

Novel scheme to study structural and thermal properties of continuously deformable molecules

D Frenkel†, G C A M Mooij‡ and B Smitt‡

† FOM Institute for Atomic and Molecular Physics, Kruislaan 407, 1098 SJ Amsterdam, The Netherlands

‡ Koninklijke/Shell Laboratorium Amsterdam (Shell Research BV), PO Box 3003, 1003 AA Amsterdam, The Netherlands

Received 20 September 1991

Abstract. In this article we present a method for calculating the chemical potential of arbitrary chain molecules in a computer simulation. The method is based on a generalization of Siepmann's method for calculating the chemical potential of chain molecules with a finite number of conformations. Next, we show that it is also possible to extend the configurational-bias Monte Carlo scheme developed recently by Siepmann and Frenkel to continuously deformable molecules. The utility of our technique for computing the chemical potential of chain molecules is demonstrated by computing the chemical potential of a fully flexible chain consisting of 10–20 segments in a moderately dense atomic fluid. Under these conditions the conventional particle-insertion schemes fail completely. In addition, we show that our novel configurational-bias Monte Carlo scheme compares favourably with conventional Monte Carlo procedures for chain molecules.

1. Introduction

When two phases are in thermodynamic equilibrium, the pressure and temperature should be the same in both phases. In addition, the chemical potential of any component in the system should be equal in the two co-existing phases.

In a numerical study of phase co-existence, the first two conditions are easy to satisfy because the temperature and pressure can be measured, or even imposed, in a Monte Carlo or molecular dynamics simulation. In principle, the same is true for the chemical potential [2]. But whereas temperature and pressure can be routinely determined at *any* density and for *any* molecular substance, the available techniques to compute (or impose) the chemical potential, have a rather limited range of applicability. To give a specific example, the most widely used technique to measure the chemical potential is based on an expression due to Widom, that relates the excess chemical potential to the change in potential energy of a test particle that is added to the system at a random position. For the canonical (N, V, T) ensemble this expression reads [3]

$$\mu^{\text{ex}} = -\frac{1}{\beta} \ln \langle \exp(-\beta \Delta U) \rangle_{N, V, T} \quad (1)$$

where $\beta = 1/k_B T$ and ΔU denotes the interaction energy of the test particle with all other particles that are present in the system. Although this expression is completely

general, its practical applicability is limited to moderately dense fluids of atoms or simple molecules. To see why this is so, it is instructive to consider a fluid consisting of 'hard core' molecules. In that case, $\langle \exp[-\beta\Delta U] \rangle_{N,V,T}$ can be interpreted as the probability that a test particle, inserted at random in the system, will not overlap with any of the particles present in that system. At high densities, such a successful insertion is unlikely and therefore it is necessary to perform relatively long simulations to obtain reliable statistics. If instead of atoms or simple molecules we consider chain molecules, this problem becomes even worse. Under conditions where it is unlikely to insert even one particle, it will be virtually impossible to insert an entire chain of such particles at a random position in the system.

Recently, Siepmann has introduced a novel technique that greatly increases the efficiency with which the chemical potential of chain molecules *with a finite number of discrete conformations* can be computed [4]. Siepmann's method is based on the Rosenbluth algorithm [5] to generate polymer conformations. In Siepmann's approach, the chemical potential is related to the average of the Rosenbluth weight factor—this will be discussed below in more detail. The importance of this technique is that it leads to much better statistics because the Rosenbluth scheme expresses the insertion probability as an average of many small numbers rather than the average of a large number of 0's and a few 1's.

Subsequently, it was demonstrated that the computation of the chemical potential of self-avoiding chain molecules on a lattice can be made even more efficient by combining it with an algorithm that allows exact enumeration of the conformational entropy of ideal chains on a lattice [6]. However, neither scheme is of much help if one is interested in the calculation of the chemical potential of molecules that do not have a finite number of conformations, such as fully flexible chains or worm-like chains.

In the present paper, we present an efficient scheme to compute the (excess) chemical potential of a flexible chain molecule. This technique can be applied to fully flexible chains and to semi-flexible molecules. We have used this method to compute the excess chemical potential of fully flexible chain molecules consisting of 10–20 segments in an atomic fluid.

An essential step in our method for calculating the chemical potential is the growing of a chain at a random position in the system. We show that the same growth process can be used as a basis for a configurational-bias Monte Carlo (CBMC) procedure. The CBMC method has recently been introduced as a Monte Carlo technique to sample systems consisting of chain molecules with a finite number of conformations [1]. In the present article we generalize the CBMC method to continuum systems. In this procedure, a trial configuration is generated by growing a chain at a random position in the system or by re-growing parts of the chain. We shall show that this Monte Carlo method yields the correct (Boltzmann) distribution of configurations. The present method makes it possible to perform simulations of chain molecules in the Gibbs ensemble [7–9]. We compare our scheme with conventional Monte Carlo algorithms for chain molecules. It turns out that the present method is particularly good at achieving large-scale conformational changes.

2. Chemical potential of flexible molecules

In this section we present our approach to computing the chemical potential of fully flexible chain molecules in an atomic fluid. We start with a discussion of the

theoretical foundation of the method. Next we present results of chemical potential calculations for infinite dilution of chain molecules consisting of 10–20 segments in a moderately dense fluid.

2.1. Theoretical aspects

Although it is our aim to develop a scheme to measure the excess chemical potential of fully flexible molecules, it is most convenient to introduce the method by considering a molecule with a finite number of conformations and then take to the limit in which the number of conformations goes to infinity. In this context, it is instructive to consider first Siepmann's method [4] to measure the chemical potential of a chain molecule with a finite number of conformations. Next the method is generalized to continuously deformable chains and to chains with strong intra-molecular interactions.

2.1.1. Siepmann's method. The probability (P) for generating a polymer with a conformation Γ using the Rosenbluth algorithm [5] is given by

$$P_{\Gamma} = \prod_{i=2}^l \frac{\exp(-\beta u_{\Gamma,i})}{Z_i} \quad (2)$$

where l is the number of segments of the trial conformation and Z_i is the sum of the Boltzmann factors of all b directions of segment i , where b is the number of possible orientations for any polymer segment (e.g. $b = 3$ if the polymer backbone corresponds to a non-reversing random walk on a diamond lattice):

$$Z_i = \sum_{j=1}^b \exp(-\beta u_j) .$$

u_{Γ} denotes the potential energy of segment i of the chain with conformation Γ (note that this energy excludes the contributions of segments $i + 1$ to l , so the total energy of the chain is given by $U_{\Gamma} = \sum_{i=1}^l u_{\Gamma,i}$), j enumerates all possible orientations from which the i th segment of the chain can 'choose'. Finally, u_j denotes the potential energy of the j th trial orientation of segment i . The Rosenbluth weight factor of a chain conformation Γ is defined as

$$W_{\Gamma} \equiv \exp(-\beta u_{\Gamma,1}) \prod_{i=2}^l \frac{Z_i}{b} . \quad (3)$$

We now consider the average of the weight factor W , obtained by using the Rosenbluth scheme to generate a large number of trial conformations of a chain molecule in a system ('bath') consisting of N particles. If we generate a sufficiently large number of such chain conformations we obtain the following average of the weight factor.

$$\langle W_l \rangle \equiv \frac{\sum_{q^N} \langle \sum_{\Gamma} P_{\Gamma}(q^N) W_{\Gamma}(q^N) \rangle_{\Gamma} \exp[-\beta U(q^N)]}{\sum_{q^N} \exp[-\beta U(q^N)]} \quad (4)$$

where the first summation in the numerator runs over all configurations, q^N , of the system and the second summation runs over all conformations Γ of the test polymer.

The angular brackets denote averaging over all starting positions of the test molecule. The set q^N denotes both the centre-of-mass positions and the internal coordinates (if any) of the solvent molecules. $U(q^N)$ is the potential energy of a bath in a configuration q^N . Note that the test polymer does not form a part of the N -particle system. Therefore, the probability of finding the remaining particles in a configuration q^N does not depend on the conformation Γ of the polymer.

For a given configuration of the system q^N , we can define an average Rosenbluth factor

$$\bar{W}_l(q^N) \equiv \sum_{\Gamma} P_{\Gamma}(q^N) W_{\Gamma}(q^N). \quad (5)$$

Substituting (2) and (3) in (5) and dropping the argument q^N yields

$$\bar{W}_l = \sum_{\Gamma} \frac{1}{b^{l-1}} \exp(-\beta U_{\Gamma}). \quad (6)$$

Upon substitution of this expression in (4) we obtain

$$\langle W_l \rangle = \frac{\sum_{q^N} \{ \sum_{\Gamma} \exp[-\beta U_{\Gamma}(q^N)] \} \exp[-\beta U(q^N)]}{\sum_{q^N} b^{l-1} \exp[-\beta U(q^N)]}. \quad (7)$$

We recall that b^{l-1} is simply the number of possible conformations of an ideal (non-interacting, non-self-avoiding) polymer. Therefore, the denominator in (7) is simply equal to the partition function of the N -particle system plus an ideal polymer. We denote this quantity by $Z_N(\text{solvent} + \text{ideal polymer})$. The numerator is equal to the partition function of the N -particle system plus an *interacting* chain molecule, i.e.

$$\langle W_l \rangle = \frac{Z_N(\text{solvent} + \text{polymer})}{Z_N(\text{solvent} + \text{ideal polymer})}. \quad (8)$$

This shows that the average Rosenbluth factor is directly related to the excess chemical potential of the polymer, i.e.

$$-\beta \mu_{\text{ex}} = -\ln \langle W_l \rangle. \quad (9)$$

2.1.2. Fully flexible chains. Up to this point we have basically followed the approach of Siepmann [4] who computed the Rosenbluth weight by enumerating *all* possible orientations for adding a new segment. Clearly, exploring *all* possible orientations of a new segment is no problem if we deal with a molecule that has only a finite number of conformations. However, such an approach would not work at all for a continuously deformable, flexible molecule. Surprisingly, however, it turns out that it is, in fact, possible to construct a rigorous scheme to sample the Rosenbluth weight of any flexible molecule using only a random subset of all possible segment orientations, *even if the total number of orientations is infinite*. We shall now proceed to prove this crucial result. Doing this we keep summing the configurations as if there were a finite number of orientations, but we keep in mind that in the limit $b \rightarrow \infty$ this is not the case. In this limit the probabilities as we give them here become probability

densities and the sum over all configurations, $\sum \Gamma$, is replaced by $\int \dots \int d\Gamma_1 \dots d\Gamma_l$, an integral over the $l(d-1)$ -dimensional space of all polymer conformations.

Let us consider a situation where, instead of generating all b orientations of trial segment i , we only generate a (sub-) set $\{m\}_i$ (see also figure 1 for the notation). Let us assume that the set $\{m\}_i$ contains k elements. We denote the probability of generating a trial segment with the orientation Γ_i by P_{Γ_i} . The probability to generate a specific chain conformation Γ is given by

$$P_{\Gamma} = \prod_{i=2}^l k P_{\Gamma_i} \frac{\exp(-\beta u_{\Gamma_i})}{Z_{\{m\}_i}} \tag{10}$$

where

$$Z_{\{m\}_i} = \sum_{j=1}^k \exp(-\beta u_{m_j}).$$

The Rosenbluth weight factor for the sets of trial orientations $\{m\}_i$ is given by

$$W_{\Gamma} = \exp(-\beta u_{\Gamma_1}) \prod_{i=2}^l \frac{Z_{\{m\}_i}}{k}. \tag{11}$$

If we now average over all possible choices for the sets $\{m\}_i$, and all possible chain conformations, we can compute the average Rosenbluth factor

$$\bar{W}_l = \sum_{\Gamma} \prod_{i=1}^l P_{\Gamma_i} \exp(-\beta u_{\Gamma_i}). \tag{12}$$

We must now specify the probabilities P_{Γ_i} . For the moment, let us consider the case in which all trial conformations are generated with the same *a priori* probability. That implies that every segment has an equal probability to end up in any of the b possible directions and

$$P_{\Gamma_i} = \frac{1}{b}. \tag{13}$$

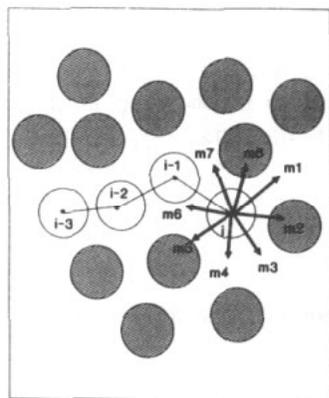


Figure 1. Schematic picture of the growing of the chain.

So we obtain the following expression for the average Rosenbluth factor in (12).

$$\bar{W}_l = \sum_{\Gamma} \prod_{i=1}^l \frac{1}{b} \exp(-\beta u_{\Gamma_i}) \sum_{\Gamma} \frac{1}{b^{l-1}} \exp(-\beta U_{\Gamma}) \quad (14)$$

which is indeed identical to (6). Hence, we have shown that the Rosenbluth weight of an arbitrary flexible molecule can be sampled using only a subset of all permissible segment orientations. Note that for $k = 1$, the method reduces to the conventional Widom insertion method.

2.1.3. Chains with strong intra-molecular interactions. In order to compute the Rosenbluth weight of polymers with strong intra-molecular interactions, for example stiff polymers ('worm-like chains'), we can modify the probabilities P_{Γ_i} in such a way that the probability to generate a given segment orientation is dictated by the Boltzmann factor associated with the intra-molecular energy $u_{\Gamma_i}^{\text{int}}$, i.e.

$$P_{\Gamma_i} = \frac{\exp(-\beta u_{\Gamma_i}^{\text{int}})}{C} \quad (15)$$

where $C = \sum_{j=1}^b \exp(-\beta u_j^{\text{int}})$ is a normalization constant in which the summation runs over *all* b -segment directions. If $b \rightarrow \infty$, the summation is replaced by an integration. This, however, constitutes no special problems, as the resulting integral is simply a normalization constant that need not be evaluated in practice. The probability to generate a polymer in configuration Γ is now given by

$$P_{\Gamma} = \prod_{i=2}^l k \frac{\exp(-\beta u_{\Gamma_i}^{\text{int}})}{C} \frac{\exp(-\beta u_{\Gamma_i}^{\text{ext}})}{Z_{\{m\}_i}} \quad (16)$$

where

$$Z_{\{m\}_i} = \sum_{j=1}^k \exp(-\beta u_{m_j}^{\text{ext}}).$$

The probability for selecting one orientation is now calculated by considering only those interactions which have not been accounted for in u^{int} . We denote these interactions by u^{ext} . These interactions have to be included in the corresponding Rosenbluth factor for the particular set of orientations $\{m\}_i$

$$W_{\Gamma} = \exp(-\beta u_{\Gamma_1}) \prod_{i=2}^l \frac{Z_{\{m\}_i}}{k} \quad (17)$$

The average Rosenbluth factor of a semi-flexible molecule can then be written as

$$\bar{W}_l = \sum_{\Gamma} P_{\Gamma} W_{\Gamma} = \sum_{\Gamma} \frac{1}{C^{l-1}} \prod_{i=1}^l \exp[-\beta(u_{\Gamma_i}^{\text{int}} + u_{\Gamma_i}^{\text{ext}})] \quad (18)$$

This equation is similar to (6) but now the interactions are split up in an internal and an external contribution and C^{l-1} is simply the partition function of a polymer with only internal interactions. If we combine (18) with (4) we obtain an expression for the Rosenbluth weight of a semi-flexible molecule thermally averaged over all configurations of the N -particle system. Equation (8) relates this average to the excess chemical potential of the semi-flexible molecule.

In the next sections we show that the above recipe to compute the excess chemical potential of an arbitrary (semi-) flexible molecule can, in fact, be used under conditions where the conventional particle-insertion schemes fail completely.

2.2. Model and computational details

In order to test the scheme to compute the chemical potential of continuously deformable molecules as described in section 2.1, we have performed a series of computer simulations in which we compute the excess chemical potential at infinite dilution (which is related to the Henry coefficient) of a fully flexible chain molecule in a moderately dense atomic fluid. These systems have been the topic of several studies [10–15].

2.2.1. The model. In our model both the solvent and the polymer segments interact via a Lennard-Jones potential

$$\phi_{i,j}^{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (19)$$

in which r is the distance between particles i and j . In addition to this Lennard-Jones interaction, two adjacent polymer segments are connected with an infinitely stiff spring (bead-rod model) with length σ . In this model the polymer is thus 'dissolved' in its own segments.

In our simulations we have considered the following systems.

(i) A system with repulsive and attractive interactions. For this case the potential was truncated at $R_c = 2.5\sigma$

$$u(r) = \begin{cases} \phi^{\text{LJ}}(r) & r \leq R_c \\ 0 & r > R_c. \end{cases} \quad (20)$$

The contribution to the energy of interactions with $r > R_c$ was estimated analytically, by assuming that $g(r) = 1$ for $r > R_c$ [2].

(ii) A system with only repulsive interactions. In this case the potential was truncated (and shifted) at $R_c = 2^{1/6}\sigma$

$$u(r) = \begin{cases} \phi(r) - \phi(R_c) & r \leq R_c \\ 0 & r > R_c. \end{cases} \quad (21)$$

In addition, we have considered for both systems the limit that the density of the solvent is zero, i.e. a polymer which has no interactions with the solvent but only with its own segments.

Since we are interested in the chemical potential of the polymer at infinite dilution, the polymer has to be considered as a test particle and does not influence the configurations of the solvent.

2.2.2. Computational details. The Monte Carlo simulations were performed at solvent densities $\rho^* = \rho\sigma^3 = 0.5$ and $\rho^* = 0.6$ (and some at $\rho^* = 0.0$) at a temperature $T^* = k_B T/\epsilon = 1.2$. We have performed the simulations in cycles, each cycle consisting of a displacement step and a test polymer insertion step. In the displacement step N_{part} particles are selected at random and given (sequentially) a random displacement such that the average acceptance ratio is approximately 50%. The test particle insertion step consists of N_{ins} attempts to insert a test polymer using the Rosenbluth algorithm.

We have used the following algorithm to generate the conformations of the test polymer.

Step 1. A map of the system is constructed, which consists of cells which are labelled occupied (or free), when the addition of a test particle would result in an overlap (or no overlap) [16]. We define two particles to be overlapping if the distance between these two particles is smaller than R_{min} , where R_{min} is chosen such that for $r < R_{\text{min}}$, $U(r) > U_{\text{max}}$. We have used $U_{\text{max}} = 10\epsilon$.

If a test particle were to be placed in an occupied cell, the energy of this particle would be infinite and hence its contribution to the Rosenbluth factor would be zero.

With this map we can use a very large number of attempts at high density, since an insertion which would be 'rejected' can be detected immediately.

Step 2. N_{ins} attempts to insert a single (test) polymer bead i at a random position in the system. If this attempt is not rejected immediately by the map of step 1, the energy of this particle is calculated and its contribution to the numerator of the Rosenbluth factor is calculated.

Step 3. Growing of the test polymer. If the Rosenbluth factor of the test polymer i with current length l is not equal to zero, a set of k random orientations ($\{m_j\}_i$) are chosen around the last segment of the chain such that

$$|r_{i_l} - r_{m_j}| = \sigma. \quad (22)$$

Of each of these k orientations the energy and the contribution to the Rosenbluth factor are calculated. One of these orientations (Γ_j) is selected with probability

$$P \approx \frac{\exp(-\beta u_{\Gamma_j})}{\sum_{j=1}^k \exp[-\beta u(m_j)]} \quad (23)$$

and the Rosenbluth factor

$$W_l = W_{l-1} \frac{1}{k} \sum_{j=1}^k \exp[-\beta u(m_j)] \quad (24)$$

is updated.

Step 4. Steps 2–4 are repeated until the test polymers reaches the desired length.

It is important to note that in this scheme we only continue to grow those polymers that have a Rosenbluth factor $W \neq 0$. It is, however, essential that those test polymers which have been prevented from growing, and thus have a $W = 0$, are taken into account in the averaging procedure given by (5).

In order to study finite size effects, we have performed simulations for a varying number of solvent particles. Furthermore, we have studied the influence of the periodic boundary conditions on the results. For small systems it can be expected that for large chain lengths, the chains will start to feel their own segments via the periodic boundary conditions. In practice, simulations should always be carried out in systems that are sufficiently large to suppress the spurious effects of periodic boundary conditions. In order to test the influence of this effect we have performed some simulations with periodic boundary conditions for both the solvent and the test polymer and some simulations in which periodic boundary conditions are only used for the solvent and for the interactions of solvent particles with the test polymer but no periodic boundary conditions between polymer segments.

In the algorithm to compute the chemical potential, a choice has to be made for the number of orientations in the set $\{m\}_i$. If we choose $k = 1$, the method reduces to the conventional Widom insertion method and will not give reliable statistics. If k is chosen to be very large, too much time will be spent calculating the Rosenbluth factor of one chain. In order to study the influence of k on the accuracy of the results we have performed simulations for various values of k .

The details of all simulations that we have performed are shown in table 1. In the next section we refer to the number of the run as given in this table.

Table 1. Details of the simulations. NR gives the number of the run, N is the number of solvent particles, ρ^* is the density of the solvent, N_{MC} is the number of Monte Carlo cycles, R_c is the cut-off radius, PBC_p indicates whether periodic boundary conditions for the interactions between test polymer segments are used, k is the number of selected orientations, N_{ins} is the number of attempted insertions per Monte Carlo cycle, and CPU is total amount of CPU time (on the Cray-XMP) in seconds.

NR	N	ρ	$N_{MC}/10^3$	R_c/σ	PBC _p	k	N_{ins}	CPU
1	500	0.6	3	2.5	yes	1	250	610
2	500	0.6	3	2.5	yes	2	250	625
3	500	0.6	3	2.5	yes	10	250	700
4	500	0.6	3	2.5	yes	20	250	900
5	500	0.6	3	2.5	yes	50	250	1825
6	500	0.6	3	2.5	yes	100	250	3525
7	108	0.6	10	2.5	yes	50	250	1750
8	108	0.6	10	2.5	no	50	250	1800
9	500	0.6	3	2.5	yes	50	250	3100
10	0	0.0	1	2.5	no	50	1	50
11	108	0.6	30	1.12	no	50	500	3000
12	500	0.5	4	1.12	no	50	500	3400
13	0	0.0	1	1.12	no	50	1	50

2.3. Results and discussion

We have calculated the excess chemical potential

$$\beta\mu^{\text{ex}}(l) = \beta\mu(l) - \beta\mu(\text{ideal polymer}, l) \quad (25)$$

of a chain with l beads in a solvent of its own segments at infinite dilution. In the previous section it was shown that the efficiency of our method depends on the choice of the number of trial orientations (k). In table 2 the results for the chemical potential are given for various values of k .

Table 2. The excess chemical potential ($\beta\mu^{\text{ex}}$) as a function of the number of beads (l) for various values of k . The number of the simulation refers to table 1. The small subscript gives the accuracy of the results, so -2.43_5 means -2.43 ± 0.05 . '—' indicates that the number of successful insertions was insufficient to obtain a reliable estimate for the chemical potential.

l	$k = 1$	$k = 2$	$k = 10$	$k = 20$	$k = 50$	$k = 100$
NR	1	2	3	4	5	6
1	-2.45 ₅	-2.45 ₈	-2.43 ₄	-2.45 ₆	-2.44 ₄	-2.43 ₅
2	-5.3 ₅	-5.0 ₈	-5.0 ₂	-5.0 ₁	-5.05 ₇	-5.05 ₉
3	-8 ₂	-7 ₁	-7.3 ₃	-7.3 ₂	-7.3 ₂	-7.4 ₁
4	—	—	-9.5 ₄	-9.8 ₄	-9.9 ₂	-9.9 ₂
5	—	—	-11.8 ₇	-12.0 ₇	-12.3 ₃	-12.2 ₃
6	—	—	-13 ₁	-14 ₁	-14.8 ₄	-14.5 ₅
7	—	—	-16 ₂	-17 ₂	-18 ₁	-17 ₁
8	—	—	-18 ₂	-18 ₁	-19.7 ₉	-18.9 ₇
9	—	—	—	-21 ₃	-22 ₂	-22 ₁
10	—	—	—	—	-23 ₁	-24 ₁
11	—	—	—	—	-26 ₂	-27 ₂
12	—	—	—	—	-28 ₂	-28 ₁
13	—	—	—	—	-30 ₂	-28 ₁
14	—	—	—	—	-32 ₂	-29 ₂

For $l = 1$ (one bead) the calculation reduces simply to the determination of the chemical potential of the Lennard-Jones fluid. Our results are in excellent agreement with previous simulations [16–18].

An important result, shown in table 2, is that within the accuracy of our calculations the results are indeed independent of the choice of k . This demonstrates that reliable estimates for the chemical potential can be obtained for *any* value of k , provided the simulations are continued sufficiently long.

This table also demonstrates that with the conventional Widom test particle insertion method, $k = 1$, reliable estimates of the chemical potential can be calculated within a reasonable amount of computer time only for chains with less than four beads. For $l > 3$ this method fails completely. The method discussed in the present paper, however, yields reliable estimates of the chemical potential for much larger chain lengths within a reasonable amount of CPU time.

Not surprisingly, the results suggest that for small chains ($l < 5$) it is not efficient to use a large value of k . Note that in table 2 we have used the same number of Monte Carlo cycles for each simulation. For a detailed comparison of the efficiency of the various values of k , it would be more appropriate to compare the results for the same amount of total CPU time. A rough indication of the accuracy at equal amounts of CPU time can be obtained by multiplying the standard deviation by the square root of the ratio of the total CPU time of the current simulation and the total CPU time of the simulation with $k = 100$. This would, for example, reduce the standard deviations for $k = 1$ by approximately a factor 2.5. For large chains one has to use a large value for k in order to obtain any results at all. Comparison of the results for $k = 50$ with the results for $k = 100$ shows that the accuracy did not improve significantly, while the total CPU time almost doubled. On the basis of this admittedly rather crude comparison, we decided to use $k = 50$ in the rest of our simulations.

In table 3 we give the probability that a chain is inserted 'successfully' in the solvent, provided that the first segment has been inserted successfully. We call an insertion successful if the Rosenbluth factor is different to zero—recall that W was set to zero if one of the segments of the chain had an energy $u_{r_i} > 10\epsilon$. This table illustrates once more that for all but the smallest molecules, it is indeed impossible to insert a chain at random. With the Rosenbluth algorithm, however, this probability can be increased significantly. For example for $k = 50$, during this simulation approximately 270 successful insertions for a chain of 20 segments could be obtained. Of course, it would require more CPU time to obtain a reliable estimate of the chemical potential, but it does demonstrate the significant improvement compared to the random insertion.

Table 3. The acceptance percentage as a function of the number of beads (l) of the polymer for various values of k . The acceptance is expressed as the ratio (in %) of the fraction of accepted configurations and the fraction of accepted monomers ($l = 1$) at the given conditions. The fraction of successful insertion of monomers was 5.5%. The number of the simulation refers to table 1

l	$k = 1$	$k = 2$	$k = 10$	$k = 20$	$k = 50$	$k = 100$
NR	1	2	3	4	5	6
1	100%	100%	100%	100%	100%	100%
3	1.0%	3.3%	29%	49%	74%	92%
6	10^{-5}	10^{-4}	3.2%	12%	35%	47%
10	0	0	0.15%	1.8%	12%	20%
14	0	0	10^{-5}	0.28%	4.1%	8.0%
20	0	0	0	10^{-4}	0.66%	2.0%

It is instructive to compare the random insertion scheme with the method presented in this work, from a computational point of view. If we assume that the probability of a successful insertion of a monomer in the solvent has a probability p_m and that the insertion of the other segments is not influenced by the part of the chain which is already present, then the probability of a successful insertion of a chain with length l is given by

$$P_{\text{random}}(l) = p_m^l. \quad (26)$$

In the scheme of growing a polymer as presented in this work, we try for each new segment k trial orientations. If we assume again that the probability for a successful insertion is not influenced by the segments that are already present and we assume that each trial is an independent attempt [19], the probability of a successful addition of at least one segment out of the k trials is given by

$$P(\text{seg}) = 1 - (1 - p_m)^k. \quad (27)$$

Then the expression for the probability of a successful insertion of a chain with length l becomes

$$P(l) = p_m [1 - (1 - p_m)^k]^{l-1}. \quad (28)$$

If we recall that the probability of a successful insertion of a monomer at $\rho^* = 0.6$ is approximately 5%, then according to (26) a successful insertion of a chain with length 10, using the random insertion method, would have a probability of 10^{-13} . The method described in this work, however, would give a probability (equation (28), with $k \approx 50$) of 2%. We can use (28) to estimate how k should vary with l in order to keep $P(l)$ approximately constant. To a first approximation, we find that the number of trial directions should depend logarithmically on the length of the polymer, i.e. $k \sim \log(l - 1)$.

In figure 2 and table 4 the values of the chemical potential of a polymer in a system with 108 and 500 solvent molecules are given. In figure 2 we compare the results for a system with periodic boundary conditions (PBC) for the solvent and the test polymer (run 7), with a system with PBC only for the solvent-solvent interactions and polymer-solvent interactions (run 8). For small chain lengths ($l < 10$) no significant differences between the two systems are observed. For larger systems ($l > 10$), however, the chemical potential for the polymer with PBC (run 7) tends to increase, while the chemical potential for the polymer without PBC (run 8) continues to decrease.

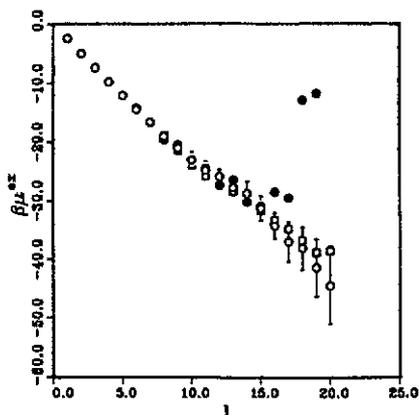


Figure 2. N -dependence of the excess chemical potential ($\beta\mu^*$) as a function of the number of beads of the chain l for system 1. ● (run 7) $N = 108$ and PBC for the solvent and the test polymer, ○ (run 8) $N = 108$ and PBC only for the solvent, and □ (run 9) $N = 500$ and PBC for the solvent and the test polymer.

Table 4. N -dependence of the excess chemical potential ($\beta\mu^{\text{ex}}$). l is the number of beads and N is the number of solvent molecules. The number of the simulation refers to table 1. PBC_p denotes whether periodic boundary conditions between the polymer segments were used. The small subscript gives the accuracy of the results, so -2.43_5 means -2.43 ± 0.05 .

l	$N = 108$	$N = 108$	$N = 500$
NR	7	8	9
PBC _p	yes	no	yes
1	-2.44 ₄	-2.44 ₂	-2.43 ₅
2	-5.0 ₁	-5.02 ₅	-5.01 ₉
3	-7.4 ₂	-7.4 ₁	-7.5 ₁
4	-9.7 ₃	-9.8 ₂	-9.9 ₂
5	-12.1 ₄	-12.1 ₃	-12.2 ₃
6	-14.2 ₅	-14.5 ₆	-14.5 ₅
7	-16.7 ₈	-17 ₂	-17 ₁
8	-20 ₂	-19.1 ₅	-18.9 ₇
9	-21 ₁	-21 ₁	-22 ₁
10	-23 ₃	-23 ₁	-24 ₁
11	-25 ₂	-25 ₁	-26 ₂
12	-27 ₅	-26 ₁	-28 ₂
13	-27 ₅	-28 ₁	-28 ₁
14	-30 ₅	-29 ₂	-29 ₂
15	-31 ₆	-31 ₂	-32 ₃
16	-30 ₇	-34 ₂	-33 ₃
17	-12 ₈	-37 ₃	-35 ₃
18	-12 ₉	-38 ₄	-37 ₃
19	—	-41 ₅	-39 ₄
20	—	-44 ₇	-39 ₅

It is interesting to compare these values for the chemical potential with the results for a system with 500 particles with PBC used for both the solvent and the polymer (figure 2, run 9). Figure 2 shows that the results for the chemical potential of run 9 have no significant differences with run 8 (the system of 108 solvent particles without PBC for the polymer). This suggests that the increase of the chemical potential, as observed in run 7, is due to interactions of the polymer via the periodic boundary conditions with its own segments. This also explains why the total amount of CPU time for run 7 is much smaller than for run 8. Because of the overlap with its own segments via the periodic boundary conditions many more chains have stopped growing in an early stage in run 7.

In figure 3, results for the excess chemical potential as a function of the chain length are presented for system 2 for $\rho^* = 0.5$ and $\rho^* = 0.6$ (run 11 and 12 respectively). This figure shows that the chemical potential increases linearly for chains with $l < 11$, but surprisingly the slope of μ^{ex} suddenly increases as the chain length is increased. For an isolated polymer chain of the same length (run 13), this effect is not observed (see figure 4). This indicates that the change in slope is a solvent effect. This conclusion is supported by the observation that at a higher solvent density (see figure 3) the change of slope occurs for a shorter chain length ($l \approx 8$). Moreover, a similar effect can be observed in system 1 (see figure 2) and also for the isolated chain such an effect is not observed (run 10, see figure 5).

As the excess chemical potential determines the solubility of the polymer in the

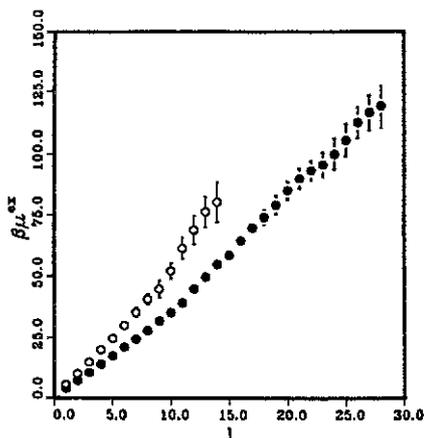


Figure 3. The excess chemical potential, $\beta\mu^{ex}$, of the polymer as a function of the number of beads of the chain l for the system in which the interactions are only repulsive (system 2, runs 11 and 12). The solvent density was \bullet $\rho^* = 0.5$ and \circ $\rho^* = 0.6$.

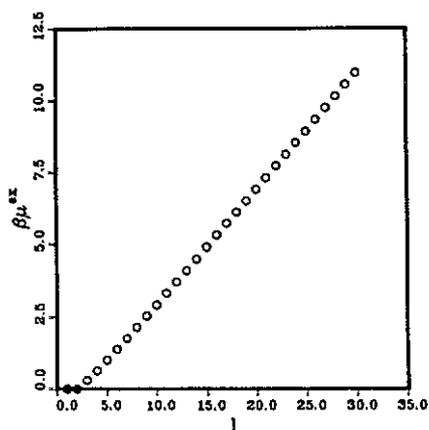


Figure 4. The excess chemical potential, $\beta\mu^{ex}$, of an isolated self-avoiding chain of repulsive Lennard-Jones particles (system 13), as a function of the number of 'atoms' (l) in the chain.

liquid of monomers, our calculations indicate a rather sudden change in the dependence of polymer solubility on chain length. The observed change of slope in μ^{ex} indicates that, beyond a certain length, it becomes more difficult to add additional segments to a chain molecule. As the probability of insertion decreases rapidly with increasing density, the most obvious interpretation is that beyond a certain length, additional segments are inserted in a denser fluid. The sudden change in slope in μ^{ex} suggests that there is a change of the segment density of the polymer in solution. In itself, such an effect is not surprising. What is surprising is that this effect appears to take place quite abruptly even for relatively short chains. At present, a quantitative explanation for this effect is lacking. It would, however, be interesting to see whether recent theories on dilute polymer solutions [20, 21] would reproduce this effect of the

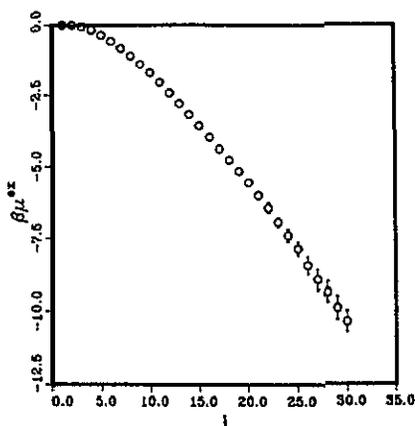


Figure 5. The excess chemical potential, $\beta\mu^{\text{ex}}$, of an isolated self-avoiding chain of Lennard-Jones particles (system 10), as a function of the number of 'atoms' (l) in the chain.

solvent on the excess chemical potential of polymer.

3. Configurational-bias Monte Carlo

In most current Monte Carlo algorithms for chain molecules, trial configurations are generated by, for example, reptation or crankshaft moves [22, 23]. With such local schemes, trial configurations are usually quite similar to the previous configuration. In contrast, a complete re-growing of the chain at a random position in the system corresponds to a very large step in configurational space. Here, we present a configurational-bias Monte Carlo method by which this can be achieved. First some theoretical aspects of this method are discussed. In addition we demonstrate the utility of the method through simulation of the collapse of a polymer in a poor solvent.

3.1. Theoretical aspects

In section 2.1 we outlined a procedure for growing a chain. Although this procedure can be used to measure the excess chemical potential of any flexible chain, it *cannot* be used to generate equilibrium configurations of the chains. As usual, the Rosenbluth scheme introduces a bias in the sampling procedure that has to be corrected by weighting of the generated configurations. This can be achieved by using an extension of the configurational-bias Monte Carlo method of [1]. In the remainder of this section, such a procedure is described.

3.1.1. Configurational-bias Monte Carlo—general aspects. In constructing a Monte Carlo procedure it is advisable to ensure that the sampling scheme satisfies the detailed balance condition. This implies that the rate $K_{(a|b)}$ at which configurations a are transformed into configurations b equals the reverse rate, i.e.

$$K_{(a|b)} = K_{(b|a)}. \quad (29)$$

Let us now compute $K_{(a|b)}$ for an arbitrary Monte Carlo move. We assume that the number of chains in conformation a is given by N_a (and N_b for the density of conformation b), the probability that a chain in conformation b is generated by P_b , and the probability of acceptance $\text{acc}(a|b)$. The rate $K_{(a|b)}$ is then given by

$$K_{(a|b)} = N_a P_b \text{acc}(a|b) \quad (30)$$

and equally for the reverse rate

$$K_{(b|a)} = N_b P_a \text{acc}(b|a). \quad (31)$$

Substitution of (30) and (31) into (29) gives

$$\frac{P_b \text{acc}(a|b)}{P_a \text{acc}(b|a)} = \frac{N_b}{N_a}. \quad (32)$$

If we want to sample a Boltzmann distribution of the configurations, the acceptance rules in (32) must be chosen such that

$$\frac{P_b \text{acc}(a|b)}{P_a \text{acc}(b|a)} = \frac{\exp(-\beta U_b)}{\exp(-\beta U_a)} \quad (33)$$

where U denotes the energy of a configuration. There are many choices that satisfy this condition, an obvious choice being the Metropolis form

$$\text{acc}(a|b) = \min \left(1, \frac{P_a / \exp(-\beta U_a)}{P_b / \exp(-\beta U_b)} \right). \quad (34)$$

Recall that in conventional Monte Carlo algorithms any configuration is chosen with equal probability. For this particular case (34) becomes the conventional Metropolis acceptance rule

$$\text{acc}(a|b) = \min(1, \exp[-\beta \Delta U(a, b)]) \quad (35)$$

where ΔU is the energy differences between configurations b and a .

3.1.2. The Rosenbluth method. If we generate a new conformation by growing a chain using the Rosenbluth algorithm, the probability of generating a chain in a particular configuration b is given by (see also section 2.1.1)

$$P_b = \prod_{i=2}^l \frac{\exp(-\beta u_{b_i})}{Z_i} \quad (36)$$

where u_{b_i} denotes the energy of segment i of a chain in configuration b and $Z_i = \sum_{j=1}^k \exp(-\beta u_j)$. The corresponding Rosenbluth factor is defined by

$$W_b = \exp(-\beta u_{r_1}) \prod_{i=2}^l \frac{Z_i}{k}. \quad (37)$$

This means that as $U_b = \sum_{i=1}^l u_{b_i}$,

$$P_b / \exp(-\beta U_b) = 1/k^{l-1} W_b \quad (38)$$

and the same holds for configuration a . So the acceptance probability given by (34) becomes

$$\text{acc}(a|b) = \min(1, W_b/W_a). \quad (39)$$

This derivation shows that in a Monte Carlo procedure the Rosenbluth method can be used to generate trial configurations of chain molecules with the correct (Boltzmann) distribution *provided* that the acceptance rules are chosen such that (33) is satisfied.

3.1.3. *Fully flexible chain molecules.* In the previous section we have shown that the Rosenbluth method for growing chains can be used to generate trial configurations. In this derivation it is implicitly assumed that the Rosenbluth factor of the old and new configurations are known. This causes no problems in lattice models since all possible configurations can be explored, and therefore the Rosenbluth factor can be calculated exactly. In a continuum model, however, it is impossible to explore all possible configurations. Below we show that a Monte Carlo procedure can be constructed using only a subset of configurations. We first consider chains which are fully flexible. In the next section we discuss chains with strong intra-molecular interactions.

As in section 2.1.2, we consider a situation in which we only generate a subset of all possible orientations. This subset contains k elements. The probability of generating such a set is $P_{\{m\}_i}$. If we use the Rosenbluth method, the probability of generating a chain in configuration b is given by

$$P_b = \prod_{i=2}^l \sum_{\{m\}_i} P_{\{m\}_i} \frac{\exp(-\beta u_{b_i})}{Z_{\{m\}_i}} \quad (40)$$

where $Z_{\{m\}_i} = \sum_{j=1}^k \exp(-\beta u_{m_j})$ and where the prime on the first summation means that this summation runs over all sets of k orientations which include orientation b_i . Detailed balance is guaranteed if we impose a stronger condition, one that we denote by 'super-detailed balance'. By definition, super-detailed balance is satisfied if for *any* specific choice of a set of trial orientations, $\{m\}_i$ and $\{m'\}_i$, $K_{(a|b)}\{m\}_i; \{m'\}_i$ and $K_{(b|a)}\{m\}_i; \{m'\}_i$ are equal, where $\{m'\}_i$ denote the trial directions for configuration a . The probability of generating a chain via a specific set of trial directions $\{m\}_i$ and of choosing a set of trial directions, $\{m'\}_i$, from all possible sets that contain configuration a is given by

$$P_b = \prod_{i=2}^l P_{\{m\}_i} P_{\{\text{rest}'\}_i} \frac{\exp(-\beta u_{b_i})}{Z_{\{m\}_i}} \quad (41)$$

where $\{\text{rest}'\}_i$ denotes the set of $k-1$ orientations $\{m'\}_i$ excluding orientation a_i . Actually, P_b in (41) denotes the probability to generate a *new* trial conformation b from the set $\{m\}_i$ and a set of trial orientations $\{\text{rest}'\}_i$ around the old conformation a . As we shall see below, such an extended definition of a trial move is necessary in order to impose super-detailed balance. The Rosenbluth factor that corresponds to configuration b and the set of orientations $\{m\}_i$ is

$$W_b = \exp(-\beta u_{\Gamma_{b_1}}) \prod_{i=2}^l \frac{Z_{\{m\}_i}}{k} \quad (42)$$

whereas the Rosenbluth factor of configuration a with orientations $\{m'\}_i$ will be

$$W_a = \exp(-\beta u_{\Gamma_{a_1}}) \prod_{i=2}^l \frac{Z_{\{m'\}_i}}{k} \quad (43)$$

If we substitute (41) for both configurations a and b in (34), we see that we have to use (39) for the acceptance probability.

A convenient way to choose a set $\{m'\}_i$ that contains the orientation a_i is to place the same set of additional directions as has been used for generating a new trial configuration around the original conformation a_i .

3.1.4. *Chains with strong intra-molecular interactions.* In some applications chains with strong intra-molecular interactions are used (worm-like chains, chains with flexible joints, or chain molecules with bending and torsion potentials). For these types of chain molecules it is favourable to separate the intra-molecular interactions u^{int} from the inter-molecular interactions u^{ext} . A similar procedure is used for the calculation of the chemical potential of these molecules (see also section 2.1.3).

For these types of systems we generate a set of k trial orientations $\{m\}_i$ with a probability $P_{\{m\}_i}$, prescribed by the Boltzmann factor associated with the intra-molecular energy (see also (15))

$$P_{\{m\}_i} = \frac{1}{C^k} \prod_{j=1}^k \exp(-\beta u_{m_j}^{\text{int}}). \quad (44)$$

where $C = \sum_{j=1}^b \exp(-\beta u_j^{\text{int}})$ is a normalization constant. If we impose super-detailed balance as described in the previous section, we have to consider the probability of generating a chain via one particular choice of trial orientations, $\{m\}_i$, and of choosing one set, $\{m'\}_i$, out of all possible sets of trial orientations that contain orientation a_i ,

$$P_b = \prod_{i=2}^l P_{\{m\}_i} P_{\{\text{rest}'\}_i} \frac{\exp(-\beta u_{b_i}^{\text{ext}})}{Z_{\{m\}_i}} \quad (45)$$

where $Z_{\{m\}_i} = \sum_{j=1}^k \exp(-\beta u_{m_j}^{\text{ext}})$ and where $\{\text{rest}'\}_i$ denotes the set of $k-1$ orientations $\{m'\}_i$ excluding orientation a_i . The corresponding Rosenbluth factor is

$$W_b = \exp(-\beta u_{\Gamma_{b_i}}) \prod_{i=2}^l \frac{Z_{\{m\}_i}}{k}. \quad (46)$$

Now as $U_b = \sum_{i=1}^l u_{b_i}^{\text{int}} + u_{b_i}^{\text{ext}}$ we get

$$P_b / \exp(-\beta U_b) = P_{\{\text{rest}\}_i} P_{\{\text{rest}'\}_i} / C k^{l-1} W_b \quad (47)$$

where $\{\text{rest}\}_i$ denotes the set of $k-1$ orientations $\{m\}_i$ excluding orientation b_i . We can repeat this for configuration a generated with a set of k trial orientations, $\{m'\}_i$. The convenient choice for $\{m'\}_i$ that was mentioned in section 3.1.3, means that $P_{\{\text{rest}\}_i} = P_{\{\text{rest}'\}_i}$. Substitution of (47) into (34) again gives (39) for the acceptance probability.

In the next section, we report numerical tests of the CBMC method for such molecules.

3.2. Model and computational details

As an example of an application of the configurational-bias Monte Carlo method, which has been described in section 3.1, we equilibrate a chain molecule, initially prepared in an elongated conformation, in a poor solvent. However, the method is not restricted to dilute solutions and can be applied to the study of dense polymer systems as well. We compare this method with a more conventional MC scheme,

namely the reptation algorithm [22, 23]. Alternatively, we could have compared our scheme with the pivot algorithm [24] which has been shown to be ergodic [25]. However, for dense solutions, the acceptance of trial moves of the pivot algorithm is very low and the comparison would be unfair. If we had performed simulations of an isolated chain, we should expect the pivot algorithm to be more efficient than the reptation scheme. In that case a comparison of the CBMC scheme with the pivot algorithm would have been more natural.

3.2.1. The model. Again we considered a system in which both the solvent and the polymer segments interact via the Lennard-Jones potential given by (19), which was truncated at $R_c = 2.5\sigma$ and shifted. And again the polymer segments were attached by freely rotating, infinitely stiff springs of length σ . To simulate a polymer in a poor solvent, we took the strength ϵ of the interactions to be more attractive for polymer-polymer interactions ($\epsilon_{pp} = 1.0$) than for polymer-solvent ($\epsilon_{ps} = 0.1$) interactions. The solvent-solvent interactions were the same as those describing the bead-bead interaction in the polymer ($\epsilon_{ss} = 1.0$).

3.2.2. Computational details. The computations were performed at a solvent density $\rho^* = 0.23$ and at a temperature $T^* = 1.0$. The polymer was 100 segments long. At first the polymer was prepared in a fully stretched configuration and the Lennard-Jones fluid was equilibrated around it. Then we performed a large number of Monte Carlo moves of the polymer. Here we compare the results of two different algorithms.

First we consider the Configurational-bias Monte Carlo (CBMC). To generate a new polymer configuration and to compute its Rosenbluth factor, we used the same steps 2-4 as described in section 2.2.2. To implement the acceptance criterion given by (39), we also require the Rosenbluth factor of the old configuration in the possibly new environment. This Rosenbluth factor can be computed in exactly the same way, except that there are some restrictions on the choice of the set of k trial orientations for selecting the i th bond in step 3. Obviously one of these orientations has to be the actual orientation of the i th bond of the old chain. As already mentioned in section 3.1.4 a convenient choice is to take the same set $\{m\}_i$, already used for selecting the i th bond of the new chain, and to replace the selected orientation by the orientation of the i th bond of the old chain. Having computed the old and the new Rosenbluth factor we accepted the new configuration with a probability given by (39). In this way we can regrow the whole chain at once, but it is also possible to cut the chain at a random position and to regrow only part of the chain, or to regrow it from the other end. This is very important for simulations of long chains, where complete regrowing can have a low acceptance rate.

A number of simulation parameters can be varied to optimize the CBMC simulation. For instance we can vary the function $P(\Delta l)$ that determines the probability with which we attempted to regrow Δl segments of the chain. Moreover, we can vary the number of trial directions, k . An optimal choice for k exists because for low k values the acceptance is low while for large k the amount of computer time increases. The following simple argument provides us with a rough estimate of the optimal choice of k for a given length of chain, Δl , that has to be regrown. Using the reptation algorithm as described below one needs of the order of l^2 accepted moves to get a completely new configuration and using CBMC this number is, for a fixed Δl , $l^2/\Delta l^2$. The amount of CPU time for one move is proportional to $1/\text{acc}(\text{REP})$ for the reptation and to $k\Delta l/\text{acc}(\text{CBMC})$ for the CBMC, where $\text{acc}(\text{REP})$ and $\text{acc}(\text{CBMC})$

are respectively the fraction of accepted moves during the reptation and the CBMC runs. So the CPU time needed to get a new configuration is, for reptation,

$$l^2 / \text{acc}(\text{REP}) \quad (48)$$

and for the CBMC,

$$\frac{l^2 k}{\Delta l \text{acc}(\text{CBMC})} \quad (49)$$

This means that for the choice of k we should expect CBMC to be more efficient than reptation if

$$k < \Delta l \frac{\text{acc}(\text{CBMC})}{\text{acc}(\text{REP})} \quad (50)$$

We have tried several choices for k as a function of Δl . In the latter scheme, a trial move involves removing one segment from one end of the chain (chosen at random) and adding it to the other end with a random orientation.

As mentioned above, we compare the CBMC method with the well known reptation algorithm [22, 23]. Using this algorithm means just cutting off one segment on one side of the chain and trying to add it onto the other side with a random orientation. The Boltzmann factors of the old and the new positions determine the acceptance probability.

3.3. Results and discussion

In order to obtain a quantitative measure of the efficiency of the configurational-bias Monte Carlo simulations described in the previous section, we studied the rate of equilibration of two quite different properties of the chain, namely the radius of gyration R_G , defined as

$$R_G = \sqrt{\frac{1}{l^2} \sum_{1 \leq i < j \leq l} (r_i - r_j)^2} \quad (51)$$

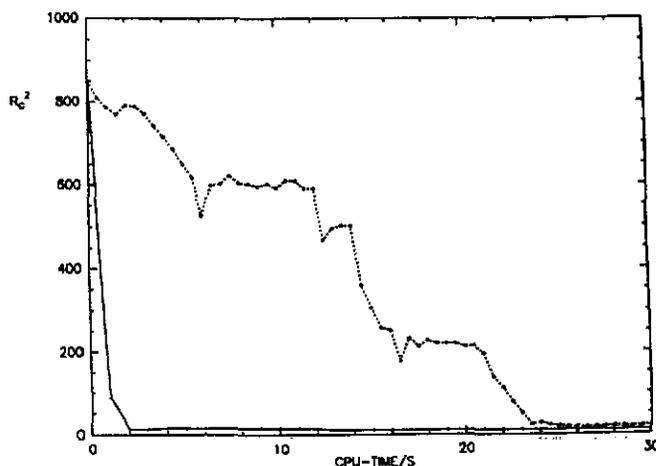


Figure 6. The radius of gyration, R_G , of the chain as a function of the CPU time for both the configurational-bias Monte Carlo (---) and the reptation algorithm (.....).

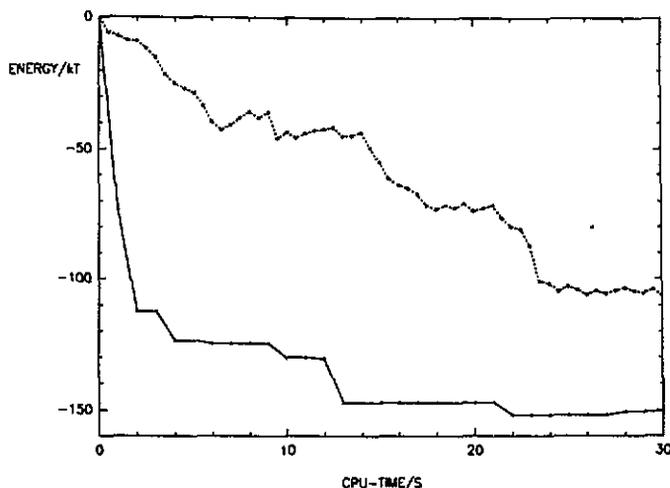


Figure 7. The energy, E , of the chain as a function of the CPU time for both the configurational-bias Monte Carlo (---) and the reptation algorithm (.....).

where r_i is the position of the i th segment, and the internal energy, E , of the chain. We also computed the 'time evolution' of the same quantities using the reptation algorithm. In order to facilitate the comparison we plotted the actual CPU time used by both schemes. Figure 6 shows the 'equilibration' of R_G^2 and figure 7 shows the same for E . Both the results obtained by the configurational-bias Monte Carlo and by the reptation algorithm are presented. What we show is the initial part of the simulation where the largest conformational changes take place. As can be seen from figure 6, the CBMC is very efficient at achieving such large conformational changes: the radius of gyration relaxes about ten times faster to its final value than during the reptation run. In this part of the simulation the energy also equilibrates much faster using CBMC. After the initial chain collapse, the internal energy keeps decreasing slowly, while the radius of gyration fluctuates around its equilibrium value. The time that it takes to achieve total equilibration of E is longer than shown in figure 7. During this final approach to equilibrium, the chain undergoes mainly small conformational changes and, under these circumstances, the reptation algorithm is about as efficient as the CBMC scheme. We note, however, that there are many situations where the reptation algorithm, unlike the CBMC scheme, cannot be used at all (e.g. grafted chains, branched polymers).

In our simulation we have the freedom to choose the probability $P(\Delta l)$ with which we attempt to regrow a piece of chain of Δl segments and the number of trial orientations k at each step. The simple argument given in section 3.2.2 suggests how to choose those parameters. From (49) it follows that Δl should be large (to make large configurational changes) while the acceptance of the moves should not get too low. A higher value of k would lead to a larger acceptance but would, at the same time, increase the computational costs. In any case, k should not be increased above the limit given in (50) because for larger k values the present scheme cannot compete with the simple reptation algorithm. We took $P(\Delta l)$ to be inversely proportional to $\Delta l^{2/3}$ which means that 50% of the trial moves were attempts to regrow a piece of chain shorter than 13 segments. For k we took a minimum number of 4 and let it increase logarithmically with Δl , which is a guess derived from the handwaving

arguments given in section 2.3. Our choices for k and Δl are almost certainly not optimal. However, it is difficult to make general statements about the optimal values of k and Δl , as these will depend on the system under consideration. A plot of the limiting value of k , given by (50), averaged during the run, is shown in figure 8. This value decreased because the acceptance of the moves of large Δl decreased. The average value of k is also shown in the plot. Here again we see that, especially when large configurational changes take place, CBMC is advantageous, because k stays well below its limiting value.

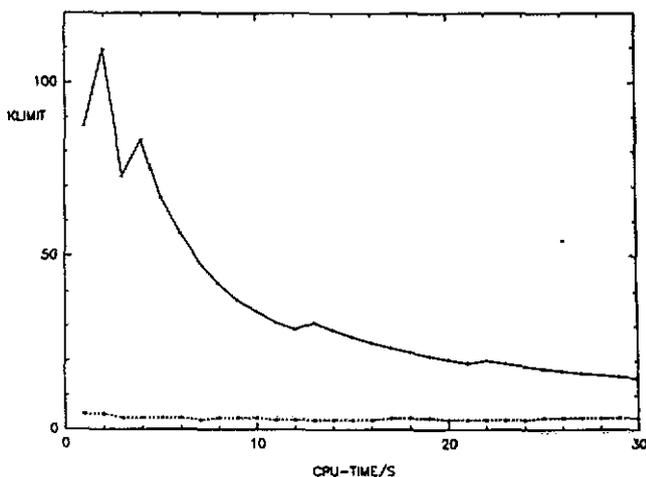


Figure 8. The limiting value of k as given by (50) (---) and the average value of k (.....) as a function of the CPU time.

These results indicate that CBMC is a useful method to achieve large configurational changes. When only minor re-arrangements are needed, a small value of k and Δl becomes more advantageous. Of course, the conventional reptation limit corresponds to the limit $k = \Delta l = 1$.

4. Conclusions

In this paper we have presented a method for calculating the chemical potential of arbitrary chain molecules. Whereas the conventional Widom test particle insertion method gives reliable results for the chemical potential for chains up to 3 beads inserted in a moderately dense atomic fluid ($\rho^* = 0.5-0.6$), our method yields accurate results for chains consisting of 10-30 beads.

An alternative method for calculating the chemical potential for systems containing polymers has recently been proposed by Kumar *et al* [26], in which the chemical potential is estimated by calculating the change in free energy caused by the addition of an extra segment to the polymer. The chemical potential is obtained by assuming that this free energy is representative for all segments. The fact that we observe a non-linear length dependence of the chemical potential of a chain molecule strongly suggests that the assumption underlying the scheme of [26] may be too naive.

In the second part of this article we showed that the novel 'configurational-bias' Monte Carlo sampling scheme of Siepmann and Frenkel [1] can be extended

rigorously to continuously deformable molecules. In this CBMC scheme, we make use of the same Rosenbluth sampling that was used for the chemical potential calculations. A comparison is made with the reptation algorithm for the collapse of a polymer chain in a poor solvent. The CBMC is shown to be advantageous especially when large configurational changes are needed.

Furthermore, our results make it possible to combine the Gibbs ensemble technique [7–9] with the configurational-bias Monte Carlo procedure [27]. This would be a significant extension of the range of applicability of the Gibbs ensemble technique, since at present this ensemble cannot be used for calculating phase diagrams involving chain molecules.

Finally, we note that the configurational-bias Monte Carlo has been presented exclusively as a method to generate *polymer* conformations. In fact, the method is more general than that. It can be used as a scheme to perform collective rearrangements of any set of labelled coordinates. Such a scheme could, for instance, be used to carry out Monte Carlo moves to swap n small particles within a volume ΔV with one large particle that occupies the same (excluded) volume. Similarly, one could envisage a Monte Carlo move in which part of a polymer *plus the surrounding solvent* is regrown.

Acknowledgments

The authors would like to thank A Z Panagiotopoulos, I Szeifer and B Widom for useful suggestions and comments on this work. We thank A Alavi for a critical reading of the manuscript. The investigations reported in this paper were supported in part by 'Scheikundig Onderzoek Nederland' (SON) with financial aid from 'Nederlandse Organisatie voor Wetenschappelijk Onderzoek' (NWO). The work of the FOM Institute is part of the research program of FOM and is supported by 'Nederlandse Organisatie voor Wetenschappelijk Onderzoek' (NWO).

References

- [1] Siepmann J I and Frenkel D 1992 *Mol. Phys.* **75** 59
A simple version of the same scheme was presented earlier by Harris J and Rice S A 1988 *J. Chem. Phys.* **88** 1298
- [2] Allen M P and Tildesley D J 1987 *Computer Simulations of Liquids* (Oxford: Clarendon)
- [3] Widom B 1963 *J. Chem. Phys.* **39** 2802
- [4] Siepmann J I 1963 *Mol. Phys.* **39** 2802
- [5] Rosenbluth M N and Rosenbluth A W 1955 *J. Chem. Phys.* **23** 356
- [6] Mooij G C A M and Frenkel D 1991 *Mol. Phys.* **74** 41
- [7] Panagiotopoulos A Z 1987 *Mol. Phys.* **61** 831
- [8] Panagiotopoulos A Z, Quirke N, Stapleton M and Tildesley D J 1988 *Mol. Phys.* **63** 527
- [9] Smit B, de Smedt Ph and Frenkel D 1989 *Mol. Phys.* **68** 931
- [10] Rapaport D C 1979 *J. Chem. Phys.* **71** 3299
- [11] Webman I, Lebowitz J L and Kalos M H 1980 *Phys. Rev. B* **21** 5540
- [12] Bruns W and Bansal R 1981 *J. Chem. Phys.* **74** 2064
- [13] Smit B, van der Put A, Peeters C J, de Swaan Aronds J and Michels J P J 1988 *Chem. Phys. Lett.* **144** 555
- [14] Smit B, Cox, K R and Michels J P J 1989 *Mol. Phys.* **66** 97
- [15] Luque J, Sanamaria J and Freire J J 1989 *J. Chem. Phys.* **91** 584
- [16] Deitrick G L, Scriven L E and Davies H T 1989 *J. Chem. Phys.* **90** 2370

- [17] Heinbruch U and Fischer J 1987 *Mol. Simul.* **1** 109
- [18] Smit B and Frenkel D 1989 *J. Phys.: Condens. Matter* **1** 8659
- [19] For a large number of attempts many orientations will be very close to each other, and are therefore not really independent. Therefore the above argument will not be valid for a very large number of attempts
- [20] Szeifer I and Widom B 1989 *J. Chem. Phys.* **90** 7524
- [21] Szeifer I 1990 *J. Chem. Phys.* **92** 6940
- [22] Kron A K 1965 *Polymer Sci. USSR* **7** 1361
- [23] Wall F T and Mandel F 1975 *J. Chem. Phys.* **63** 4592
- [24] Lal M 1969 *Mol. Phys.* **17** 57
- [25] Madras N and Sokal A D 1988 *J. Stat. Phys.* **50** 109
- [26] Kumar S K, Szeifer I and Panagiotopolous A Z 1991 *Phys. Rev. Lett.* **66** 2935
- [27] Mooij G C A M, Frenkel D and Smit B in preparation