Monte Carlo simulations of off-lattice polymers

J. Phys.: Condens. Matter 7 (1995) 3089–3097. Printed in the UK

Monte Carlo simulations of off-lattice polymers

Peter Grassberger and Rainer Hegger
Physics Department, University of Wuppertal, 42097 Wuppertal, Germany

Received 28 November 1994, in final form 21 February 1995

Abstract. We point out that a newly introduced recursive algorithm for lattice polymers has a much wider range of applicability. In particular, we apply it to the simulation of off-lattice polymers with Lennard-Jones potentials between non-bonded monomers and either delta or harmonic potentials between bonded monomers. Our algorithm allows particularly easy calculations of the free energy, and seems in general more efficient than other existing algorithms.

1. Introduction

In a number of recent papers, new Monte Carlo schemes for simulating off-lattice polymers have been proposed [1–8]. A particular aspect there was the calculation of chemical potentials. This is not easy in most schemes, which has led to some controversy [9, 10]. In this paper we want to point out that a recently introduced recursive implementation of the enrichment method for lattice polymers [11] can be easily adapted to this problem. It is both easy to implement and efficient, and the computation of the chemical potential is straightforward. Indeed, for polymers without long-range monomer–monomer interactions and without interactions with any solvent our method seems to be faster than all methods mentioned above. It is *not* efficient for systems at extremely low energies (when Boltzmann factors due to attractive potentials between individual monomers become >10), and for systems with long-range forces [12, 13].

Though we shall apply our algorithm only to polymer systems, we should point out that it is much more general. It can be applied to any equilibrium system which can be broken up into discrete units, labelled by an index $i = 1, \ldots, N$, and whose internal energy can be written as

$$U = \sum_{i=1}^{N} U_i \tag{1}$$

where U_i depends *only* on units with label $i' \leqslant i$. In the case of a polymer with potential u_{ij} *between non-bonded monomers and potential* v_i *between monomers* i *and* $i-1$, we choose of course $U_i = v_i + \sum_{j<i} u_{ij}$ for $i > 1$, while $U_1 = v_1$ is evaluated with $x_0 = 0$. Thus the start of the polymer chain is anchored to $x = 0$.

The algorithm basically tries to build the system by assembling it unit by unit. To obtain the correct Boltzmann weights, the entire assembled configuration has to be discarded occasionally (with probabilities depending on these weights) if the following units do not fit. A typical example is a self-avoiding walk on a lattice, where the configuration has to be discarded if the next step would lead to a self-intersection. In order to compensate for this ‘attrition’, we use an ‘enrichment’ method, the basic idea of which was already

0953-8984/95/163089+09$19.50 © 1995 IOP Publishing Ltd

introduced more than 30 years ago [14]. Instead of trying just one continuation from a partially assembled system, the intermediate sample is 'enriched' by replicas which serve as starting points for independent continuations. Thus from each partly assembled system more than one continuation is attempted. These attempts are made irrespective of whether one of them is successful or not, which leads immediately to an unbiased Gibbs ensemble and which distinguishes the method from most other proposals to overcome attrition.

But in contrast to earlier implementations of the enrichment idea which were of 'breadth first' type [15, 16, 17, 18], our implementation is 'depth first' [19]. This implies completely different data structures. In particular, it means that a partly assembled system can reside in the fast memory of the computer even in very large simulations. This avoids the very time-consuming data transfer needed in breadth first implementations, unless the systems are very small. As discussed in [17], simulating a system with N units one has to simulate $\gg N$ ensemble members simultaneously in a breadth first algorithm. This requires a storage space $> \mathcal{O}(N^2)$, which limits severely the system sizes. But even if the entire ensemble fits into the memory, a breadth first algorithm is slower by a factor $\mathcal{O}(N)$ since adding a new unit takes a time $\mathcal{O}(N)$ [17] (there is a finite probability that a new replica has to be created) while it needs only a time $\mathcal{O}(1)$ for a depth first algorithm. Also, the simplest and most intuitive implementation of a depth first algorithm is via recursive function calls. In this case the compiler performs all the book-keeping which is fast but quite non-trivial in this approach. Thus our method avoids all problems which have made the enrichment method unpopular in the past.

The only disadvantage of a depth first implementation is that we have to guess the attrition in advance. In a breadth first approach, level i is treated only after all previous levels have been finished, and thus the attrition on the previous levels is known. If we know the amount of attrition sufficiently well from other sources, we do not have any problem in a depth first approach either. Otherwise, the best strategy is to start with small systems and a conservative estimate of the attrition, and to increase the system size in separate runs as the attrition gets better and better known [20].

2. Algorithm

Our aim is to compute the partition function

$$Z_N = \int \mathrm{d}\boldsymbol{x}_1 \ldots \mathrm{d}\boldsymbol{x}_N \, \mathrm{e}^{-\beta U(\boldsymbol{x}_1 \ldots \boldsymbol{x}_N)} \qquad \beta = (k_\mathrm{B} T)^{-1}. \tag{2}$$

In addition, we consider also 'partial partition functions' which describe the last $N-i$ units in the static background field created by the first $i+1$ units,

$$Z_{N-i|i}(\boldsymbol{x}_1 \ldots \boldsymbol{x}_i) = \int \mathrm{d}\boldsymbol{x}_{i+1} \ldots \mathrm{d}\boldsymbol{x}_N \, \mathrm{e}^{-\beta \sum_{j=i+1}^{N} U_j(\boldsymbol{x}_1 \ldots \boldsymbol{x}_j)}. \tag{3}$$

They can be written recursively,

$$Z_{N-i+1|i-1}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}) = \int \mathrm{d}\boldsymbol{x}_i \, \mathrm{e}^{-\beta U_i(\boldsymbol{x}_1 \ldots \boldsymbol{x}_i)} \, Z_{N-i|i}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i) \tag{4}$$

with $Z_{0|N}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \equiv 1$ and $Z_{N|0} \equiv Z_N$. The basic strategy will be to compute Monte Carlo estimates for the partial partition functions using this recursion relation (this will be done implicitly, and the explicit code needed to do it will be very compact). The total partition function is generated by 'assembling' the last units first (which means just summing over suitable statistical samples), and working one's way back. The task is completed when finally the sample points for the first unit are summed over. With this strategy in mind we will in the following concentrate on one typical step in the recursion relation where $Z_{N-i|i}(\boldsymbol{x}_1 \dots \boldsymbol{x}_i)$ is assumed to be known, and $Z_{N-i+1|i-1}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1})$ is to be computed.

We assume that the potential U_i can be split into two parts,

$$U_i(\boldsymbol{x}_1 \dots \boldsymbol{x}_i) = U_i^{(0)}(\boldsymbol{x}_1 \dots \boldsymbol{x}_i) + \Delta U_i(\boldsymbol{x}_1 \dots \boldsymbol{x}_i) \tag{5}$$

with the following properties:

(i) the partial partition functions $Z^{(0)}_{N-i|i}$ associated with $U^{(0)}$ can be computed for each background configuration $(\boldsymbol{x}_1 \dots \boldsymbol{x}_i)$ either analytically or by some other method which is independent of the present Monte Carlo method;

(ii) a fast (pseudo-)random number generator exists which produces points $\boldsymbol{\xi}_k$ distributed with the density

$$\rho_i^{(0)}(\boldsymbol{\xi}|\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}) = \frac{Z^{(0)}_{N-i|i}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi})}{Z^{(0)}_{N-i+1|i-1}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1})} \, \mathrm{e}^{-\beta U_i^{(0)}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi})} \tag{6}$$

(notice that this is normalized due to (4));

(iii) ΔU_i is 'kind' in the sense that the integral $\int \mathrm{d}\boldsymbol{\xi}\, \rho_i^{(0)} \mathrm{e}^{-\beta \Delta U_i}$ converges for all background configurations and is not too big.

We should stress that the last condition affects only the efficiency of the method, but is not related to any bias. In particular, we make no series expansion or truncation in higher powers of ΔU_i, or anything of that sort. In general, it will be sufficient if $U_i^{(0)}$ has correct asymptotic behaviour for the integral to converge, and has roughly the same shape as U_i.

Our $U^{(0)}$ is similar to the 'guiding field' in [16, 17].

Using this decomposition of U_i one shows easily that

$$\begin{aligned} Z_{N-i+1|i-1}&(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}) \\ &= Z^{(0)}_{N-i+1|i-1}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}) \int \mathrm{d}\boldsymbol{\xi}\, \rho_i^{(0)}(\boldsymbol{\xi}|\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}) \\ &\quad \times \mathrm{e}^{-\beta \Delta U_i(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi})} \frac{Z_{N-i|i}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi})}{Z^{(0)}_{N-i|i}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi})}. \end{aligned} \tag{7}$$

The integral over $\boldsymbol{\xi}$ can now be approximated by a sum over random points $\boldsymbol{\xi}_k$ obtained by means of the above random number generator, and we obtain

$$\begin{aligned} Z_{N-i+1|i-1}&(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}) \\ &= Z^{(0)}_{N-i+1|i-1} \lim_{M\to\infty} \frac{1}{M} \sum_{k=1}^{M} \mathrm{e}^{-\beta \Delta U_i(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi}_k)} \frac{Z_{N-i|i}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi}_k)}{Z^{(0)}_{N-i|i}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi}_k)}. \end{aligned} \tag{8}$$

Assume that we have already an estimator for $Z_{N-i|i}$. Then an estimator for $Z_{N-i+1|i-1}$ is obtained by either associating a weight

$$w_i(\boldsymbol{\xi}_k; \boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}) = \frac{1}{M} \frac{Z^{(0)}_{N-i+1|i-1}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1})}{Z^{(0)}_{N-i|i}(\boldsymbol{x}_1 \dots \boldsymbol{x}_{i-1}, \boldsymbol{\xi}_k)} \, \mathrm{e}^{-\beta \Delta U_i(\boldsymbol{x}_1 \dots \boldsymbol{x}_i, \boldsymbol{\xi}_k)} \tag{9}$$

with each $\boldsymbol{\xi}_k$, or—and this is the method used in our approach—by replacing each $\boldsymbol{\xi}_k$ in the average by $p_i w_i$ replicas of itself, each counted with unit weight and labelled by an index α. Here p_i is a parameter (independent of $\boldsymbol{x}_i$ and $\boldsymbol{\xi}_k$) which is in principle arbitrary (more about it will be said below) and which controls the size of the sample by counterbalancing the attrition† during the step $i \to i-1$. This gives then our MC estimate

$$Z_{N-i+1|i-1}(x_1 \dots x_{i-1}) \approx \frac{1}{p_i} \sum_{k=1}^{M} \sum_{\text{replicas } \alpha} Z_{N-i|i}^{[\alpha]}(x_1 \dots x_{i-1}, \xi_k). \tag{10}$$

The superscript α on the partial partition function on the RHS is to indicate that we use of course different sample points $(x_{i+1} \dots x_N)$ for each replica, even though the backgrounds are the same.

The parameter p_i has to be chosen carefully: large p_i will lead to samples whose sizes increase quickly with i, leading to excessive CPU times for large N, while small p_i lead to too small samples for large i. Optimally, p_i should be chosen such that the sample size is roughly independent of i. Notice that this means in particular that M will not be large. Indeed, most numerical results quoted below are obtained with $M = 1$. Large statistics are not obtained by trying many different continuations of each partially built chain, but by making many independent runs. It is only for extremely low temperatures (not studied here) that $M \gg 1$ should be advantageous since it allows a more uniform covering of configuration space.

We should point out that the above two possibilities (of using w_i either as a weight or as a multiplicity of replicas) are indeed just two special cases within a much wider range of possible choices. They are all distinguished by a different balance between equidistribution of the statistical sample and equal weights put on all sample points. It might well be that for different purposes different variants are optimal, but we shall in the following discuss only the choice of uniform weights, corresponding to perfect importance sampling. It has the advantage that all thermal averages are just normal averages without any additional weight factors except for the weights p_i. In particular, the incremental chemical potential is simply

$$\mu_i = -k_B T \log \frac{Z_i}{Z_{i-1}} \approx -k_B T \log \frac{m_i}{m_{i-1} p_i} \tag{11}$$

where m_i is the total number of sample point replicas at level i (i.e., the total number of subroutine calls at depth i in the algorithm described below).

Technically, our method is implemented by means of a recursively called subroutine which has the level i as argument. Basically, it just chooses a random point ξ, inserts a monomer at this point and computes its weight $w_i(\xi)$, and makes on average $p_i w_i(\xi)$ calls to itself if $i < N$, with new argument $i+1$. After returning from all these subroutine calls, the monomer at ξ is removed and the subroutine is left. This is of course complemented by updating the statistics for whatever observable is to be measured.

Finally, we have to specify what we mean by 'make... calls *on average*'. In principle, we can choose any distribution for the number of calls, provided it gives the right average. But efficiency will depend on this choice. One possibility would be e.g. a Poissonian (this would be in spirit with the first application of this method in a lattice model [20]), but in general this is not the best choice. As in lattice models, it seems that the optimal choice depends on the specific situation, and in extreme cases (steep potentials and low temperatures) some experimentation will be needed. As a rule of thumb we propose to choose the distribution such that it has the smallest possible variance [11]. Thus, if $p_i w_i(\xi) = k + \eta$ with $\eta \in [0, 1)$ and k integer, we first make k calls and then add one more call with probability η.

† We use the word 'attrition' for conformity with the use in the literature on self-avoiding walks. It should be noted that in our case it does not necessarily imply a depletion of chains, but can also imply the opposite, depending on the sign of the potential.

The N dependence of the required CPU time depends on several details. For high temperatures and weak effects of self-avoidance the end of the chain makes essentially a random walk. Thus to create one statistically independent chain of N monomers we need $\mathcal{O}(N^2)$ steps. This is the same as e.g. for reptation [21]. The actual CPU time depends on whether the potential is cut off at a finite distance or not, and whether efficient neighbour search [23] is used in the former case. All algorithms become less efficient for low temperatures and high monomer densities, but it seems that our algorithm is least affected. The pivot algorithm, which is the most efficient for high T and weak self-avoidance [22] ($\sim\mathcal{O}(N)$ steps), becomes essentially useless in this limit.

3. Applications

We applied this to two versions of a model where non-bonded monomers interact by Lennard-Jones potentials in 3D space,

$$u_{ij} = 4[r_{ij}^{-12} - r_{ij}^{-6}] \tag{12}$$

$$r_{ij} = |\boldsymbol{r}_{ij}| = |\boldsymbol{x}_i - \boldsymbol{x}_j|. \tag{13}$$

In the first version [2], the force between bonded monomers is harmonic,

$$v_i = \frac{\kappa}{2}\,(r_{i,i-1} - 1)^2 \tag{14}$$

$$\kappa = 400 \tag{15}$$

for $r_{i,i-1} > 0.5$. For $r_{i,i-1} < 0.5$, the potential is infinite. In the second version [7], it is simply provided by hard rods which keep a fixed distance $r_{i,i-1} = 1$. Notice that we did not truncate the potential at large distances, in contrast to previous studies [1, 7]. As far as we can see, the approximations involved in correcting for this truncation should be the only possible source for eventual disagreements with these works.

In the second version, a natural choice of $U_i^{(0)}$ is given by an isotropic delta potential† fixing $r_{i,i-1}$ at 1. Thus the vectors ξ were chosen randomly on the surface of a sphere centred at x_{i-1}. The corresponding partial partition functions are $Z_{n|i}^{(0)} = Z_n^{(0)} = (4\pi)^n$.

In the first version the choice is less obvious since it is not so easy to produce random points according to $\mathrm{e}^{-\beta v_i}$ with v_i given by (14). We thus took

$$U_i^{(0)} = v_i(r_{i,i-1}) + k_{\mathrm{B}}T\ln(r_{i,i-1}^2). \tag{16}$$

For this choice the radial distribution function is a Gaussian centred at $r = 1$, $\rho^{(0)}(r_{i,i-1}) \propto r_{i,i-1}^{-2}\,\mathrm{e}^{-\beta v_i}$, and $Z_n^{(0)} = [(2\pi)^3/(\kappa\beta)]^{n/2}$ for $\kappa\beta \gg 1$‡.

Figure 1 shows our data for the excess chemical potential $\beta\mu^{\mathrm{ex}}$ for the hard rod and the harmonic potential, respectively. The value $\beta = 1/1.2$ was chosen to compare our data to those of [7]. From figure 5 of that paper we see that $\beta\mu^{\mathrm{ex}} = -10.4 \pm 0.4$ for a hard-rod chain with $N = 30$ monomers. This is much smaller (in absolute value) than our estimate $\beta\mu^{\mathrm{ex}} = -15.35 \pm 0.05$. Part of this discrepancy can be explained by the fact that the Lennard-Jones potential was truncated in [7] at $r = r_{\mathrm{c}} \equiv 2.5$ and its contribution from $r > r_{\mathrm{c}}$ was estimated analytically. This was not done in our simulations, where all potentials were taken into account exactly. But by performing the same truncation as in [7] without correcting for it at all, we estimated that even this is much too small (≈9–10%)

† By this we mean of course not strictly a delta function but a potential which is singular enough to give a delta function for the equilibrium density, $\mathrm{e}^{-\beta U_i^{(0)}} = \delta(r_{i,i-1} - 1)$.

‡ If this condition is not fulfilled we get an additional term proportional to an error function due to the centring of $r_{i,i-1}$ around unity.

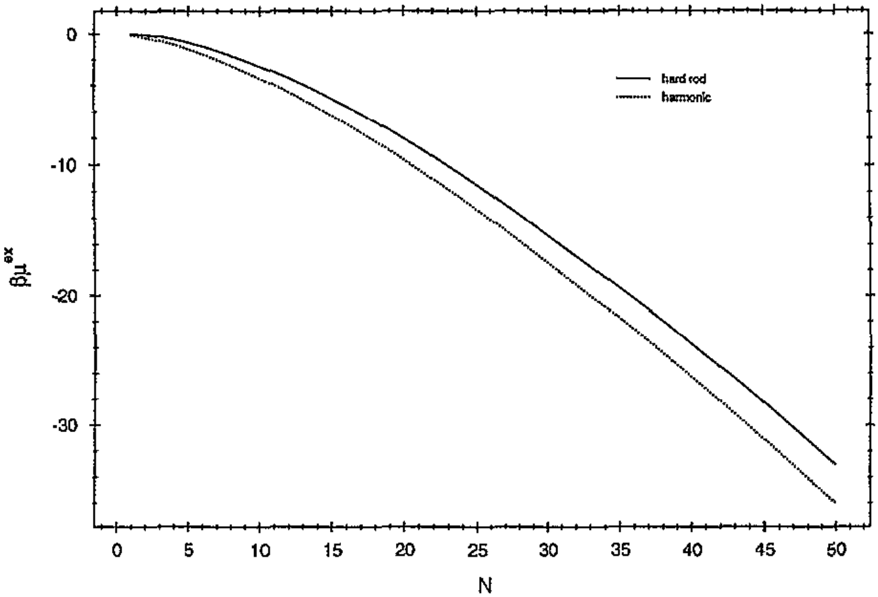

Figure 1. Results for the excess chemical potential at $T = 1/\beta = 1.2$. The upper curve is related to the hard-rod potential, while the lower one is the result for the harmonic potential of (15). Statistical errors are smaller than the thickness of the line.

to explain the discrepancy. The simulations of [1] have too large statistical errors for a similarly detailed comparison.

Averages of the squared end-to-end distance for longer chains are shown in figure 2. They clearly indicate that $\beta = 1/1.2$ is far below the Θ temperature in agreement with the fact that μ^{ex} is negative and decreasing with N. In figure 3 we show our data for $\beta\mu^{\mathrm{inc}} = \beta(\mu_N^{\mathrm{ex}} - \mu_{N-1}^{\mathrm{ex}})$, which represents the free energy needed to add one more monomer to the chain. At the Θ point we expect μ^{inc} to be independent of N. The chains are still too short to pin down β_Θ exactly, but the plots unambiguously show that β_Θ is much smaller then 0.36, the value given in [2]. Together with the data from figure 2 we would say that $0.2 \leqslant \beta_\Theta \leqslant 0.23$. Both the data of figure 2 and that of figure 3 were produced using the hard-rod potential for bonded monomers and the full LJ potential for non-bonded monomers, i.e. without performing any cut-off at large $r_{i,j}$. Each data set is based on at least 10^6 'tours'. We mean by 'tour' the set of all (correlated) chains produced by the same initial subroutine call from the main routine.

4. Generalizations and outlook

In the above applications, we did not truncate the potential at large r. Thus, inserting a new monomer takes a time $\mathcal{O}(N)$. For larger N this is no longer tolerable. If the potential is truncated in such a case, one should also use efficient data structures for searching for relevant neighbours [23]. With this we can achieve $\mathcal{O}(1)$ behaviour.

Our algorithm as described above can become inefficient for two main reasons, but in both cases this can be cured by minor modifications: first of all, if the temperature is very low, the Boltzmann factors $\mathrm{e}^{-\beta\Delta U_i}$ for single monomers can become very large. The efficiency of the method results from the fact that large Boltzmann factors for the entire chain are split into smaller factors for individual monomers. If the latter factors themselves

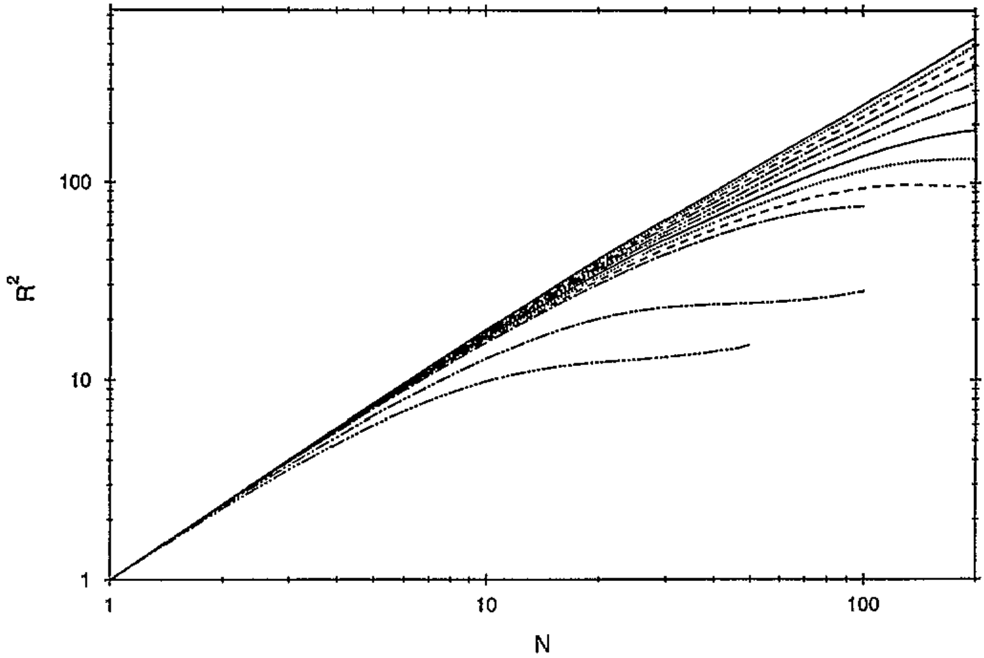

Figure 2. Simulation results for the average squared end-to-end distance for $\beta = 0.175, \ldots, 0.375$ with $\Delta\beta = 0.025$ and additionally for $\beta = 0.4, 0.6, 1/1.2$. The data clearly show that the chains are already collapsed at $\beta \geqslant 0.25$.

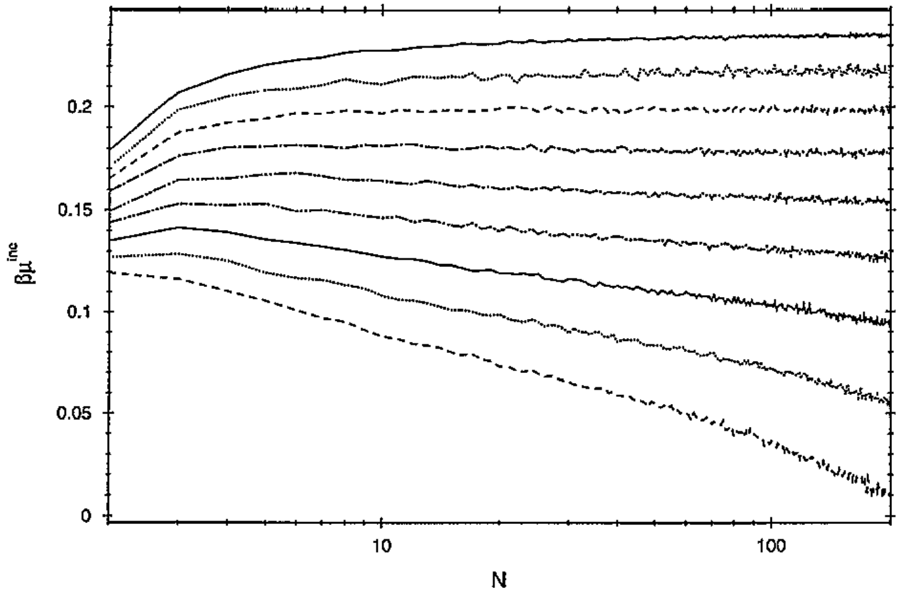

Figure 3. Simulation results for $\beta\mu^{\text{inc}}$ for $\beta = 0.175, \ldots, 0.375$ with $\Delta\beta = 0.025$. One can see that μ^{inc} is independent of N for $\beta \approx 0.2$ to 0.225. So we can expect the Θ point somewhere between these two values.

become large and are spread over a large interval, then most random positions will either be immediately discarded since they have very small w_i, or they will have very large w_i

and lead thus to very many replicas and correspondingly to huge fluctuations. A case where this made our method quite unsuccessful is the random heteropolymer model of [24].

The simplest way out of this dilemma consists in choosing $M \gg 1$. In this way we will have even more points which are essentially useless since they have very small w_i, but the regions in state space with high probability will be sampled more evenly, and this should be more important. Another popular remedy consists in making the bulk of the simulations at higher temperatures, and quenching down to the desired temperatures in regular time intervals [18].

The other case where the above algorithm gives poor results is provided by systems in which originally favoured configurations lead finally to dead ends, while originally unfavoured configurations become more important as the growth of the system continues. Two specific examples are dense polymer systems in 2D [6] and polymers with long-range repulsive forces [12, 13].

Assume we want to grow a long self-avoiding walk in a finite 2D region such that it fills a large fraction of the area. As we place the first monomers, all configurations are equally likely, but those which leave large closed voids are all dead ends since the walk later on cannot penetrate into the voids. Obviously it would be much better in this case to bias the walks against large voids from the very beginning.

The situation is similar for a polymer with repulsive Coulomb potentials. There, unless the force is very strong, the effect of the repulsion is not too strong for end monomers, and hence new monomers are added without a strong radial bias, but the stretching force on a monomer deep inside the chain is much bigger (since all forces essentially add up), and the configuration is much more stretched inside the chain. Thus, when an existing chain is to be elongated, most of the existing configurations have to be discarded, in order to be replaced by stretched configurations which at first (when they are assembled) are very improbable.

As we said, a way out of this problem is to use biased walks. This means that the number of replicas is not strictly proportional to w_i but is larger in those regions which we suspect to become more important later. Of course it means also that we have to replace (10) by a weighted sum. It is e.g. known that the Rosenbluth trick [25] leads to a bias towards more compact SAWs. We found indeed that our method with Rosenbluth weights instead of uniform weights was more efficient in giving SAWs which fill a square with periodic boundary conditions, but it leads to much larger statistical fluctuations at low density. We should point out that this possibility of biasing is *independent* of the choice of $U^{(0)}$, something which seems to have been missed in [16, 17, 18].

Finally, we tried our method also for non-polymeric systems. For instance, we simulated the 2D Ising model with spins numbered in the same way as they would be e.g. stored in a FORTRAN array. Though the method worked decently, it could not compete either with cluster flip methods (due to their much more efficient moves) or with conventional Monte Carlo schemes which can be made very efficient by vectorization and multispin coding. We should mention that the possible application to spin models, and to the Ising model in particular, was also pointed out in [26] in the context of the breadth first approach.

Acknowledgments

PG wants to thank A Vulpiani for hospitality at the University of Rome where part of this work was done. He also wants to thank D Frenkel for correspondence and S Carraciolo, E Marinari and G Parisi for very helpful discussions. We are indebted to H Meirovitch and I Chang for pointing out [16], [17] and [18]. Unfortunately, this happened only after we had used our method in several papers (and after the present paper was essentially finished).

We apologize to the above authors for not having cited them in previous papers. This work was supported by the DFG, Sonderforschungsbereich 237.

References

[1] Kumar S K, Szleifer I and Panagiotopoulos A Z 1991 *Phys. Rev. Lett.* **66** 2935

[2] Kumar S K 1992 *J. Chem. Phys.* **96** 1490

[3] Harris J and Rice S A 1988 *J. Chem. Phys.* **88** 1298

[4] de Pablo J J, Laso M and Suter U W 1992 *J. Chem. Phys.* **96** 6157

[5] Laso M, de Pablo J J and Suter U W 1992 *J. Chem. Phys.* **97** 2817

[6] Siepmann J I and Frenkel D 1991 *Mol. Phys.* **75** 59

[7] Frenkel D, Mooij G C A M and Smit B 1991 *J. Phys. C: Solid State Phys.* **3** 3053

[8] Sheng Y-J, Panagiotopoulos A Z, Kumar S K and Szleifer I 1994 *Macromolecules* **27** 400

[9] Smit B, Mooij G C A M and Frenkel D 1992 *Phys. Rev. Lett.* **68** 3657

[10] Kumar S K, Szleifer I and Panagiotopoulos A Z 1992 *Phys. Rev. Lett.* **68** 3658

[11] Grassberger P and Hegger R 1994 *J. Phys. A: Math. Gen.* **27** 4069, 7265; 1995 *J. Chem. Phys* at press; 1995 *Phys. Rev.* E at press

[12] Jönsson B, Peterson C and Söderberg B 1993 *Phys. Rev. Lett.* **71** 376; 1993 *Preprint* LU-TP 93-15

[13] Irbaeck A 1993 *Preprint* LU-TP 93-20

[14] Wall F T and Erpenbeck J J 1959 *J. Chem. Phys.* **30** 634, 637

[15] Grishman R 1973 *J. Chem. Phys.* **58** 220

[16] Garel T and Orland H 1990 *J. Phys. A: Math. Gen.* **23** L621

[17] Higgs P G and Orland H 1991 *J. Chem. Phys.* **95** 4506

[18] Velikson B, Garel T, Niel J C and Orland H, 1992 *J. Comput. Chem.* **13** 1216

[19] Tarjan R 1972 *SIAM J. Comput.* **1** 146

[20] Grassberger P 1993 *J. Phys. A: Math. Gen.* **26** 1023

[21] Kremer K and Binder K 1988 *Comput. Phys. Rep.* **7** 259

[22] Madras N and Sokal A D 1988 *J. Stat. Phys.* **50** 109

[23] Schreiber T 1995 *Int. J. Bif. Chaos* at press

[24] Iori G, Marinari E and Parisi G 1992 *J. Phys. A: Math. Gen.* **24** 5349; 1993 *Preprint*

[25] Rosenbluth M N and Rosenbluth A W 1955 *J. Chem. Phys.* **23** 356

[26] Niel J C and Orland H 1990 *Preprint* SPhT/90-137