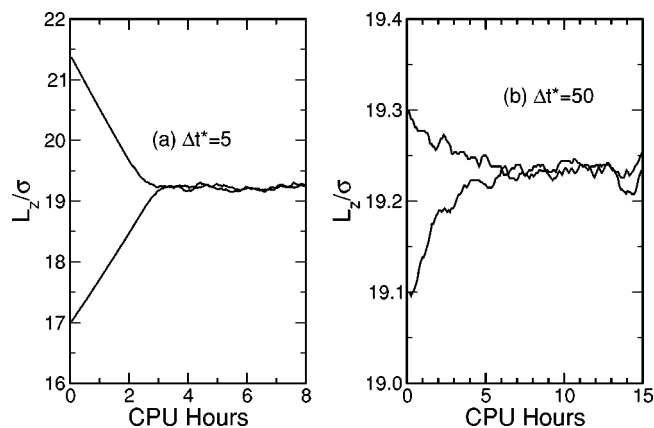


FIG. 2. Box length as a function of CPU hours using the box length search algorithm starting from different initial box lengths with (a) $\Delta t^*=5$ and (b) $\Delta t^*=50$.

Both simulations were started with box lengths of 17.0σ and 21.4σ . All simulations were performed on 666 MHz Alpha EV67 processors. The most apparent difference between the behavior of the box length search algorithm and the variable box length Monte Carlo algorithm is the absence of fluctuations in the box length search algorithm. The fluctuations about the equilibrium value in the Monte Carlo algorithm are substantial ($\sim 0.3\sigma$) and slow. In the box length search algorithm, the fluctuations are smaller ($\sim 0.1\sigma$) and quick. Since the fluctuations in the box length search algorithm do not obey any thermodynamic norm, the only reliable conclusion is that the minimum free energy box length lies somewhere between the maximum and minimum box length value. In order to more precisely determine the optimal value for the box length we ran a second set of simulations starting at 19.1σ and 19.3σ with $\Delta t^*=50$. With the increased value for Δt^* , the algorithm suppresses even more of the fluctuations, reducing the uncertainty (the amplitude of the fluctuations) to 0.02σ .

Although the Monte Carlo algorithm converges to the equilibrium value, the amplitude and slowness of the fluctuations hamper the determination of the precise value of the free energy minimum. The Monte Carlo results in Fig. 1

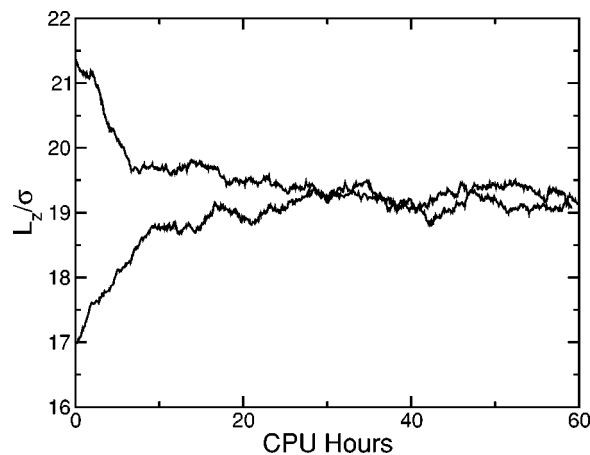


FIG. 3. Box length as a function of CPU hours using variable box length Monte Carlo algorithm starting from different initial box lengths.

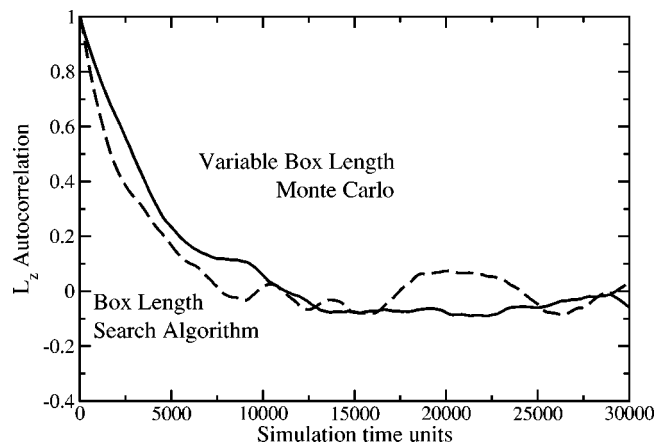


FIG. 4. Autocorrelation function of the box length for the box length search and the variable box length Monte Carlo algorithms.

illustrate this drawback. Although the Monte Carlo algorithm simulation was run for 390 CPU hours for the larger system, the algorithm determined the appropriate box length with less precision ($9.49 < d/\sigma < 9.69$) than the box length search algorithm achieved in 3 h ($9.55 < d/\sigma < 9.65$). Using a traditional Monte Carlo constant pressure simulation in this case would require additional time to determine the appropriate pressure and then convergence would be even slower than the variable box length Monte Carlo algorithm because of the volume fluctuations.

In order to better quantify the performance of the box length search algorithm and the variable box length Monte Carlo algorithm, we calculated the autocorrelation function of the box length in the z dimension while each method was being used. We have plotted these autocorrelation functions versus CPU hours in Fig. 4. The box length in the box length search algorithm appears to “forget” its old value slightly faster than the variable box length Monte Carlo algorithm, (although this difference is on the order of the uncertainty in the autocorrelation itself). This result might at first seem surprising since the box length search algorithm was over 100 times faster than the variable box length Monte Carlo algorithm. Apparently, the box length search algorithm explores a much narrower range of phase space, so that during the time it takes the algorithm to “forget” its previous value, it is able to get a much better estimate of the optimal box length than the Monte Carlo algorithm.

Another autocorrelation function that can help to elucidate the box length search algorithm’s superior performance compared to the variable box length Monte Carlo algorithm is the autocorrelation function of the difference between the pressure in the z dimension and the average of the pressures in the x and y directions. This is useful because the box length search algorithm’s performance depends directly on how quickly these pressure fluctuations die out. The autocorrelation functions for $P_z - (P_x + P_y)/2$ and the autocorrelation function for the total pressure, $P = (P_x + P_y + P_z)/3$ were obtained by running the simulation at constant volume and averaging each pressure sample over 50 reduced time units. The results are shown in Fig. 5. From this plot, we can see that the pressure difference is nearly decorrelated within

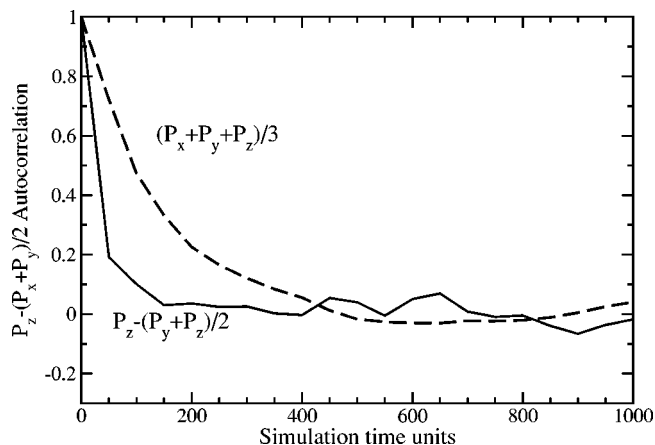


FIG. 5. Autocorrelation function of the difference between the pressure in the z dimension and the average pressure in the x and y dimensions and the autocorrelation function for the total pressure.

50 reduced time units, which is much faster than the decorrelation time for the box length seen in Fig. 4. It is also faster than the decorrelation of the total pressure, indicating that the pressure in each dimension fluctuates cooperatively. This cooperative fluctuation means the box length in the box length search algorithm will fluctuate less than in an algorithm that depends on the total pressure, such as a constant pressure Monte Carlo algorithm. It is the pressure difference’s fast decorrelation time that leads to the fast performance of the box length search algorithm.

Although the performance measurements given thus far are based on simulations of chains of length 10, one might wonder if the performance difference between the algorithms would diminish for systems with longer chains. The relaxation time of these longer chains might be expected to dominate any dynamics in the system and cancel advantages that were seen with the box length search algorithm for short chain systems. To test this, we performed additional simulations on a system containing 640 chains of length 40. The temperature was adjusted to maintain a constant value of χN at 49. The simulations were performed with the parameters used in the previous simulations: $\Delta t^* = 50$ and $\Delta L/L = 0.0001$ for the box length search algorithm and $\Delta L = 0.002\sigma$ for the variable box length Monte Carlo algorithm. The average time between MC box length change attempts was maintained at 0.01. We plot $L_z/2$ versus CPU time in Fig. 6 for both algorithms starting with the periodic spacing at 36.5, which is somewhat lower than the optimal value of 36.9. The box length search algorithm takes about 35 CPU hours to arrive at the optimal value of 36.9 and then fluctuates with an amplitude of about 0.2. The box length in the variable box length Monte Carlo simulation fluctuates between 36.2 and 37.4 for the duration of the simulation. Even after 600 h, we were unable to determine the optimal box length based on the Monte Carlo data. We also tested the longer chain system starting with the box length at 32.5, which is much lower than the optimal value. The box length versus CPU hours for this run is plotted in Fig. 7 for both algorithms. The box length search algorithm continues to maintain a substantial advantage over the variable box length

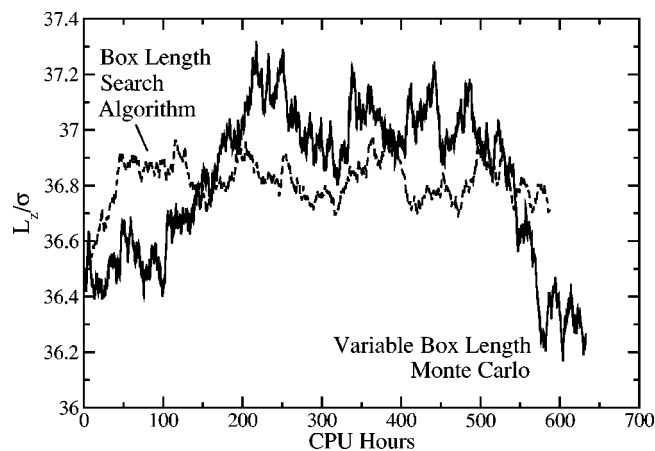


FIG. 6. Box length as a function of CPU hours using the box length search and variable box length Monte Carlo algorithms for a system of 640 chains of length 40. The box length starts at a 36.5σ , which is slightly less than the optimal value.

Monte Carlo algorithm, converging about 10 times more quickly than the Monte Carlo algorithm.

Although we cannot make a quantitative comparison between the performance of the two algorithms, we can say that the box length search algorithm continues to perform much better than the variable box length Monte Carlo algorithm, even for longer chains.

IV. SUMMARY AND DISCUSSION

We have developed a box length search algorithm that converges to the box lengths at which the free energy of the system is at a minimum while maintaining constant volume. Such an algorithm is useful for finding appropriate box lengths for periodic structures for further NVT simulations. We have also developed a Monte Carlo algorithm that explores NVT phase space while allowing the box lengths to fluctuate. We have compared the two algorithms using a block copolymer lamellar structure and verified that the box length search algorithm converges to the box length with the

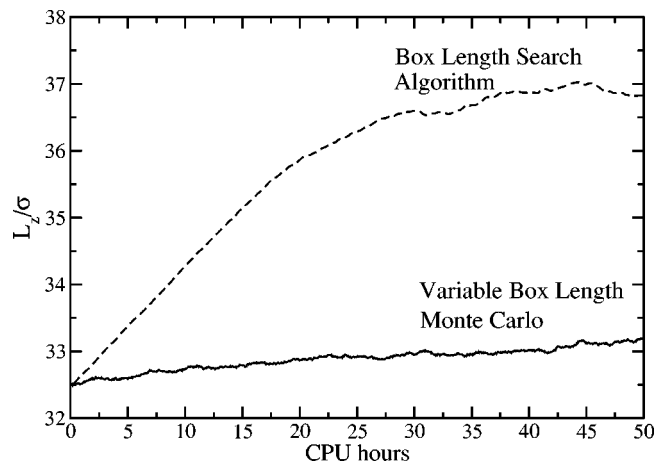


FIG. 7. Box length as a function of CPU hours using the box length search and variable box length Monte Carlo algorithms for a system of 640 chains of length 40. The box length starts at a 32.5σ , which is much less than the optimal value.

lowest free energy as determined from the Monte Carlo simulation. We also found that the box length search algorithm converges more quickly to the optimal box length than the Monte Carlo algorithm because the amplitude of the fluctuations in the box length is reduced.

Although the box length search algorithm performed well in the cases described here, there are some more complex cases that the algorithm cannot handle well. For instance, if the free energy is insensitive to the box length, then the algorithm assumes that the minimum has been reached and suppresses fluctuations that would move the box length out of the area of insensitivity. If the free energy contains multiple minima, the algorithm can get stuck in a metastable state. In such a case, the variable box length Monte Carlo algorithm could be used to properly simulate the system. A more direct approach that maintains the spirit of the box length search algorithm would be to sample P_x , P_y , and P_z as a function of the box lengths and then plot the difference in pressure between the dimensions ($P_z - P_x$ and $P_z - P_y$) to determine the box lengths at which the pressure differences are zero. This method avoids any convergence problem with the box length search algorithm but would still converge more rapidly than the variable box length Monte Carlo algorithm. However, even the Monte Carlo sampling technique would be inadequate for many systems with multiple minima. The lamellar system investigated in this paper does contain multiple minima corresponding to identical periodic spacing but different lamellar orientations or number of repeat units. Neither algorithm was able to sample these different orientations. If sampling multiple minima is truly important, the most robust approach would be to start multiple simulations with different initial box lengths and allow either one of the algorithms to find the free energy minima of the system. Finally, we should point out that the box length search algorithm's inability to locate the global minimum in some systems with multiple free energy minima does not detract from its applicability to cases without multiple free energy minima.

ACKNOWLEDGMENTS

This work was supported by the GAANN Computational Sciences Fellowship of the U. S. Department of Education and the Office of Energy Research, Basic Sciences, Chemical Science Division of the U. S. Department of Energy under Contract No. DE-FG05-91ER14181. Acknowledgment is made to the Donors of the Petroleum Research Fund administered by the American Chemical Society for partial support of this work.

- ¹M. Parrinello and A. Rahman, *Phys. Rev. Lett.* **45**, 1196 (1980).
- ²M. Parrinello and A. Rahman, *J. Appl. Phys.* **52**, 7182 (1981).
- ³R. Najafabadi and S. Yip, *Acta Metall.* **17**, 1199 (1983).
- ⁴M. Parrinello and A. Rahman, *J. Chem. Phys.* **80**, 860 (1984).
- ⁵C. L. Cleveland and U. Landman, *J. Chem. Phys.* **94**, 7376 (1991).
- ⁶C. L. Cleveland, *J. Chem. Phys.* **89**, 4987 (1988).
- ⁷M. Murat, G. S. Grest, and K. Kremer, *Macromolecules* **32**, 595 (1999).
- ⁸A. J. Schultz, C. K. Hall, and J. Genzer, *J. Chem. Phys.* **117**, 10329 (2002).
- ⁹T. P. Lodge, M. W. Hamersky, K. J. Hanley, and C.-I. Huang, *Macromolecules* **30**, 6139 (1997).

- ¹⁰N. Sakamoto *et al.*, *Macromolecules* **30**, 5321 (1997).
- ¹¹C. Lai, W. B. Russel, and R. A. Register, *Macromolecules* **35**, 4044 (2002).
- ¹²L. Leibler, *Macromolecules* **13**, 1602 (1980).
- ¹³G. H. Fredrickson and E. Helfand, *J. Chem. Phys.* **87**, 697 (1987).
- ¹⁴M. W. Matsen and F. S. Bates, *J. Chem. Phys.* **106**, 2436 (1997).
- ¹⁵M. D. Whitmore and J. Noolandi, *J. Chem. Phys.* **93**, 2946 (1990).
- ¹⁶M. W. Matsen, *J. Phys.: Condens. Matter* **13**, 0 (2001).
- ¹⁷R. G. Larson, *Macromolecules* **27**, 4198 (1994).
- ¹⁸A. Hoffmann, J. Sommer, and A. Blumen, *J. Chem. Phys.* **106**, 6709 (1997).
- ¹⁹G. Besold, O. Hassager, and O. G. Mouritsen, *Comput. Phys. Commun.* **122**, 542 (1999).
- ²⁰H. Fried and K. Binder, *J. Chem. Phys.* **94**, 8349 (1991).
- ²¹S. W. Smith, B. D. Freeman, and C. K. Hall, *J. Comput. Phys.* **134**, 16 (1997).
- ²²A. Bellemans, J. Orban, and D. V. Belle, *Mol. Phys.* **39**, 781 (1980).
- ²³D. Frenkel and B. Smit, *Understanding Molecular Simulation* (Academic, San Diego, 1996).