Recoil growth: An efficient simulation method for multi-polymer systems

S. Consta
FOM Institute of Atomic and Molecular Physics (AMOLF), Kruislaan 407, 1098 SJ Amsterdam, The Netherlands

N. B. Wilding[a)]
Department of Physics and Astronomy, The University of Edinburgh, Edinburgh EH9 3JZ, United Kingdom

D. Frenkel[b)]
FOM Institute of Atomic and Molecular Physics (AMOLF), Kruislaan 407, 1098 SJ Amsterdam, The Netherlands

Z. Alexandrowicz[a)]
Department of Materials and Interfaces, Weizmann Institute of Science, Rehovot 76100, Israel

(Received 28 August 1998; accepted 4 November 1998)

We present a new Monte Carlo scheme for the efficient simulation of multi-polymer systems. The method permits chains to be inserted into the system using a biased growth technique. The growth proceeds via the use of a retractable feeler, which probes possible pathways ahead of the growing chain. By recoiling from traps and excessively dense regions, the growth process yields high success rates for both chain construction and acceptance. Extensive tests of the method using self-avoiding walks on a cubic lattice show that for long chains and at high densities it is considerably more efficient than configurational bias Monte Carlo, of which it may be considered a generalization. © *1999 American Institute of Physics.* [S0021-9606(99)51406-2]

I. INTRODUCTION

Much current activity in polymer science focuses on systems of mutually interacting polymer chains, such as semidilute and concentrated solutions, melts, or chains tethered to a surface. Topics of interest include chain conformations, phase behavior, and relaxation dynamics with emphasis on entanglement, reptation and the glassy state. Besides the use of analytical methods, advances in the theoretical understanding of such systems draw heavily on computer simulation.[1] Unfortunately, simulations of multi-chain polymer systems are notoriously difficult and except for the lowest densities, are largely restricted to unrealistically short chains.[2,3]

The principal difficulty is one of chain entanglement. Large scale conformational rearrangements of a chain are hindered by excluded volume restrictions both with itself and with neighboring chains. This, of course, merely reflects the prevailing physical situation: Multi-chain dynamics are exceedingly slow on the time scales found in simple molecular systems. Consequently, simulation methods such as molecular dynamics (MD) which attempt to model the true physical dynamics, suffer from relaxation times that increase rapidly with chain length and density.[2] The same problem applies to Monte Carlo (MC) methods employing random local motion of chain segments, such as bond rotation algorithms or the bond fluctuation model.[2]

In many situations however, one is not explicitly interested in the intrinsic dynamics of the polymer system, but merely its static equilibrium properties. Alternatively one may wish to study relaxation, but lacks the means of efficiently preparing well equilibrated starting configurations. In such instances it is expedient to forego local MD and MC algorithms in favor of *artificial* MC dynamics, which permit a much more efficient exploration of configurational phase space. Unfortunately, most of the efficient algorithms tailored for a single chain, such as the pivot,[4] cannot be applied in the multi-chain context. One exception is the well known ''slithering snake'' algorithm which mimics the back and forth reptation motion of chains, without explicitly attempting to model the time consuming local segment fluctuations. For studies of chain diffusion in a dense multi-chain system, rapid relaxation was observed[5] using this method. Other studies, however, suggest that the slithering snake motion is prone to self-trapping[6] and that chains can become locked in ''cages.''[3] Furthermore, the method is inapplicable for tethered or nonlinear polymers.

Recently there has been some effort to develop MC algorithms specifically tailored to deal with multi-polymers systems. One such method has been proposed[7] that affects large conformational changes by splitting and bridging parts of chains. This approach is efficient, but necessarily introduces chain-length polydispersity into the system. Other specialized multi-chain algorithms suitable for lattice chains at very high densities close to or at saturation have also been developed.[8,9] However, their general applicability is limited and so will not be considered further here.

A more attractive class of MC algorithms suitable for multi-polymer simulations are *ab initio* ''growth'' algorithms that attempt to engineer large scale configurational changes by removing a chain from the system and placing it

[a)]Institut für Physik, Universität Mainz, Staudinger Weg 7, D-55099 Mainz, Germany
[b)]Electronic mail: frenkel@amolf.nl

elsewhere, or by replacing parts of a chain. The previously removed chain or chain portion is reinserted into the system with the aid of a growth algorithm. The viability of the technique as a whole depends crucially on the manner in which this growth is implemented. Several approaches are possible. The most basic is to employ a simple (nonreversal) random walk in which a chain of length N is grown as a sequence of N randomly oriented steps. The construction fails if the growing chain visits any point in space that is excluded due to the presence of other particles. In practice this approach is almost useless (particularly for dense multi-chain systems), since a growing chain sooner or later encounters excluded volume. As a result, the chain construction rate falls exponentially with N, i.e., $f_{\mathrm{con}} \simeq \exp(-c_0 N)$.

In view of the high attrition rate for the simple random walk, Rosenbluth and Rosenbluth (RR)[10] introduced a biasing technique for increasing the chain construction rate. In the original formulation of this method, one considers a single chain growing on a lattice. At the ith step of its construction, the chain is accorded k possible distinct random growth directions (on a lattice k is usually taken as $q-1$, with q the coordination number). Unlike the simple random walk, however, the chain at each step does not blindly choose a random direction from the set of k. Instead it examines all k possibilities and chooses the actual step from the subset of w_i directions that avoid excluded volume. If $w_i = 0$, then the growing chain has run into a dead-end or ''trap'' and the construction fails. Of course, choosing the actual step from the subset of w_i directions instead of the full complement of k, for $i=1, \ldots N$, introduces a bias. The total construction bias accumulated over N steps is subsequently compensated for in the calculation of averages.

Although considerably more efficient than the simple random walk construction mentioned above, the RR technique is still inefficient for very long chains due to its shortsightedness. By looking ahead only one step, it often blunders into traps. This trapping means that the attrition is still ultimately exponential in N, i.e., $f_{\mathrm{con}} \simeq \exp(-c_1 N)$, although with a coefficient $c_1 \ll c_0$. Alternatively the attrition can be reduced by employing the ''scanning method'' of chain growth.[11] For each of the k directions of an ith step, one constructs a Cayley tree of all free pathways of l further steps. The actual next step is then made probabilistically dependent on the number of progeny spawned by each of the k directions. However the CPU time expended on constructing the Cayley tree increases exponentially with l, which limits the scanning method considerably. In view of this problem, a variant of the method known as Double Scanning[12] has been developed in which only a subset of all possible future paths are explored. This improves the efficiency of the method, but both renders the sampling approximate and reduces the ability to circumvent traps.

The RR algorithm, originally developed for the study of single chains, has recently been incorporated into a multichain MC scheme known as configurational bias Monte Carlo (CBMC).[13–15] Within the canonical ensemble formulation of the CBMC scheme, one considers a system containing a fixed number of chains, each of length N. A chain is chosen at random and an attempt made to regrow it elsewhere in the system, using the RR scheme described above. If the construction succeeds, the new chain is accepted and replaces the old chain with a probability dependent on the ratio of the total construction bias of the new and old chains. The CBMC method can be used with both lattice and continuum polymer models.[16]

The chief advantage of the CBMC method is that large scale configurational changes can be performed at one go. Indeed the method has proved itself very effective at low and moderate densities for chain lengths up to a few dozen monomers. Unfortunately, it transpires that the acceptance rate for chain insertions falls exponentially with increasing chain length. This problem is symptomatic of the fact that the RR method produces chains having a probability distribution that differs exponentially in N from the Boltzmann distribution.[17] Taken together with the exponential attrition rate for chain construction, it is clear that the method will be inefficient when dealing with long chains and/or very dense systems. In such situations, the best that can be done is to try to regrow terminal or intermediate portions of chains.[18,19]

In this paper we propose a new MC method suitable for dense multi-chain systems. Our approach is close in spirit to CBMC, but instead of using the standard RR algorithm to grow chains, we employ a more sophisticated biased growth technique which we call ''recoil growth.'' The strength of the recoil growth method resides in its ability to avoid traps and excessively dense regions by deploying a retractable ''feeler'' of maximum length l, to probe the territory ahead of the growing chain. This feeler recoils from traps and dense regions, and thus guides the chain along favorable pathways. The result is a substantial improvement (compared to CBMC) in chain construction and acceptance rates at high density and long chain lengths. Such a recoil idea has been proposed by Alexandrowicz and is described in a recent publication.[20] The present application of recoil is, however, different from that of Ref. 20. In the present version, a fully grown new chain replaces an old one with the help of a MC accept–reject step determined by their respective weights, just like in CBMC. In contrast Ref. 20 utilizes a sequence of stepwise lotteries ''on the go,'' which are applied as the new chain is being grown and are determined by its own weight alone. Such a procedure involves a certain approximation (i.e., it is not exact) but is substantially more efficient.

The layout of our paper is as follows. In Sec. II we describe our recoil growth algorithm in detail and show how it may be incorporated within a MC scheme that fulfills the detailed balance condition. Then, in Sec. III we test the validity of the method and investigate its properties via extensive simulations of self avoiding walks on a cubic lattice. A comparison of efficiency with the CBMC method is also made. Finally Sec. IV summarizes and discusses our findings.

II. DESCRIPTION OF THE ALGORITHM

The form of our proposed recoil growth scheme is motivated by the need to address the two principal deficiencies of the RR growth method. The first deficiency is the inability to avoid traps. These traps have a wide distribution of depths,[17,21] and owing to its myopia, the RR algorithm is

incapable of avoiding them. Since the number of traps increases rapidly with density, the RR algorithm thus suffers low construction rates, particularly for long chains.

The second deficiency relates to the distribution of chains produced by the RR algorithm. In order to avoid excluded volume and thus maintain a relatively high construction rate, it is necessary to employ a rather large number of alternative growth directions k. Unfortunately, this leads to an indiscriminate growth procedure that yields an ensemble with low weight chains. As a result, the acceptance rate for CBMC moves falls exponentially with chain length. Clearly, therefore, any proposed improvement to the RR growth algorithm must strive to improve both the construction rate *and* the acceptance rate. Our recoil growth algorithm achieves this by positioning at the head of the growing chain, a long retractable feeler having the ability to recoil from traps and excessively dense regions. The advantage comes from the fact that if the growth procedure encounters a trap or a dense region, we do not terminate the growth of the whole chain, but merely recoil back the required number of steps (up to a maximum length l), and try elsewhere. In this way the growing chain both avoids traps and finds better pathways through the system.

In what follows we describe how the recoil growth scheme can be incorporated into a canonical MC scheme in which individual chains are regrown within the system.

A. Chain construction

A new chain is generated according to the recoil growth procedure as follows. To begin, the first monomer (denoted by $i=1$) is placed at random in the system. From this first monomer one attempts to grow a single step (bond) to the second ($i=2$) monomer. For this purpose a maximum of k distinct directions are permitted. One chooses a random direction and attempts to grow the chain a single step. If this growth is blocked, however, then another direction is chosen, repeatedly if necessary, up to the maximum of k attempts. Possible reasons for blockages are excluded volume interactions with previous monomers of the chain or with other chains. If at the bth attempt no excluded volume is encountered, we place the second monomer on the lattice, record the number of unused directions $k-b_1$, and proceed to grow the next step. The same process is repeated for all subsequent monomers $i=1, \ldots N$: the chain grows at the b_i attempt and the number of unused directions $k-b_i$ are recorded for all i.

Suppose, however, that at the ith monomer, the chain failed to grow to the length $i+1$ within the maximum of k possible attempts. In this case it returns to the $i-1$ monomer and renews its attempts to grow from there, using the $k-b_{i-1}$ previously unused directions. Similarly if it fails to grow a step at the $i-1$ monomer (within the total number of k attempts), it falls back to the $i-2$ monomer. In difficult situations the chain may thus repeatedly grow and fall back. However, it is not permitted to fall back indefinitely. If $i_{\max}$ is the *greatest* length that the chain attained in its whole growth history, then it may fall back no further than to the length $i=i_{\max}-(l-1)$. If the chain does recoil to this length (or if $n<l$ and it recoils to its starting point) and still fails, then the growth process terminates. We note that once the chain has grown up to the maximal length $i_{\max}$, monomers $i=1, \ldots, i_{\max}-(l-1)$ are fixed and no longer subject to recoil growth. This ''fixed chain'' only advances by a step when the number of steps ahead of it has attained length l. Chain monomers $i>i_{\max}-(l-1)$ may be thus regarded as constituting a *retractable feeler of maximum length* l, which allows the growing chain (of fixed monomers) to ''look ahead'' by l steps. In this sense our scheme bears some resemblence to the double scanning method of Meirovitch,[12] although as discussed further in Sec. IV, there are important differences.

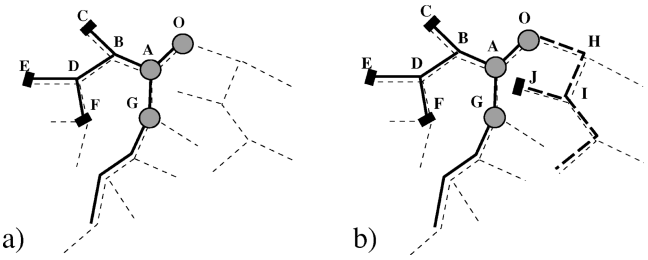

FIG. 1. (a) An example illustration of the recoil growth procedure described in the text, for $k=2, l=3$. The underlying random tree is shown as a thin dashed line. The initial monomer is placed at point O and the following scenario is played out: The growing chain first tries the path $OABC$, but finds it blocked. It recoils to point B and then succeeds to attain length l along the path $OABD$. The monomer at A then becomes fixed. Construction continues by examining the paths DE, DF which are both blocked, so the chain recoils again to point A. The open path $OAG\ldots$ is subsequently found and the second monomer is fixed at point G. (b) For the weight calculation, the backbone of the candidate chain is retraced and from each monomer i, b_i feelers are grown starting from the directions that remained *unused* during construction. From O one feeler is grown. Path $OHIJ$ is blocked, and the feeler recoils to point I, from which it subsequently attains length l. No feeler is grown from point A. The respective weights assigned to points O and A are thus: $w_1=2$, $w_2=1$.

An example of a possible recoil growth scenario is illustrated schematically in Fig. 1(a). The growth procedure can be envisaged as taking place on a underlying random tree that is revealed as the growth proceeds. The tree is formed by assigning random directions to the monomers. This concept of a tree will be used below for demonstrating that the method obeys the detailed balance condition. Should the new chain attain the desired length N, then it becomes a candidate for replacing some randomly chosen existing chain, and one proceeds to calculate the weights of both chains.

B. Weight calculation

In order to incorporate the recoil growth scheme within a MC framework it is necessary to obtain the probability of generating a particular chain configuration on the underlying random tree, since this quantity enters into the detailed balance condition described below. For convenience we shall work with the inverse of this probability, which we refer to as the chain ''weight.'' The calculation of this weight proceeds with the use of feelers. Since the procedure for weight calculation differs slightly for the candidate (new) chain and the existing (old) chains, we shall describe them separately.

1. Candidate (new) chain

One visits the successive monomers of the new chain and from each one attempts to grow (in turn) feelers of length l, exploring the $k-b_i$ first step directions that remained untried in the construction process (a feeler already exists along the chain backbone itself). These feelers are grown using exactly the same recoil growth procedure as employed for chain construction. Thus they are allowed to retreat if necessary, because of blockage by monomers belonging to the same feeler, to the candidate chain (up to where the feeler starts), and to other chains. However if a feeler recoils by l steps back to its starting point, it is deemed to have failed; successful feelers attain the length l. (For the last $i=N-l+1\cdots N$ monomers, the feelers are progressively shortened by one step). Examples of feeler growth scenarios are given in Fig. 1(b). The weight (inverse construction bias) at each step of the new chain is given by the number of alternative directions in which the ith to $i+1$th step could have been taken under the rules of our recoil growth scheme (we shall elaborate on this point at the end of this section). The total chain weight is, therefore,

$$W_n=\prod_{i=1}^{N-1}\frac{w_i}{k}, \tag{1}$$

where w_i is the number of successful feelers starting at monomer i (including the one along the existing chain) and for convenience we have normalized by k.

2. Existing (old) chain

When an existing chain is selected for possible replacement by the candidate new chain, its weight W_o is computed by constructing new feelers along $k-1$ directions randomly assigned to the monomers (the direction along the existing chain is of course not examined). Otherwise the calculation of W_o proceeds exactly as for the candidate new chain, i.e., with the help of Eq. (1).

C. Detailed balance and acceptance probability

In a MC phase space trajectory, microstates are visited with the appropriate equilibrium probability distribution if one imposes the detailed balance condition, which demands equality of the transition rates from some initial (old) state o to some final (new) state n, and back again, i.e.,

$$q_nP(n\rightarrow o)=q_oP(o\rightarrow n), \tag{2}$$

where q is the equilibrium probability (Boltzmann) distribution to which the old and new states belong, and $P(n\rightarrow o)$ is the transition probability from state n to state o.

Our canonical ensemble Monte Carlo procedure involves attempting to exchange an old chain with a new chain. For the purposes of demonstrating detailed balance, one can imagine this to occur with the help of random trees as follows.

(i) A random tree t_n is generated having k bifurcations at each step, cf. Fig. 1.

(ii) The recoil growth process is executed upon the random tree t_n and a new chain configuration is generated.

(iii) A random tree t_o is generated around the old chain, such that the old chain configuration lies on the tree.

As we shall see, the last step (iii) is necessary so that one can define a reverse Monte Carlo move.

Clearly within this formulation, the probability of generating given random trees for both the new *and* old chains enters the expression for the transition probability, which is given by

$$P(n\rightarrow o)=\sum_{t_o,t_n}P_g(c_n,t_n)P_g(t_o|c_o)P_a((c_o,t_o)\rightarrow(c_n,t_n)). \tag{3}$$

Here, $P_g(c_n,t_n)$ is the probability of generating the new chain configuration c_n on the tree t_n with the recoil growth algorithm. This may be written $P_g(c_n,t_n)=P_g(t_n)P_c(c_n|t_n)$ where $P_g(t_n)$ is the *a priori* probability of generating a given tree t_n and $P_c(c_n|t_n)=1/W_n$ is the probability of constructing the new chain on that tree. $P_g(t_o|c_o)$ in Eq. (3) is the probability of generating the old tree t_o given the old chain configuarion c_o. The probability of accepting the exchange of chains $(o\rightarrow n)$ is given by $P_a((c_o,t_o)\rightarrow(c_n,t_n))$, and the sum extends over all possible combinations of new and old trees. The probability of the reverse move $P(n\rightarrow o)$ is simply obtained by subsitituting n by o and vice versa in Eq. (3).

Detailed balance [Eq. (2)] is satisfied if one imposes the stronger condition of ''superdetailed balance,''[16,22] namely that microscopic reversibility is fulfilled for *every* particular choice of random trees t_o and t_n. If we set the tree generation probability $P_g(t)$ to be uniform for all trees, one readily finds that the new state should be accepted with a probability

$$P(o\rightarrow n)=\min\left(1,\frac{q_oW_n}{q_nW_o}\right). \tag{4}$$

This is the acceptance probability used in the Metropolis step of the MC scheme.

D. Remarks on the algorithm

Having completed our definition of the recoil growth MC method, the following remarks are appropriate.

The use of the random tree concept in the detailed balance condition simply constitutes a generalization of the concept of a specific random choice of possible one-step directions, which was introduced in Ref. 16 to demonstrate the validity of continuum CBMC. Indeed, for feeler length $l=1$, our method simply reduces to CBMC. However, the tree concept also serves to clarify a somewhat subtle point concerning the way we have formulated the construction algorithm in Sec. II A. Clearly an alternative chain construction procedure would be to attempt to grow at each step j of the fixed chain, *all* k feelers, and to choose the $j+1$ monomer randomly from among the set of first step directions of the successful feelers. However, the picture of chain growth on an underlying random tree shows that since all feeler directions are chosen randomly, it is permissible (and indeed

much more computationally efficient) to work with just one feeler at a time during construction. Thus the fixed chain grows from the j to $j+1$ monomer along the first step of a successful feeler. The remaining steps of this feeler then constitute an incomplete feeler of length $l-1$ with respect to the $j+1$ monomer. One then proceeds to attempt to extend this feeler to length l using the normal recoil growth prescription. If successful, the chain grows to the $j+2$ step along the first step of the feeler and the process repeats. Only if the existing feeler cannot be extended to length l need one attempt to grow another feeler from scratch. Of course, if the chain successfully attains length N, it is necessary to return and attempt to grow the remaining feelers $k-b_j$ from each monomer in order to calculate the weight of the candidate new chain.

With regard to the chain weight W defined in Eq. (1), we note that this is simply the product of the number of allowable directions w_i in which the chain could have grown at each step. However, it is important to appreciate that the *criterion* for what constitutes an allowable direction is arbitrary—it effects only the efficiency and is irrelevant to detailed balance. In our recoil growth scheme, the criterion adopted requires that an allowable first step direction may be continued for $l-1$ further steps, such that the entire sequence avoids excluded volume. Therefore, the number of surviving feelers w_i at each monomer constitutes the number of single step directions (out of the maximum number of k) that satisfy our criterion. Accordingly, the weight W defined in Eq. (1) is the total inverse bias of a chain's construction and by compensating for W in the acceptance probabilities, we ensure that detailed balance is obeyed. Of course, other criteria for the number of allowable directions at each step are also possible. Less stringent than ours is the RR criterion of CBMC, which requires that an allowable growth direction be able to continue for only one step (i.e., $l=1$). Still more lenient is the construction of self avoiding chains with the help of the simple random walk which allows all directions, including those that do encounter excluded volume. All three methods satisfy detailed balance, but they calculate Ws that increase, respectively, and pay the price in decreasing construction efficiency. Thus our intricate recoil growth procedure is merely an information gathering device that enables us to make a ''good'' decision in the sense of efficient sampling.

Finally, we note that if we make k smaller than the coordination $q-1$, our calculation of weight W for a given chain configuration is stochastic in the sense that a feeler's survival depends to some extent on the random choice of the k directions at each step of the l-step feelers. Thus, sometimes the weight of a chain will be underestimated, while at other times it will be over-estimated with respect to a measurement using the full complement $k=q-1$. On average, however, one expects that the correct distribution of Ws will be produced. Stochastic sampling of W is also unavoidable in the continuum implementations of the CBMC method.[16] Incidentally, k need not be an integer. Thus, for example, $\langle k\rangle = 3.5$ is obtained by half the time assigning $k=3$ and half the time assigning $k=4$. For long chains we show below that the algorithm's efficiency can be rather sensitive to the choice of $\langle k\rangle$.

III. RESULTS

To study the properties of the proposed recoil growth MC method we have performed canonical ensemble simulations of self avoiding walks on a simple cubic lattice. We begin by describing tests to establish the validity of the method and then proceed to examine its characteristics with respect to chain construction and acceptance. Finally, we assess the methods efficiency at relaxing the sample and compare it to that of the CBMC algorithm.

Our first task was to verify that the method produces chains with the correct statistical properties. This was achieved by means of a comparison with the CBMC method for the following two sets of chain length and density consisting of 100 chains:

(i) $N=100$ at a monomer density of $\rho=0.1$.
(ii) $N=40$ at a monomer density $\rho=0.6$.

These are conditions for which CBMC operates reasonably well and should, therefore, supply bountiful statistics. A large number of independent equilibrated system configurations were generated using both the CBMC method ($k=5$, $l=1$), and the recoil growth method with $k=3$, $l=5$. For both schemes we monitored the distributions of two quantities, namely the square of the gyration radius R_g^2, and the logarithm of the Rosenbluth weight W_{RR} of the chains. This latter quantity is defined as $W_{RR}=\Pi_{i=1}^{N} w_i/k$, where w_i is the number of free first-step continuations from the ith monomer, including the step of the main chain itself. The results for system (i) are shown in Fig. 2. Clearly there is a high degree of accord between the CBMC and recoil growth methods with respect to the distributions of R_g^2 and $\ln W_{RR}$. For the averages, we obtain $\langle R_g^2\rangle = 32.25(4)$, $\langle \ln W_{RR}\rangle = -8.252(8)$ for CBMC and $\langle R_g^2\rangle = 32.22(4)$, $\langle \ln W_{RR}\rangle = -8.263(12)$ for recoil growth. For system (ii) we obtain for the averages of the distributions $\langle R_g^2\rangle = 11.120(26)$, $\langle \ln W_{RR}\rangle = -19.86(6)$ for CBMC and $\langle R_g^2\rangle = 11.136(13)$, $\langle \ln W_{RR}\rangle = -19.84(3)$ for recoil growth.

We now turn to an examination of the N dependence of the chain construction and exchange acceptance rates $f_{\mathrm{con}}(N)$ and $f_{\mathrm{acc}}(N)$, respectively, for various values of $\langle k\rangle$ and l. Later we will see that for given choice of N and ρ, a relatively fast search locates compromise values of $\langle k\rangle$ and l which ensure the good performance of the recoil growth method. At present, however, we wish to elucidate systematically the role played by these parameters. To this end it is instructive to fix the feeler length at a moderate value of $l=5$ and consider the effect of varying k.

The results for $f_{\mathrm{con}}(N)$ are presented in Fig. 3(a). They show that if the construction utilizes the full complement of directions $k=q-1$, the growing chain avoids almost all traps and we find $f_{\mathrm{con}}(N)\approx 1$ over the entire range of N studied. As $\langle k\rangle$ decreases it still suffices to compensate the average loss of continuations due to excluded volume, but owing to fluctuations $f_{\mathrm{con}}(N)=\mathrm{const}\lesssim 1$. Figure 3(a) shows that this is indeed the case for $k=5,3$. However, if we set ''starvation'' values of $\langle k\rangle = 2.0$, or $\langle k\rangle = 1.8$, which do not compensate

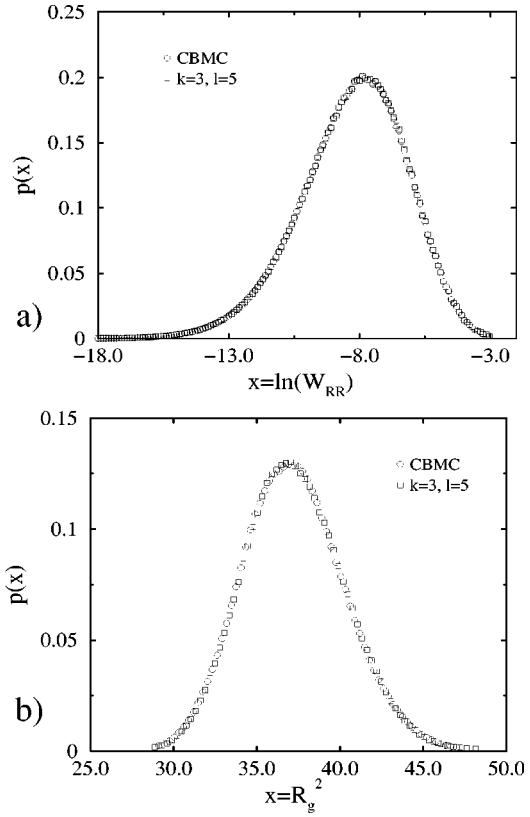

FIG. 2. Histograms showing (a) the logarithm of the Rosenbluth weights, for CBMC and recoil growth ($k=3,l=5$). (b) The radius of gyration squared. The length of the MC run is 2×10^8 cycles.

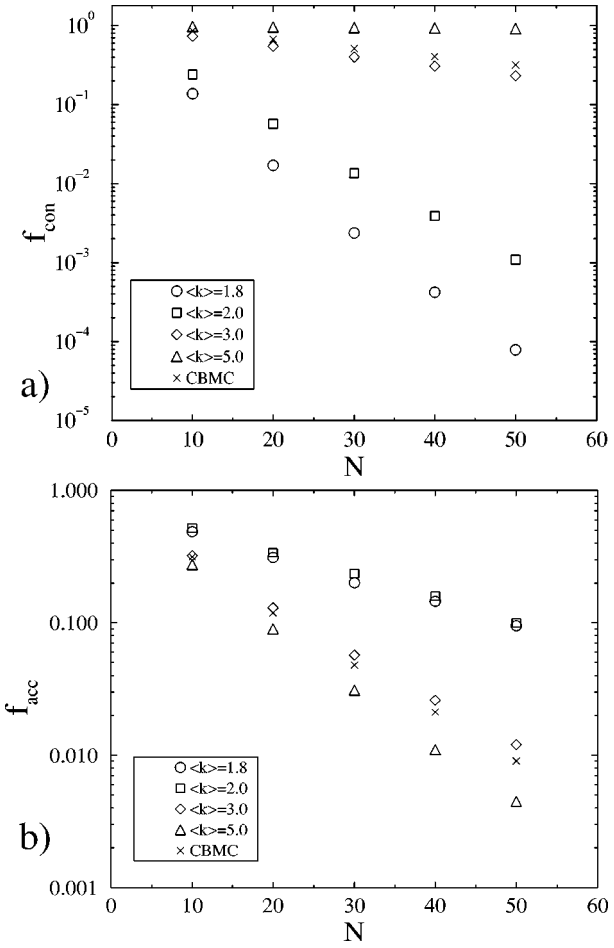

FIG. 3. (a) Construction rate $f_{\mathrm{con}}(N)$ for various values of $\langle k\rangle$. All simulations were carried out for 100 chains at density $\rho=0.5$, using $l=5$. (b) The corresponding exchange acceptance rate $f_{\mathrm{acc}}(N)$.

for the average loss due to excluded volume, we pass to a strong exponential attrition. The sharpness of the transition between the compensation and the starvation regime makes it easy to determine the marginal value of k that ensures that excluded volume is just compensated.

Having demonstrated that for a given moderate l, a large $\langle k\rangle$ helps f_{con}, we now show that a minimal $\langle k\rangle$ close to (but not within) the starvation regime serves to benefit the exchange acceptance rate, f_{acc}. As mentioned in the preceeding section, a large $\langle k\rangle$ leads to an indiscriminate growth that yields many low quality chains having relatively small weights. This implies a random sampling which differs exponentially from the correct Boltzmann distribution, which in turn leads to an acceptance rate $f_{\mathrm{acc}}(N)$ that falls exponentially with N. However, a minimal value of $\langle k\rangle$ causes the construction to recoil from relatively dense regions, and grow anew along a less difficult path, thus producing ''higher quality'' (relatively large weight) chains. Figure 3(b) shows $f_{\mathrm{acc}}(N)$ for $l=5$ at various $\langle k\rangle$. One sees that although the acceptance rate still decreases approximately exponentially with chain length, it improves markedly as k decreases.

It transpires, however, that in many cases f_{acc} is a highly misleading indicator of the sampling efficiency because the majority of chain exchanges are confined to small distinct regions of the system, leaving the other chains mostly untouched. The problem is traceable to the fact that some chains (those having low weight) are easily removed from the system. When such a chain is removed, a large vacancy is created into which another chain will preferentially grow shortly afterwards. Since the new chain is necessarily highly correlated with that it replaced, it too is quickly removed. This creates the illusion of a brisk turnover of chains, while in fact the configuration as a whole remains substantially unaltered. This effect has also been recently noted by other authors in connection with the CBMC method.[3,23]

In view of the unreliability of f_{acc} for indicating the true rate of relaxation, we have adopted an alternative measure of efficiency, namely the effective chain turnover rate, defined as the CPU time required to replace 95% of some arbitrary starting sample of chains. Optimal values of $\langle k\rangle$ and l are thus those that maximize this effective turnover rate, or indeed any other suitably chosen time autocorrelation function. In fact, in practice this optimization is not difficult to achieve and suitable values can be readily gauged from very short runs on small systems. In most cases we find that a choice of feeler length in the range $3\lesssim l\lesssim 8$ is a good starting point. The optimal choice of $\langle k\rangle$ depends on the density (and to a much smaller degree on the choice of l). Larger $\langle k\rangle$ values, approaching the coordination number are required at higher densities. The sensitivity of the efficiency to the assignment of $\langle k\rangle$ increases with chain length, so that for short chains ($N\approx 40$) $\langle k\rangle$ should be correctly estimated to within ± 0.5, while for longer chains ($N\gtrsim 100$) the value should be chosen

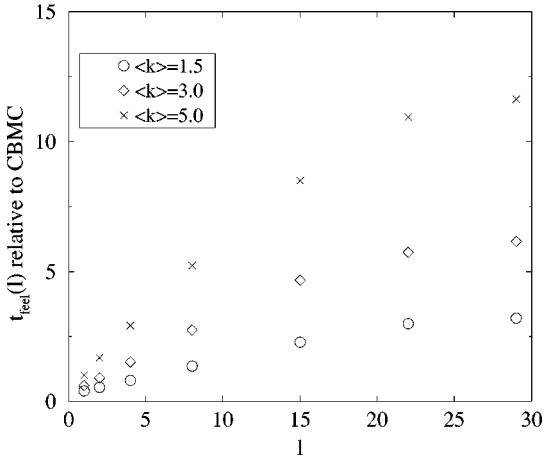

FIG. 4. The feeler construction time $t_{\text{feel}}(l)$ normalized with respect to CBMC ($k=5, l=1$) for various values of $\langle k \rangle$, for $N=30$, $\rho=0.5$.

to within ± 0.1. It should be pointed out that since the optimal $\langle k \rangle$ value is often much less than the lattice coordination number 5, the cost of growing feelers is surprisingly low. This point is illustrated in Fig. 4 where we plot for a number of $\langle k \rangle$ values the CPU time $t_{\text{feel}}(l)$ (normalized with respect to CBMC) to grow feelers of length l (for $N=30\rho=0.5$). One sees from the figure that for small l the time rises at most linearly in l (for large l the curves round off due to the progressive shortening of the feelers as the chain end is approached); in general we expect the time expended on feeler growth to increase approximately linearly in their length, with a slope that decreases markedly with $\langle k \rangle$.

To demonstrate that recoil growth method leads to net efficiency gains we have studied the decay of the autocorrelation function of the radius of gyration as a function of CPU time for a system of 100 chains of length $N=40$ at density $\rho=0.6$. Figure 5 shows that under these conditions, the efficiency of the recoil scheme is some three times higher than CBMC. As another example we have studied the rate at which the method replaces a given equilibrated starting con-

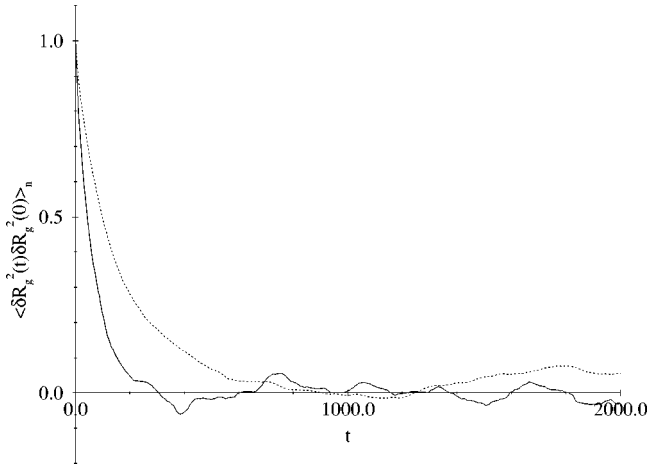

FIG. 5. Autocorrelation function of the square of the gyration radius vs CPU time for a system of 100 chains of length $N=40$ at density $\rho=0.5$. The dashed line and the solid line represent the CBMC and the recoil growth scheme for $k=3, l=5$, respectively. The subscript n in the average denotes that the correlation function is normalized to 1.

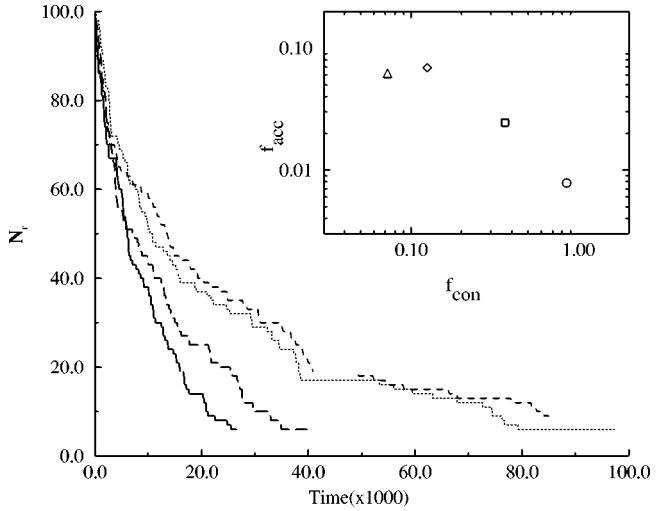

FIG. 6. The remaining fraction of the original sample of 100 chains of length $N=100$, density $\rho=0.5$ as a function of CPU time (s/ALPHA). The solid, long dashed, and dotted lines correspond to the recoil growth with $(k,l)=(2.7,10),(3,10),(5,4)$, respectively. The short dashed line corresponds to the CBMC results. The inset shows the associated construction and the percentage of acceptance rates on a log scale. The diamond, square, and circle refer to recoil growth with $(k,l)=(2.7,10),(3,10),(5,4)$, and the triangle to CBMC.

figuration of 100 chains each of length $N=100$ at density $\rho=0.5$. Figure 6 shows the fraction of chains replaced as a function of CPU time for various values of $\langle k \rangle$ and l as detailed in the caption. We find that the effective chain turnover rate is maximized for a choice $\langle k \rangle = 2.7$, $l=10$, being some six times faster than CBMC. A more comprehensive comparison of the relative efficiency of recoil growth and CBMC is presented in Fig. 7, which shows the relative efficiency of the two methods in achieving 95% turnover as a function of chain length N for three different densities $\rho = 0.3$, 0.5, and 0.7. One sees that at high densities and large chain lengths, optimized recoil growth is more efficient by up to a factor of 50. In Table I the parameters of the recoil growth used to construct Fig. 7 are presented.

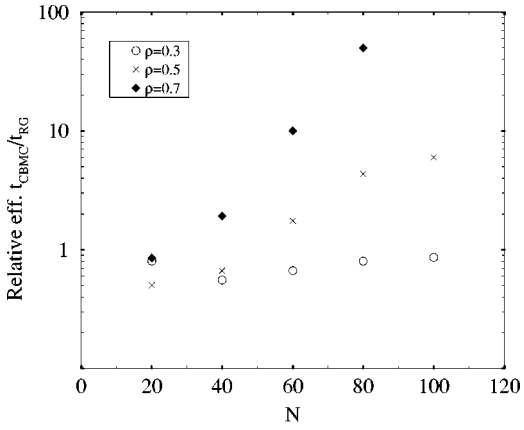

FIG. 7. The efficiency of recoil growth relative to CBMC, measured as the ratio of the CPU times required to remove 95% of a given equilibrated starting sample of 100 chains. Data are shown for a number of chain lengths at densities $\rho=0.3$, 0.5, and 0.7. In each instance, the recoil growth parameters $\langle k \rangle$ and l were chosen to yield the fastest relaxation rate.

TABLE I. The $(\langle k\rangle, l)$ values that correspond to Fig. 7.

L	20	40	60	80	100
ρ					
0.3	(2.5,3)	(2.5,3)	(2.5,4)	(2.3,6)	(2.4,6)
0.5	(2.7,3)	(2.5,5)	(2.7,6)	(2.7,9)	(2.7,10)
0.7	(3.1,4)	(3.9,4)	(3.75,6)	(3.5,8)	(4.9,10)

IV. DISCUSSION AND CONCLUSIONS

In summary we have proposed a new simulation method for dense, long-chain multi-polymer systems based on ''recoil growth.'' If a chain fails during the construction due to excluded volume, it is permitted to recoil a step and try to grow anew in another direction up to a total of k trials at each step. If still unsuccessful it may recoil a further step etc., up to a maximal predetermined length $l-1$. The construction bias is compensated for in the chain insertion–deletion stage in the same manner as for CBMC. The recoil allows the chain to avoid traps of length up to $l-1$ and hence benefits the construction rate f_{con}. However, it also allows the chain to avoid excessively dense regions of the system, growth into which would otherwise produce low quality (low weight) chains that fail the accept–reject lottery. Thus recoil also benefits f_{acc}, an effect that becomes more pronounced as the number k of directions is reduced to the barest minimum that still yields a reasonable construction rate f_{con}. The time expended on feelers also decreases markedly with decreasing k, making it possible to find (by means of a rapid preliminary search), values of k and l that optimize the relaxation rate and thus yield a substantial improvement in CPU time efficiency over CBMC. In this way the method extends the range of densities and chain lengths that can effectively be studied.

Although the major factor in the faster relaxation of recoil growth is its ability to build chains that more closely match the Boltzmann distribution, one may speculate that the stochastic nature of the weight determination procedure also plays a role in this regard. Random downward fluctuations in the weight assignments to old chains may help to dislodge high-weight chains which would not otherwise be exchanged using CBMC. Additionally, the nonlocal sampling resulting from use of long feelers may mean that relaxation (chain exchanges) in one area of the system influences the weight calculations for other quite distant chains, thus promoting their exchange too.

It is instructive to compare and contrast our method with the double scanning method (DSM) of Meirovitch[12] since, in a sense, both strive to achieve the same end, namely a stochastic look ahead scheme designed to seek out favorable pathways for the growing chain. However, the manner in which these paths are chosen differs greatly between the two methods. In the DSM, at each step of the growing chain, a typically large number (50–200) of feelers are grown according to the RR prescription. The next step direction of the growing chain is decided probabilistically dependent on the proportion of feelers surviving along each possible next step direction. In the RG scheme, however, only a single feeler is required during the construction process and the weight calculation differs from that of DSM in that next step directions receive either a weight of zero or unity. The need for many RR-type feelers to scan the future pathways is obviated by the ability of the RG feeler to recoil from excluded volume and the fact that a successful feeler is ''recycled'' at the next step of the main chain as an incomplete feeler (see Sec. II A). We, therefore, believe RG to be considerably more efficient than DSM at locating favorable pathways for the growing chain. To date, the DSM growth method has not been incorporated within an exact multipolymer Monte Carlo scheme.

It is interesting to note also a resemblance between our recoil growth and an optimized enrichment algorithm.[26] The latter constructs the ensemble of configurations for an isolated chain by allowing $\langle k\rangle$ alternative growth directions at each step. As with recoil growth, $\langle k\rangle$ is chosen so that the number of alternative growth directions equals the average loss per step due to excluded volume. Both methods require a tuning of $\langle k\rangle$ to avoid wild fluctuations of ''W.'' However, one enrichment construction produces a tree of W alternative (correlated) chain configurations, while recoil growth converts W into the weight of a single configuration. In fact, very recently a new MC growth scheme based on enrichment has been proposed.[27] The weights of the chains are confined within a desired range by eliminating probabilistically configurations with very small weight, while at the same time enriching the sample with copies of high weight chains in such a way that the resulting sample is unbiased. The method allows one to study very large chain lengths at low to moderate densities. Unfortunately we know of no reported tests of the method at high densities and can, therefore, not compare with the recoil growth method in this regime.

A number of extensions to the recoil growth method can also be envisaged. As with CBMC there should be no problem in applying the method to continuum systems. The effects of temperatures can also be incorporated by introducing a threshold for feeler survival based on its total Boltzmann weight. The construction rate for very long chains in the dense regime could also be improved by adopting a ''divide and conquer'' approach in which one tries to regrow only short chain sections at a time. Although we have presented our method within a canonical framework, it is also easy to generalize to other ensembles such as the grand canonical (constant-μVT) or Gibbs ensemble as has been done for CBMC.[24,25] In such cases, one is obliged to insert entire chains into the system (rather than regrow chain portion as is often done in canonical simulations of long chain systems), and the advantages of recoil growth over CBMC are expected to exceed those of the canonical ensemble case.

Finally, there are a number of interesting physical problems to which the recoil growth method can be applied, such as chains tethered to a surface, or branched and star polymers. Use of an open ensemble would also permit the study of phase behavior of polymer solutions and melts.[28]

ACKNOWLEDGMENTS

N.B.W. and Z.A. thank Kurt Binder, Kurt Kremer, Marcus Müller, and Peter Sollich for useful discussions. N.B.W. acknowledges the financial support of the EPSRC (Grant No. GR/L91412), the Royal Society of Edinburgh and the British

Council ARC programme. Z.A. acknowledges a Humboldt Award. S.C. acknowledges the FOM-Institute and E.U. for financial support [Contract No. 95032 CGP.4, Contract No. ERBFMBICT-972312]. The work of the FOM Institute is part of the research program of ''Stichting Fundamenteel Onderzoek der Materie'' (FOM) and is supported by NWO (''Nederlandse Organisatie voor Wetenschappelijk Onderzoek''). S.C. and D.F. thank Jonathan Doye for critical reading of the manuscript.

APPENDIX

In this Appendix, we present a more formal description of the various factors affecting the efficiency of the recoil growth method. A similar method for CBMC has been presented in Ref. 29.

The success of the recoil-growth method relies on the ability to tune the parameters k and l in order to obtain a good chain turnover rate for low computational cost. The efficiency of any MC scheme can be estimated by $(t_a t_{\mathrm{MC}})^{-1}$ where t_a is the autocorrelation time in number of MC steps and t_{MC} is the CPU time for an MC step.[6] Thus the parameters of the recoil growth scheme should be optimized to minimize the product $t_a t_{\mathrm{MC}}$. The additional cost in extending the fixed chain by one monomer is given by:

$$\mathrm{Cost}(l+1)=\mathrm{Cost}(l)+C_p\times P(l)+k_{l+1}\times P(l)\times \mathrm{Cost}(l_p), \tag{5}$$

where $\mathrm{Cost}(l)$ is the average cost of inserting a chain of length l and is determined by the number of calls to the subroutine determining the energy, $P(l)$ is the probability that a chain of length l is grown, and C_p is the cost of extending the existing feeler by one step. The third term in Eq. (5) refers to the additional cost in computing the weights, where k_{l+1} and l_p denote the number of directions and the probe length assigned to monomer $l+1$. The efficiency of an insertion is then expressed as

$$\mathrm{Eff}(l)=\frac{P(l)}{\mathrm{Cost}(l)}. \tag{6}$$

Also $P(l)$ is given by $P(l+1)=P(l)\times\langle P_{\mathrm{add}}(k_{l+1},l_p)\rangle$ where P_{add} is the probability of adding a monomer. Using the recursive relations for $P(l)$ and $\mathrm{Cost}(l)$ we can arrive at

$$\frac{\mathrm{Eff}(l+1)}{\mathrm{Eff}(l)}=\frac{\langle P_{\mathrm{add}}(k_{l+1},l_p)\rangle}{1+(C_p+k_{l+1}\times \mathrm{Cost}(l_p))\times \mathrm{Eff}(l)}. \tag{7}$$

In practice a number of equilibriated configurations for a polymer system are generated. The first monomer is inserted randomly in the system and the next is added using the recoil growth. The quantities P_{add}, C_p, and $\mathrm{Cost}(l_p)$ are computed for the addition of a third monomer to the chain backbone for different l and k. This procedure gives an initial estimate of the parameters for optimal efficiency with respect to insertion of chains. It does not include dynamical information regarding the efficient sampling of the phase space. The most efficient sampling is achieved when the overlap of distributions of weights for the new and old chains is maximized. This is achieved by choosing as small a value of k as possible consistent with a reasonable construction rate.

[1] *Monte Carlo and Molecular Dynamics simulations in Polymer Science*, edited by K. Binder (Oxford University Press, Oxford, 1995).
[2] K. Kremer and K. Binder, Comput. Phys. Rep. **7**, 259 (1988).
[3] E. Leontidis, B. M. Forrest, A. H. Widmann, and U. W. Suter, J. Chem. Soc., Faraday Trans. **91**, 2355 (1995).
[4] M. Lal and R. F. T. Stepto, J. Polym. Sci., Polym. Symp. **61**, 401 (1977).
[5] J. M. Deutsch, J. Phys. (Paris) **48**, 141 (1987).
[6] A. D. Sokal in Ref. 1.
[7] P. V. Krishna Pant and D. N. Theodorou, Macromolecules **28**, 7224 (1995).
[8] M. L. Mansfield, J. Chem. Phys. **77**, 1554 (1982).
[9] T. Pakula, Polymer **28**, 1293 (1987).
[10] M. N. Rosenbluth and A. W. Rosenbluth, J. Chem. Phys. **23**, 356 (1955).
[11] H. Meirovitch, J. Phys. A **15**, L735 (1982).
[12] H. Meirovitch, J. Chem. Phys. **89**, 2514 (1988).
[13] J. I. Siepmann, Mol. Phys. **70**, 1145 (1990).
[14] J. I. Siepmann and D. Frenkel, Mol. Phys. **75**, 59 (1992).
[15] D. Frenkel and B. Smit, Mol. Phys. **75**, 983 (1992).
[16] D. Frenkel, G. C. A. M. Mooij, and B. Smit, J. Phys.: Condens. Matter **4**, 3053 (1991).
[17] J. Batoulis and K. Kremer, J. Phys. A **21**, 127 (1988).
[18] M. Dijkstra, D. Frenkel, and J.-P. Hansen, J. Chem. Phys. **101**, 3179 (1994).
[19] F. A. Escobedo, and J. J. de Pablo, J. Chem. Phys. **102**, 2636 (1995); M. Vendruscolo, J. Chem. Phys. **106**, 2970 (1997).
[20] Z. Alexandrowicz, J. Chem. Phys. **109**, 5622 (1998).
[21] S. Hemmer and P.C. Hemmer, J. Chem. Phys. **81**, 585 (1984); P. C. Hemmer and S. Hemmer, Phys. Rev. A **34**, 3304 (1986).
[22] D. Frenkel and B. Smit, *Understanding Molecular Simulation* (Academic, Boston, 1996).
[23] J. C. Shelley and G. N. Patey, J. Chem. Phys. **102**, 7656 (1995).
[24] B. Smit, Mol. Phys. **85**, 153 (1995).
[25] G. C. A. M. Mooij, D. Frenkel, and B. Smit, J. Phys.: Condens. Matter **4**, L255 (1992).
[26] F. T. Wall and J. J. Erpenbeck, J. Chem. Phys. **30**, 634 (1959).
[27] P. Grassberger, Phys. Rev. E **56**, 3682 (1997); P. Grassberger, H. Frauenkron, and W. Nadler, cond-mat/9806321.
[28] N. B. Wilding, M. Müller, and K. Binder, J. Chem. Phys. **105**, 802 (1996).
[29] G. C. A. M. Mooij and D. Frenkel, Mol. Simul. **17**, 41 (1996).