

Parallel Gibbs-ensemble simulations

By L. D. J. C. LOYENS, B. SMIT and K. ESSELINK

Shell Research B.V., Koninklijke/Shell-Laboratorium, Amsterdam,
PO Box 38000, 1030 BN Amsterdam, The Netherlands

(Received 6 January 1995; accepted 26 April 1995)

A truly parallel algorithm for the Gibbs-ensemble simulation technique is given. The key components of this algorithm are a parallel displacement move based on the hybrid Monte Carlo method and a parallel exchange move based on a novel algorithm. For the parallel exchange move each processor generates additional trial conformations, and a trial conformation is selected with the most favourable energy. This introduces a bias, which is removed by a modification of the acceptance rules. A proof of the correctness of the parallel exchange move is given and the algorithm is verified for a Lennard-Jones system. Simulations on ten processors of an SP/2 parallel computer give a speedup of eight for large systems and a speedup of four for systems of typical size.

1. Introduction

In the early days of computer simulations, the determination of the phase diagram of a model fluid was already an important application of molecular simulation techniques. Before the development of the Gibbs-ensemble technique [1] such a calculation used to be an elaborate task which required many simulations [2]. In a Gibbs-ensemble simulation vapour–liquid or liquid–liquid phase coexistence can be determined from a single simulation [1, 3]. This technique has already been applied to systems containing particles ranging from simple Lennard-Jones [1] to *n*-alkanes as long as C_{48} [4, 5] (see [6] for an overview of the systems that have been studied).

At present most, if not all, of the Gibbs-ensemble simulations are performed on a sequential computer. At high (liquid) densities or for long chain molecules such a calculation may require significant amounts of CPU time. For these systems one would like to take advantage of the use of parallel computers. Because of the sequential nature of the Monte Carlo method—one can only perform the next move once the previous one has been accepted or rejected—parallelization of a Gibbs-ensemble algorithm is not as straightforward as, for example, a molecular dynamics algorithm. A simple way of performing a parallel Monte Carlo simulation is to distribute Q simulations over the Q processors, each with its own sequence of random numbers. It is important to note that this simple parallelization is very efficient provided that one can start with an equilibrated initial configuration. At high densities or for long chain molecules, however, the equilibration may require more CPU time than the actual simulation. For these types of applications it is therefore essential to have a ‘truly’ parallel algorithm.

In this article, we describe the development of a parallel Gibbs-ensemble algorithm. A Gibbs-ensemble algorithm consist of three types of Monte Carlo moves: particle displacement, volume changes, and particle exchanges. For each of these three types of moves a parallel algorithm has been developed. For the particle displacement and volume change steps, we have based our algorithm on parallel algorithms

available in the literature. For the particle exchange step, we have developed a novel type of algorithm based on our parallel Monte Carlo simulation technique [7].

2. Sequential algorithm

In the Gibbs-ensemble scheme simulations of the two coexisting phases (liquid–vapour or liquid–liquid) are carried out in parallel in two separate simulation boxes. The boxes are kept at a constant temperature T , the total volume of the two boxes is fixed at V , and a fixed number of N particles are distributed over the two boxes. Monte Carlo rules, which allow for changes in the number of particles and the volume, ensure that the two boxes are in thermodynamic equilibrium with each other. Since the two boxes are not in ‘physical contact’, there is no interface and the bulk properties of the two coexisting phases can be obtained directly from a single simulation.

The probability of finding a particular configuration in the Gibbs ensemble is given by [3, 6, 8]:

$$\mathcal{N}(N, V, \beta; n_1, V_1, \xi^N) dV_1 d\xi^N \quad (1)$$

$$\begin{aligned} &\propto \frac{V_1^{n_1} (V - V_1)^{N - n_1}}{(N - n_1)! n_1!} \exp[-\beta U_1(n_1)] \exp[-\beta U_2(N - n_1)] dV_1 d\xi^N \\ &= \exp(-\beta W_B) dV_1 d\xi^N. \end{aligned} \quad (2)$$

Here, $\beta = 1/k_B T$, n_1 denotes the number of particles in box 1, V_1 the volume of box 1, ξ^N denotes the positions of the particles scaled with respect to the box length, and $U_i(n_i)$ is the intermolecular potential in box i with n_i particles.

The exponent in equation (2) is called the pseudo-Boltzmann factor, and an inspection of it shows that a new configuration n can be generated by applying one of the following three moves to the old configuration o :

- (a) *displacement*; displace a randomly selected particle in one of the boxes,
- (b) *volume change*; change the volume of one of the boxes while keeping the total volume constant,
- (c) *particle exchange*; exchange particles between the two boxes.

The transition of configuration o to configuration n is accepted with probability:

$$\text{acc}(o \rightarrow n) = \min[1, \exp(-\beta \Delta W_B)], \quad (3)$$

where $\Delta W_B = W_B(n) - W_B(o)$. A simulation is set up by selecting one type of move at random with a specified frequency.

3. Parallel algorithm

An efficient parallel Gibbs-ensemble simulation requires the parallelization of each type of move. We discuss the parallelization of each move here.

3.1. Parallel displacement

In a sequential Gibbs-ensemble simulation one usually displaces only *one* particle at the time. This clearly is a bottleneck in a parallel Gibbs-ensemble simulation.

There are several methods to circumvent this bottleneck. The first one is to displace two or more particles at the time and the second one is based on the molecular dynamics technique.

If the range of the interactions is very short, the system can be divided in independent regions. In such a region the displacement of a particle does not affect the energy of the particles in another region. In each of these regions one particle can be displaced simultaneously, and the calculation is distributed over the processors. Such a scheme, however, fails for systems with long range interactions or chain molecules.

Another method is to displace simultaneously *all* of the particles as is done in molecular dynamics [2]. For a Gibbs-ensemble simulation we need to maintain a constant temperature by, for example, using a Nosé–Hoover thermostat [9, 10]. Recently, Mehlig *et al.* [11] applied the hybrid Monte Carlo scheme of Duane *et al* [12] to condensed-matter systems consisting of Lennard-Jones particles. In the hybrid Monte Carlo method random initial momenta are drawn from a Gaussian distribution and molecular dynamics is used to obtain trial configurations. The acceptance rules are chosen such that the simulations are performed at constant temperature. Mehlig *et al* [11] argue that the time step that can be used in the hybrid Monte Carlo scheme is larger than the one used in an ordinary molecular dynamics scheme.

More specifically, a hybrid Monte Carlo move from configuration o to configuration n in a randomly selected box, say box 1, consists of the following steps:

- (1) For each particle in configuration o a random velocity is drawn from a Gaussian distribution with mean 0 and width $(1/\beta)^{1/2}$. Here, we assume that all particles have the same unit mass.
- (2) A trajectory of the particles is generated during $\delta t = N_{MD}\Delta t$ by taking N_{MD} molecular dynamics time steps, each with integration step Δt , ending in configuration n .
- (3) The pseudo-Boltzmann factor

$$\exp(-\beta\Delta W_B) = \exp(-\beta\Delta U) = \exp\{-\beta[U(n) - U(o)]\} \quad (4)$$

is computed, where in the hybrid Monte Carlo scheme ΔU is called the discretization error. The new configuration is accepted with a probability given in equation (3).

Clearly, step (1) is independent for each particle; the velocities can be drawn from a Gaussian distribution in parallel. Step (2) and the energy calculation in step (3) can also be done in parallel by using a parallel molecular dynamics algorithm. Only the acceptance/rejection criterion will involve a serial step. We note that the discretization scheme used within molecular dynamics, for example the leap-frog scheme, should satisfy detailed balance.

We use the parallel hybrid Monte Carlo as the parallel displacement move within the Gibbs-ensemble simulations. The parallel molecular dynamics algorithm is based on geometric decomposition of the simulation universe [13]. In this scheme we decompose each box in the x and y direction. Each processor is responsible for an $x - y$ column of a box along the z direction. These columns all have the same size, and the particles inside a column reside on the processor to which the column is assigned. Neighbouring columns are assigned to neighbouring processors, so that logically the processors are connected in a two-dimensional torus. This method is very

well suited for short-range potentials; for results obtained with this method in the context of ordinary molecular dynamics we refer the reader to [13].

3.2. Parallel volume change

A volume change ΔV of box 1 goes together with a corresponding volume change $-\Delta V$ in box 2. For a parallel volume change move this means that we have to adapt the size of each column in each box accordingly, and the particle positions inside a column are scaled. Since there is a one-to-one correspondence between columns and processors, all of these computations can be computed in parallel. The calculation of the energy can be done as for parallel molecular dynamics.

3.3. Particle exchange

In a *sequential* Gibbs-ensemble simulation the acceptance ratio of an exchange, i.e., the number of successful moves divided by the number of trial moves, may become prohibitively low in the case of high fluid densities and/or for chain molecules. In particular, for this move one would like to take advantage of parallel computing. The conventional algorithm has only limited possibilities for parallelization; in fact the only possibility is to distribute the calculation of the energy of the particle that is exchanged. The parallelization of the energy calculation for a single particle, however, scales very poorly and therefore does not result in a significant speedup. A more attractive route has been explored [7] where the parallel insertion of chain molecules in a single box is discussed. Several trial conformations are generated in parallel and the most favourable is selected with the highest probability. This procedure, however, generates a bias that should be removed by an appropriate acceptance rule. For a parallel Gibbs-ensemble simulation this implies that one still tries only *one* particle exchange at a time, using the parallel algorithm to increase the acceptance ratio of the exchange.

We now describe the parallel algorithm of the exchange move. Assuming that we are moving a particle from box 1 to box 2, the old configuration with n_1 particles in box 1 with volume V_1 is denoted by o and the new configuration ($n_1 - 1$ particles in box 1 with volume V_1) by n (see figure 1). Furthermore, we assume that we have a total of Q processors.

- (1) One particle to be moved from box 1 to box 2 is picked randomly and its potential energy $u(o)$ is computed.
- (2) N_{try} positions $\xi_{j,q}(n)$, $j = 1, 2, \dots, N_{\text{try}}$ are picked randomly in box 2 for each processor $q = 1, 2, \dots, Q$. For all of these trial positions the potential energies $u_{j,q}(n)$ are computed and the factor $Z(n)$ defined by

$$Z(n) \equiv \sum_{j,q}^{Q \times N_{\text{try}}} \exp[-\beta u_{j,q}(n)], \quad (5)$$

is then computed globally. A trial position, say $\xi(n)$, is picked with a probability

$$P(n) = \frac{\exp[-\beta u(n)]}{Z(n)}. \quad (6)$$

Note that this selection criterion favours a trial position with a low energy.

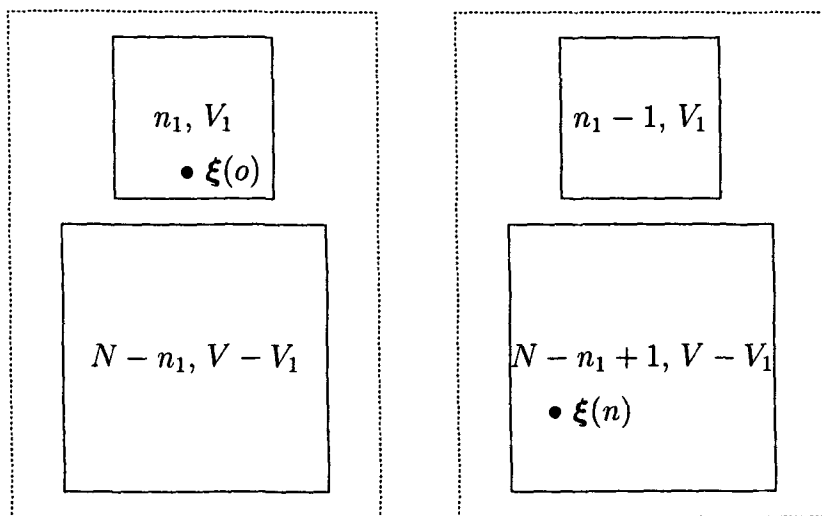


Figure 1. A schematic representation of the parallel exchange move in which the boxes of the old configuration o (on the left) and the new configuration n (on the right) are depicted. The particle to be removed from box 1 has a position $\xi(o)$, and the particle to be inserted in box 2 has a position $\xi(n)$.

- (3) In box 1, the box of the old configuration, N_{try} positions $\xi_{j,q}(o)$ are picked randomly by each processor q , but for one processor $N_{\text{try}} - 1$ positions are picked, so that the total number of positions chosen is $Q \times N_{\text{try}} - 1$. The potential energies $u_{j,q}(o)$ for these positions are computed and the quantity $Z(o)$ defined by

$$Z(o) \equiv \exp[-\beta u(o)] + \sum_{j,q}^{Q \times N_{\text{try}} - 1} \exp[-\beta u_{j,q}(o)] \quad (7)$$

is determined globally.

- (4) The step $o \rightarrow n$ is accepted with a probability

$$\text{acc}_{\text{exch}}(o \rightarrow n) = \min(1, P_{\text{exch}}), \quad (8)$$

$$P_{\text{exch}} = \exp(-\beta \Delta W_B) \frac{\exp[-\beta u(o)] Z(n)}{\exp[-\beta u(n)] Z(o)}. \quad (9)$$

In Appendix A we prove that this algorithm indeed samples the desired distribution. Note that in the case of $N_{\text{try}} = 1$ and one processor $Q = 1$, the acceptance rule given by equation (9) is the same as the acceptance rule used in a sequential Gibbs-ensemble simulation. For large systems and a large number of processors $P_{\text{exch}} \approx 1$. This is shown in Appendix B.

The parallel Gibbs-ensemble algorithm was implemented on an IBM 9076 scalable POWER-parallel system 2 (SP/2) comprising 30 powerful workstations (processors). Each processor is connected to a fast switch used for communication. Only ten processors were available for parallel processing.

4. Results and discussion

To test the performance of our algorithm we have calculated the vapour–liquid coexistence curve of the Lennard-Jones fluid. The phase diagrams of the Lennard-Jones fluid is well characterised from sequential Gibbs-ensemble simulations [1, 14, 15, 16]. The Lennard-Jones potential is an example of a typical short-range interaction potential of the form

$$\phi(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right], \quad (10)$$

where r is the distance between two particles. This potential is usually truncated at distance R^c , and for the molecular dynamics it is convenient to shift the potential as well, i.e.,

$$\Phi(r) = \begin{cases} \phi(r) - \phi(R^c), & \text{if } r \leq R^c, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

The cut-off radius was set to $R^c = 2.5\sigma$. The phase diagram depends on the way the potential is truncated [16], and therefore we cannot compare our results with the data for the full Lennard-Jones potential (see [16] for details).

4.1. Verification

To test whether the parallel algorithm indeed gives identical results to the usual sequential method, we have compared the results obtained from a sequential simulation with those obtained from a parallel simulation. These simulations were done with 432 ($2 \times 6 \times 6 \times 6$) Lennard-Jones particles distributed evenly over two boxes with initial dimensions $(8.82\sigma)^3$. The particle positions were initialized on the corner points of a cubic lattice. The simulations were done in cycles, and in each cycle we performed 30 Monte Carlo moves that were selected with a prescribed probability: 80% particle exchanges, 16% hybrid Monte Carlo moves, and 4% volume changes. In a hybrid Monte Carlo move the equations of motion were integrated for ten time steps, the time step is selected such that 50% of the moves are accepted. The maximal volume change was set to such a value that 50% of the attempts were accepted. For this test, the parallel simulations were performed on an SP/2 using four processors ($Q = 4$, $N_{\text{try}} = 1$). During the exchange step each processor generated one trial position, which increases the number of attempts to exchange a particle by a factor of four. A total simulation consisted of 11 000 cycles and the averaging was done over the last 5000 cycles.

In figure 2 we compare the results obtained from a sequential simulation with those obtained from a parallel simulation. Figure 2 shows that indeed the parallel simulations and sequential simulations yield identical results.

4.2. Performance of the hybrid Monte Carlo move

Here, we report on the use of the hybrid Monte Carlo scheme as the displacement move in the parallel Gibbs-ensemble algorithm. During the equilibration period we adjusted the time step used in the hybrid Monte Carlo moves to give a 50% acceptance of the attempts. This time step depends on the density as shown in table 1; for comparison we have added the time step that is used in a conventional

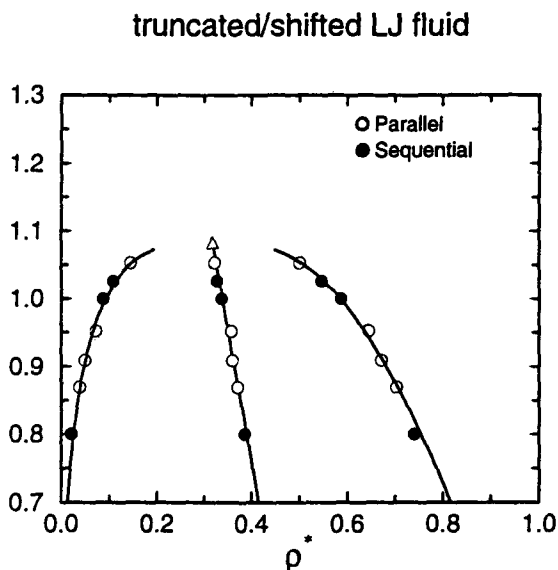


Figure 2. Phase diagram of the three-dimensional Lennard-Jones fluid obtained from parallel simulations. The circles are the reduced densities of the vapour and liquid phases obtained from the parallel simulations for various reduced temperatures. The full lines represent fits to the scaling law and rectilinear law (middle line) [17]. The triangle is the estimate of the critical point for the parallel simulations, $T_{\text{crit}}^* \cong 1.1$. As a reference, the full dots represent some results of the sequential simulations.

Table 1. Average molecular dynamics time steps used in the hybrid Monte Carlo scheme for each phase. After each run these time steps were dynamically changed in order to obtain a 50% acceptance probability for the parallel displacement move. For comparison we have included the time step that is used in a conventional molecular dynamics simulation [2]. For $T^* = 0.7$ we performed a simulation of a larger system on $Q = 10$ processors. All other (reduced) temperatures are for simulations performed on $Q = 4$ processors.

T^*	N	Δt_{vapour}	Δt_{liquid}
0.700	14 364	0.017	0.000 31
0.870	432	0.0210	0.002 24
0.909	432	0.0182	0.002 23
0.952	432	0.0139	0.002 35
1.053	432	0.0084	0.002 87
MD	432	0.002	0.002

molecular dynamics simulation using the leap-frog scheme. We found that the time step is significantly larger compared to molecular dynamics only for the very dilute vapour phase. For the liquid phase we found a time step that is about the same as the one used in molecular dynamics. We should remark that we did not optimize the number of time steps (N_{MD}) before the acceptance test was performed; instead, we used $N_{\text{MD}} = 10$ as is suggested in [11]. The findings of Glaser [18], however, indicate that for efficient sampling a much larger value of N_{MD} of at least 100 is necessary.

In another test, we scaled up the two simulation boxes containing 432 Lennard-Jones particles by a factor of three in each dimension while keeping the density constant. This gives a system of 14 364 Lennard-Jones particles in a $(682.13\sigma)^3$ box. This system was followed for 75 000 cycles at a low reduced temperature of 0.7, just above the triple point. In each cycle we performed 105 Monte Carlo moves that were selected with a prescribed probability: 95% particle exchanges, 4% hybrid Monte Carlo moves, and 1% volume changes. The time steps used in hybrid Monte Carlo and the maximal volume displacement were adjusted dynamically as described before. For this test, the parallel simulations were performed on an SP/2 using ten processors ($Q = 10$, $N_{\text{try}} = 1$). Averaging was performed over the last 15 000 cycles. For this system, the number of successful exchanges per 1000 cycles was on average 224 out of 100 026 attempts. The results are also given in table 1.

These observations suggest that, at least for a Gibbs-ensemble simulation, it is advantageous to use hybrid Monte Carlo for the vapour phase. For large systems and/or for the liquid phase, hybrid Monte Carlo exhibits a poor performance, and it is more efficient to use an ordinary constant temperature molecular dynamics algorithm. Our results are in contrast with the results of Brodz and de Pablo [19] for a simulation of silica who conclude that the time step in a hybrid Monte Carlo simulation is twice as large as that for a comparable molecular dynamics simulation.

4.3. Scalability

From the point of view of parallel computing it is interesting to investigate the efficiency of our algorithm. The hybrid Monte Carlo scheme and the total energy calculation in the volume step scale very well with the number of processors. The novel aspect of this work is the particle exchange step. In order to address the scalability, we benchmarked our parallel Gibbs-ensemble algorithm for different numbers of processors under as similar conditions as possible. We followed the large 14 364 particles Lennard-Jones system for 3000 cycles on $Q = 1, 2, 4, 6, 8, 10$ processors while keeping the total number of exchange attempts per cycle constant at 1000. For example, on one processor we tried 1000 exchange attempts, on four processors we tried 250 exchange attempts per processor, etc.

For these benchmarks we measured the wall-clock time required to complete the simulation. This may lead to small deviations in the timing results, but this way the results we present are of more practical interest.

The statistics are gathered in table 2. From this table, we see that the scaling behaviour for the parallel Gibbs-ensemble algorithm is roughly linear with the number of processors, although for $Q = 10$ processors the decrease in run-time is less than expected. On the basis of the timing results, the highest speedup obtained is 7.8 for a system containing 14 364 particles. In practice Lennard-Jones simulations are usually performed for much smaller systems. For such a system containing 1024 particles we obtained a speedup of 4.2 on $Q = 10$ processors and a speedup of 3.1 on $Q = 4$ processors. It is instructive to see how the acceptance probabilities (in percent) of the parallel exchange move depend on the number of processors (see table 2). Clearly, a linear scaling behaviour of the exchange probability with the number of processors is observed (0.71% on one processor, 2.63% on four processors, and 6.31% on ten processors). This means that one can effectively use our parallel exchange algorithm when the acceptance probability for the exchange move in a sequential simulation becomes too low.

Table 2. Statistics of the exchange move for various numbers of processors and system sizes. The column with the heading 'Accepted' gives the number of exchange moves accepted. The column with the heading $\langle P_{\text{exch}} \rangle$ gives the percentage acceptance probability of the exchange move. The measured wall-clock time reported is the total simulation time in hours. The first six entries give the statistics of a large 14 364 particle simulation. The last four entries give the statistics of a smaller 1024 particle simulation.

N	Q	N_{exchange}	Accepted	$\langle P_{\text{exch}} \rangle$	Time (h)
14 364	1	3 000 008	21 425	0.71	62.7
14 364	2	1 500 070	20 450	1.36	28.8
14 364	4	750 033	19 750	2.63	13.7
14 364	6	497 962	18 375	3.69	9.9
14 364	8	371 992	17 575	4.72	8.2
14 364	10	300 047	18 925	6.31	8.0
1 024	1	2 999 981	23 478	0.78	5.2
1 024	4	750 006	15 518	2.07	1.7
1 024	10	299 806	16 152	5.39	1.2

5. Concluding remarks

In this work we have presented a (truly) parallel Gibbs-ensemble algorithm. We call this algorithm truly parallel because it has the proper scaling behaviour with respect to number of particles and CPU time. Whereas the volume and displacement steps in a Gibbs-ensemble simulation can be parallelized using available algorithms, the exchange step required the development of a novel algorithm. In this algorithm attempts to insert a particle are performed in parallel, and the trial conformation with the most favourable energy is selected with the highest probability for insertion. The bias that is introduced by this selection step is removed exactly by a modification of the acceptance rules.

We have implemented a parallel Gibbs-ensemble algorithm on an SP/2 parallel computer to calculate the vapour-liquid equilibrium curve of the Lennard-Jones fluid. Close to the triple point for a large system of 14 364 particles we obtained a speedup of eight in the case of ten processors.

At this point we want to emphasize that our parallel Gibbs-ensemble algorithm is by no means limited to Lennard-Jones particles. On the contrary, we expect the method to be more advantageous for Gibbs-ensemble simulation of systems containing chain molecules which have much lower acceptance probabilities for the exchange move compared to atomic systems. In addition, these simulations of chain molecules are usually performed for a significantly larger number of particles compared to the usual simulations of Lennard-Jones systems [5]. For these large systems, the use of a parallel Gibbs-ensemble algorithm of the type presented here is particularly advantageous.

We thank Matt Glaser for illuminating discussions about the performance of the hybrid Monte Carlo method.

Appendix A

Here we prove that the use of the parallel exchange rule defined in equations (8) and (9) generates configurations that are distributed according to a

pseudo-Boltzmann distribution, i.e., the probability of finding a configuration is given according to equation (2).

The standard technique is based on demonstrating that detailed balance is obeyed, i.e., the flow of configurations going from o to n should be equal to the reverse:

$$K(o \rightarrow n) = K(n \rightarrow o). \quad (\text{A } 1)$$

For simplicity we discuss the case $N_{\text{try}} = 1$, and we drop the first subscript of $\xi_{j,q}$, which denotes the scaled coordinates of a trial position. More notational conventions are given in figure 1.

The flow from o to n is equal to the product of the probability of being in state o , the probability of generating state n , and the probability of acceptance (of the parallel exchange move):

$$K(o \rightarrow n) = \mathcal{N}(o) \times P(n) \times \text{acc}_{\text{exch}}(o \rightarrow n). \quad (\text{A } 2)$$

Configuration n can be generated in an infinite number of ways from configuration o . For this transition, the parallel exchange algorithm specifies that Q trial positions are generated in box 2, and the selected trial insertion position is $\xi(n)$. A set of trial positions is denoted by

$$\mathbf{b}(Q, n) = \{\xi_q \mid 1 \leq q \leq Q, \exists_q \xi(n) = \xi_q\}. \quad (\text{A } 3)$$

The probability of generating this trial set is

$$p_{\text{gen}}(\mathbf{b}) = d\xi_1 \dots d\xi_Q, \quad (\text{A } 4)$$

because the positions are drawn independently from a uniform distribution. The set of all sets $\mathbf{b}(Q, n)$ which include $\xi(n)$ is denoted by $\tilde{\mathbf{b}}(Q, n)$, i.e.,

$$\tilde{\mathbf{b}}(Q, n) = \{\mathbf{b}(Q, n)\}. \quad (\text{A } 5)$$

Note that this superset contains all possible ways of generating Q trial positions containing configuration n .

Now the flow from o to n has to be taken over all trial sets $\tau \in \tilde{\mathbf{b}}(Q, n)$:

$$K(o \rightarrow n) = \mathcal{N}(o) \sum_{\tau \in \tilde{\mathbf{b}}(Q, n)} p_{\text{gen}}(\tau) P(n|\tau) \text{acc}_{\text{exch}}(o \rightarrow n|\tau). \quad (\text{A } 6)$$

From

$$P[n|\tau(n)] = \frac{\exp[-\beta u(n)]}{Z[\tau(n)]}, \quad (\text{A } 7)$$

$$P[o|\tau(o)] = \frac{\exp[-\beta u(o)]}{Z[\tau(o)]}, \quad (\text{A } 8)$$

$$p_{\text{gen}}[\tau(o)] = p_{\text{gen}}[\tau(n)], \quad (\text{A } 9)$$

where trial set $\tau(n)$ means $\tau \in \tilde{\mathbf{b}}(Q, n)$, and from equations (A 1) and (A 2) it follows that

$$\begin{aligned} & \exp[-\beta W_B(o)] \exp[-\beta u(n)] \sum_{\tau(n)} \frac{\text{acc}_{\text{exch}}[o \rightarrow n|\tau(n)]}{Z[\tau(n)]} \\ &= \exp[-\beta W_B(n)] \exp[-\beta u(o)] \sum_{\tau(o)} \frac{\text{acc}_{\text{exch}}[n \rightarrow o|\tau(o)]}{Z[\tau(o)]}. \end{aligned} \quad (\text{A } 10)$$

This simplifies to

$$\sum_{\tau(n)} \frac{\text{acc}_{\text{exch}} [o \rightarrow n | \tau(n)]}{Z[\tau(n)]} = \exp(-\beta\Delta W_B) \frac{\exp[-\beta u(o)]}{\exp[-\beta u(n)]} \sum_{\tau(o)} \frac{\text{acc}_{\text{exch}} [n \rightarrow o | \tau(o)]}{Z[\tau(o)]}, \quad (\text{A } 11)$$

where $\Delta W_B = W_B(n) - W_B(o)$.

The detailed balance condition is contained in this equation. On the left-hand side of this equation the term $\text{acc}_{\text{exch}} [o \rightarrow n | \tau(n)]$ is present, which stands for the acceptance probability of going from configuration o to configuration n given a generated trial set $\tau(n)$. From the parallel exchange algorithm it follows that for each transition from o to n two trial sets are generated, i.e., one containing position $\xi(o)$ and one containing $\xi(n)$. Hence,

$$\text{acc}_{\text{exch}} [o \rightarrow n | \tau(n)] = \sum_{\tau(o)} p_{\text{gen}}[\tau(o)] \text{acc}_{\text{exch}} [o \rightarrow n | \tau(n), \tau(o)]. \quad (\text{A } 12)$$

Therefore equation (A 11) can be rewritten as

$$\sum_{\tau(n)} \sum_{\tau(o)} \frac{\text{acc}_{\text{exch}} [o \rightarrow n | \tau(n), \tau(o)]}{Z[\tau(n)]} = \sum_{\tau(o)} \sum_{\tau(n)} \mathcal{F}(n, o) \frac{\text{acc}_{\text{exch}} [n \rightarrow o | \tau(o), \tau(n)]}{Z[\tau(o)]}, \quad (\text{A } 13)$$

where

$$\mathcal{F}(n, o) = \exp(-\beta\Delta W_B) \frac{\exp[-\beta u(o)]}{\exp[-\beta u(n)]}. \quad (\text{A } 14)$$

For each term on the left-hand side there is a corresponding term on the right-hand side. By imposing the condition

$$\frac{\text{acc}_{\text{exch}} [o \rightarrow n | \tau(n), \tau(o)]}{\text{acc}_{\text{exch}} [n \rightarrow o | \tau(o), \tau(n)]} = \mathcal{F}(n, o) \frac{Z[\tau(n)]}{Z[\tau(o)]}, \quad (\text{A } 15)$$

for each pair of trial sets $\tau(n)$ and $\tau(o)$ one easily fulfils equation (A 13). Our parallel exchange move satisfies this condition, thereby proving that detailed balance is satisfied.

Appendix B

Here we demonstrate that at equilibrium the acceptance probability P_{exch} as given by the rule in equation (9) approaches one for a large number of processors Q and for large systems.

As before we assume that we are moving a particle from box 1 to box 2, and we introduce the notation $n_2 = N - n_1$ and $V_2 = V - V_1$.

From

$$\exp(-\beta\Delta W_B) \frac{\exp[-\beta u(o)]}{\exp[-\beta u(n)]} = \frac{n_1}{V_1} \frac{V_2}{n_2 + 1}, \quad (\text{B } 1)$$

and

$$\left\langle \frac{Z(n)}{Z(o)} \right\rangle = \left\langle \frac{\sum_{j,q}^{Q \times N_{\text{try}}} \exp[-\beta u_{j,q}(n)]}{\exp[-\beta u(o)] + \sum_{j,q}^{Q \times N_{\text{try}}-1} \exp[-\beta u_{j,q}(o)]} \right\rangle \quad (\text{B } 2)$$

$$\approx \frac{\langle Q \exp[-\beta u^+(n)] \rangle}{\langle \exp[-\beta u(o)] + (Q-1) \exp[-\beta u^+(o)] \rangle} \quad (\text{B } 3)$$

$$\approx \frac{\langle \exp[-\beta u^+(n)] \rangle}{\langle \exp[-\beta u^+(o)] \rangle}, \quad (\text{B } 4)$$

where $u^+(n)$ gives the test particle energy of a (ghost) particle in box 2 and $u^+(o)$ gives the test particle energy of a (ghost) particle in box 1 (we assume that Q is large), it follows that for large systems

$$\langle P_{\text{exch}} \rangle \approx \left\langle \frac{n_1}{V_1} \frac{V_2}{n_2 + 1} \right\rangle \frac{\langle \exp[-\beta u^+(n)] \rangle}{\langle \exp[-\beta u^+(o)] \rangle} \quad (\text{B } 5)$$

$$\approx \frac{\langle \rho_1 \rangle \langle \exp[-\beta u^+(n)] \rangle}{\langle \rho_2 \rangle \langle \exp[-\beta u^+(o)] \rangle} \quad (\text{B } 6)$$

(we can neglect the density fluctuations).

Now, the chemical potential μ_2 in box 2 (see [15], p. 43) is given by

$$\mu_2 \approx -\frac{1}{\beta} \ln \left\{ \frac{1}{\Lambda^3} \left\langle \frac{V_2}{n_2 + 1} \exp[-\beta u^+(n)] \right\rangle \right\}, \quad (\text{B } 7)$$

or alternatively

$$\langle \exp[-\beta u^+(n)] \rangle = \Lambda^3 \langle \rho_2 \rangle \exp(-\beta \mu_2) \quad (\text{B } 8)$$

where Λ is the de Broglie wavelength. At equilibrium $\mu_1 = \mu_2$ holds, and by substituting equation (B 8) (and a similar expression for μ_1) in equation (B 5) one obtains

$$\langle P_{\text{exch}} \rangle \approx 1. \quad (\text{B } 9)$$

References

- [1] PANAGIOTOPOULOS, A. Z., 1987, *Molec. Phys.*, **61**, 813.
- [2] ALLEN, M. P., and TILDESLEY, D. J., 1987, *Computer Simulation of Liquids* (Oxford: Clarendon).
- [3] SMIT, B., DE SMEDT, PH., and FRENKEL, D., 1989, *Molec. Phys.*, **68**, 931.
- [4] SIEPMANN, J. I., KARABORNI, S., and SMIT, B., 1993, *Nature*, **365**, 330.
- [5] SMIT, B., KARABORNI, S., and SIEPMANN, J. I., 1995, *J. chem. Phys.*, **102**, 2126.
- [6] PANAGIOTOPOULOS, A. Z., 1992, *Molec. Simul.*, **9**, 1.
- [7] ESSELINK, K., LOYENS, L. D. J. C., and SMIT, B., 1995, *Phys. Rev. E*, **51**, 1560.
- [8] FRENKEL, D., 1990, *Computer Modelling of Fluids, Polymers and Solids*, NATO ASI, edited by C. R. A. Catlow (Dordrecht: Kluwer).
- [9] NOSÉ, S., 1984, *J. chem. Phys.*, **81**, 511.
- [10] HOOVER, W. G., 1986, *Phys. Rev. A*, **34**, 2499.
- [11] MEHLIG, B., HEERMANN, D. W., and FORREST, B. M., 1992, *Phys. Rev. B*, **45**, 679.
- [12] DUANE, S., KENNEDY, A. D., PENDLETON, B. J., and ROWETH, D., 1987, *Phys. Lett. B*, **195**, 2.
- [13] ESSELINK, K., SMIT, B., and HILBERS, P. A. J., 1993, *J. comput. Phys.*, **106**, 101.
- [14] PANAGIOTOPOULOS, A. Z., QUIRKE, N., STAPLETON, M., and TILDESLEY, D. J., 1988, *Molec. Phys.*, **63**, 527.

- [15] SMIT, B., 1990, *Computer Simulation of Phase Coexistence: from Atoms to Surfactants*, Ph.D. thesis, Rijksuniversiteit Utrecht, The Netherlands.
- [16] SMIT, B., 1992, *J. chem. Phys.*, **96**, 8639.
- [17] SMIT, B., and WILLIAMS, C. P., 1990, *J. Phys.: condens. Matter*, **2**, 4281.
- [18] GLASER, M., 1994, personal communication.
- [19] BRODZ, F. A., and DE PABLO, J. J., 1994, *Chem. Engng Sci.*, **49**, 3015.