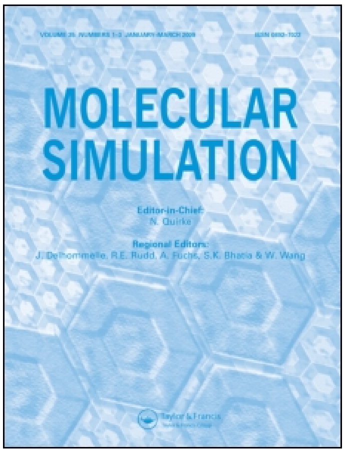

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Dynamic Monte Carlo Simulation of Polymers: Cooperative Move Algorithm

Yuan Qin; Hong-Lai Liu; Ying Hu

Online publication date: 13 May 2010

To cite this Article Qin, Yuan , Liu, Hong-Lai and Hu, Ying(2003) 'Dynamic Monte Carlo Simulation of Polymers: Cooperative Move Algorithm', Molecular Simulation, 29: 10, 649 — 654

To link to this Article: DOI: 10.1080/0892702031000103185

URL: http://dx.doi.org/10.1080/0892702031000103185

Molecular Simulation, Vol. 29 (10–11), October–November 2003, pp. 649–654

Taylor & Francis
Taylor & Francis Group

Dynamic Monte Carlo Simulation of Polymers: Cooperative Move Algorithm

YUAN QIN, HONG-LAI LIU* and YING HU

Department of Chemistry, East China University of Science and Technology, Shanghai, 200237, People's Republic of China

(Received September 2002; In final form December 2002)

If the elementary move is carefully chosen, the Monte Carlo process can be used to simulate the dynamics of the polymers in a coarse-grained sense. It is generally believed that only local move algorithms, for example, generalized Verdier–Stockmayer algorithm and bond fluctuation method, are suitable for dynamic Monte Carlo simulation of polymer. In this work, we use a cooperative move algorithm, in which several connected segments of the polymer can move collectively, to simulate the self-avoiding walk (SAW) and random walk (RW) chains on the 2-dimensional triangle lattice. We find this cooperative move algorithm can reproduce the result of Rouse theory well, so it can be used as a dynamic Monte Carlo simulation algorithm. The cooperative move algorithm is more realistic than the conventional local move ones in that it can mimic the tensile forces in the polymer chains.

Keywords: Monte Carlo simulation; Polymer; Dynamics

The dynamic Monte Carlo simulation plays an important role in the study of polymer dynamics [1]. The key point of dynamic Monte Carlo simulation is to choose the appropriate algorithm that can correctly depict the dynamics of the real system. As early as 1962, Verdier and Stockmayer [2] began to use Monte Carlo simulation to study the dynamics of polymers, they used two one-segment move: kink jump and end rotation. Hilhorst and Deutch [3] pointed out that the kinetic scheme chosen by Verdier and Stockmayer led to severe constrains on the chain motion and resulted in artificial slowing down of the chain relaxation. To modified Verdier and Stockmayer's algorithm, they proposed a third kind of move: crankshaft rotation. Now it is generally accepted that the combination of three local motions: kink jump, end rotation and crankshaft rotation, can mimic the real dynamics of polymer [1]. In 1988, Carmesin and Kremer [4] proposed a bond fluctuation method. Compared with the generalized Verdier–Stockmayer algorithm, the bond fluctuation method introduced the characteristics of free space into the lattice model and hence the chains can move more freely. However, the local move nature remains unchanged. Now most dynamic Monte Carlo simulations of polymer are based on the bond fluctuation method.

Different with the one or two segments local move algorithm, cooperative move algorithms stress collectively motion of the segments. In 1987, Pacula [7] has proposed a collective move algorithm to simulate the polymer melt, which consists of collective replacement of segments within a closed loop of motion. The significant feature of Pacula algorithm is that it can successfully move the chains on a lattice with all sites occupied. However, this algorithm stresses too much on the intermolecular collectively motions and results in wrong dynamics of polymer melts. For a long time, cooperative move algorithms are not used as dynamic Monte Carlo simulation algorithm, although they are used as an effective algorithm to study static properties. Recently, three groups of authors independently proposed intramolecular cooperative move algorithm to study the dynamics of polymer: Azuma and Takayama [11] proposed a new bond fluctuation method which incorporates non-local move to simulate the electrophoresis of polyelectrolyte. To increase the time efficiency of the simulation of polymer melt, Haire *et al.* [14] adopt a cooperative move algorithm. By correcting the error in literature [5], we [12,13] also get a cooperative move algorithm.

*Corresponding author.

ISSN 0892-7022 print/ISSN 1029-0435 online © 2003 Taylor & Francis Ltd
DOI: 10.1080/0892702031000103185

The cooperative move algorithms proposed by the three groups consist of collective move of several connected segments, this kind of motion can be interpreted as the effect of tensile force in the chain.

It is generally believed that only local move algorithm can be used to study the dynamics of polymer [1]. In this paper, we want to emphasize the utility of the cooperative move algorithm as a scheme to simulate the dynamics of polymer. The paper is organized as follows. First, we give a description of the algorithm. Then we use this algorithm to simulate the self-avoiding walk (SAW) and random walk (RW) chains on 2-dimension triangle lattice and the results are compared with Rouse theory.

THE ALGORITHM

We get the algorithm by correcting the error of Lu and Yang's [5] algorithm. In Lu and Yang's work, 2-dimensional square lattice was used and the bond length can be 1 or $\sqrt{2}$, thus the coordinate number of the lattice is 8. This is the simplest case of the bond fluctuation method [4]. During the simulation, a segment and a cavity among the 8 neighbors of the segment are selected randomly (for the dense solution the cavity is selected prior to the segment [5]), then the segment is tried to move to the cavity. If the selected segment is not the end, there are three possible cases: 1. both the two bonds connecting the segment break or bond crossing occurs; 2. both the two bonds remain connected; 3. one bond breaks and the other remains connected. For case 1, the move fails and the old configuration is re-accounted. For case 2, one segment move is performed. For case 3, Lu and Yang [5] suggest the so-called middle slithering move: the broken portion of the chain perform slithering motion towards the selected segment thus the broken bond is re-connected, as shown in Fig. 1(a). If the selected segment is the end of chain, the situation is similar. Obviously, the middle slithering motion often makes a large portion of the chain move simultaneously. This is unphysical, so it seems inappropriate to use this kind of move to perform dynamic Monte Carlo simulation. More severely, Feng *et al.* [6] pointed out that this kind of move violate the detail balance condition, this can be clearly seen from Fig. 1(a): the right configuration can be changed into the left one by one move, but the reverse change is impossible. The actual calculation reveals that the static properties by this algorithm are greatly different from the correct value [6].

We notice that there are some special segments in the bond fluctuation model, such as the s ones shown in Fig. 1, which can be called slack segments. The slack segments consist of those segments whose nearest neighbor segments satisfy the bonding condition. For convenience, the end segments are also regarded as slack segments. It is obvious that after performing the middle slithering move, the selected segment becomes a slack segment. We find that when performing the middle slithering move, if the slithering move stop at the nearest slack segment, as shown in Fig. 1(b), the detail balance condition can be satisfied. As there are many slack segments, the number of the segments performing slithering move decrease greatly. Physical meaning of this move mode can be understood as follows: there is tensile force in the chain, when one segment is moved, under the tensile force the nearby segments will move collectively, but the tensile force can only transmit through the "tight" portion of the chain and the transmission stop at the slack segment, so the possibility for a large portion of the chain to move collectively is low.

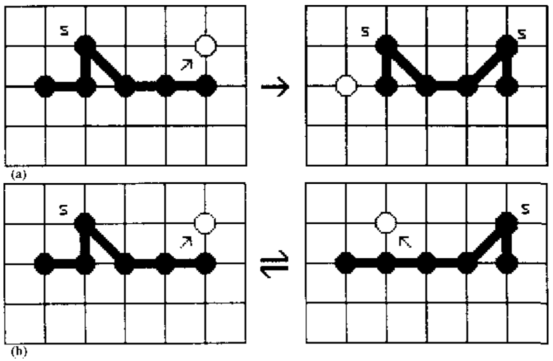

FIGURE 1 (a) Middle slithering move proposed by Lu *et al.* [15], which is wrong in that detail balance condition is violated. (b) The collective move of this work, which satisfies detail balance condition.

The choosing of the lattices type needs further discussion. The above presentation is based on the simple bond fluctuation method, if bond fluctuation is not allowed, it needs two segments to form "slack segments" in square lattice. And triangle lattice is a more natural choice in that only one segment is required to form slack segment. Figure 2 shows the move mode on the triangle lattice.

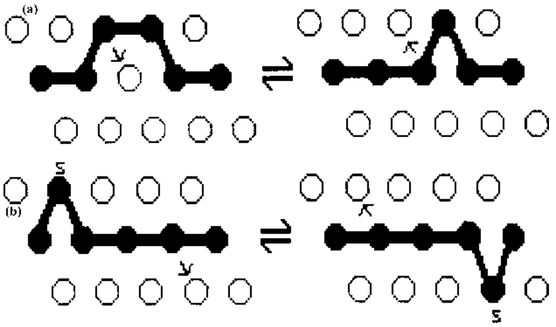

FIGURE 2 Move mode of the algorithm on triangle lattice: (a) the single segment move; (b) the cooperative move.

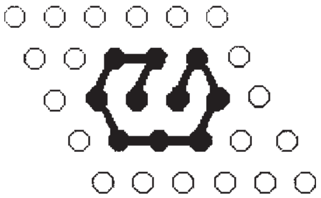

FIGURE 3 A configuration which cannot be disturbed by one segment move but can be disturbed by cooperative move.

Compared with free space models, the lattice models have the advantage of high speed of computation, but also bring the cumbersomeness of the polymer motion. For example, the generalized Verdier–Stockmayer algorithm has severe ergodic problem in 2-dimensional square lattice. It should be pointed out that there are two reasons for this disadvantage: 1. the lattice structures restrict the polymer's motion greatly; 2. the often-used local move algorithm itself is problematic. Carmesin and Kremer's bond fluctuation method combine the free space characteristics with the lattice model and make the polymer move more easily, so this method overcomes the first difficulty, but the fluctuating bond length brings new complexity. It is interesting that the present cooperative motion algorithm is ergodic even the bond fluctuation method isn't employed. Figure 3 shows a polymer configuration on triangle lattice, which cannot be disturbed by one segment move but can be disturbed by cooperative move.

It should be noted that cooperative move is not a new idea and there are several similar algorithms in literatures. In 1987, Pacula [7] has proposed a collective move algorithm which can successfully disturb the polymer melt with unity concentration. This algorithm included two kinds of move: the intramolecular and intermolecular cooperative moves. Pacula's algorithm stress too much on the intermolecular collective move and it cannot reproduce the dynamic of the polymer melt correctly. The failure of Pacula's algorithm in simulating the dynamics of polymer melt, to some extent, has discouraged the application of the cooperative move algorithm in polymer dynamic Monte Carlo simulation. In the recent work of Haire *et al.* [14], only the intramolecular cooperative move is used to simulate dynamics of the polymer melt and the correct results are obtained. The general reptation algorithm [8–10], which consists of slithering move of portion of the chain between two randomly chosen slack segments, can be regarded as intramolecular cooperative moves. However, this kind of move often involves large portion of segments, this is unphysical, hence it is only used as an efficient algorithm to simulate the static properties and not used to study the dynamics of polymer. Obviously, the present algorithm is a special case of the general reptation algorithm: only the segments between two neighbor slack segments can perform the slithering move, this reduces the number of the segments per move greatly and is more realistic. Recently, to simulate the gel electrophoresis of the polyelectrolyte, Azuma *et al.* [11] has proposed a cooperative algorithm which can overcome the difficulties that the polymers are trapped by the obstacles in relatively large electrical fields. Azuma *et al.*'s algorithm is based on the bond fluctuation method and the procedure to perform the cooperative move is complicated. Relatively, the present algorithm is more straightforward. Azuma *et al.*'s work is illuminable in that they demonstrated the significance of the cooperative move in the case where the tensile force in the chain play an important role. In this paper, the term "slack segment" and the physical interpretation of the algorithm is adopted from Azuma's work.

SIMULATION RESULTS

We simulate the RW chain and the SAW chain on the triangle lattice using the present algorithm. The simulation is performed on a rhombus lattice and the periodical boundary conditions are used. In the present paper, every segment move is used as the time unit and the results are the average of 2000 independent samples.

Figure 4 shows the number of segments involved in one attempted move for the 32 segments chain. It is shown that 41.7% move is failed, 12.5% involved only one segment, 16.2% involved two segments and those involved more than 10 segments is only 0.5%. Figure 5 shows the square of the end-to-end distance R_n^2. It is shown that the following scaling laws are satisfied:

$$R_n^2 = (N-1)^n \qquad (1)$$

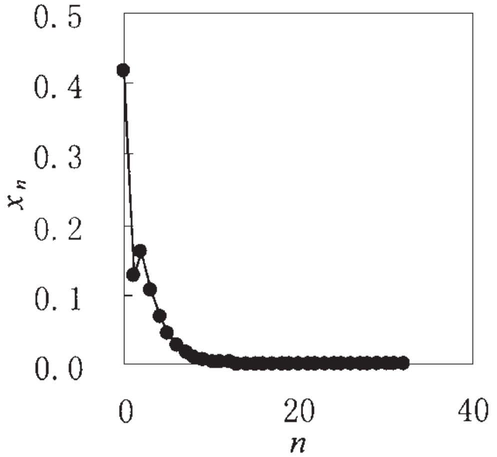

FIGURE 4 The average number of the segments involved in one attempted move.

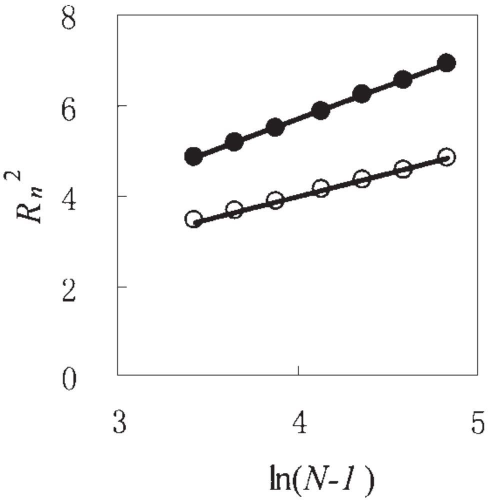

FIGURE 5 The square of the end to end distance ○: RW chains ●: SAW chains.

For RW chain, n is 1.0 and for SAW chain, n is 1.5, both are in accordance with the theory.

For the dynamic properties we first check three correlation functions:

$$g_1(t) = \langle [R_{1/2}(t) - R_{1/2}(0)]^2 \rangle \quad (2)$$

$$g_2(t) = \langle [(R_{1/2}(t) - R_{\text{mc}}(t)) - (R_{1/2}(0) - R_{\text{mc}}(0))]^2 \rangle \quad (3)$$

$$g_3(t) = \langle [R_{\text{mc}}(t) - R_{mc}(0)]^2 \rangle \quad (4)$$

where $R_{1/2}$ is the coordinate of the central segment, R_{mc} is the coordinate of the mass center. Figure 6 shows the changes of g_i with time for SAW chain. In short time scale the scaling exponent of g_1 is 0.6, in long time scale the scaling exponent of g_1 is 1.0. In all time scale the scaling exponent of g_3 is 1.0. In short time g_2 coincides with g_1 and in the long time g_2 approaches a limiting value. For RW chain, the situation is similar, the difference is that the scaling exponent of g_1 and g_2 is 0.5 instead of 0.6. These results accord with the theoretical prediction.

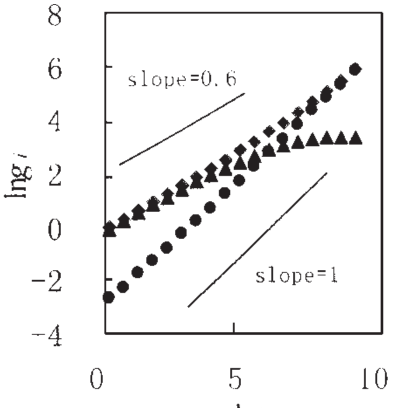

FIGURE 6 For SAW chain ($N = 64$), g_1, g_2 and g_3 vs t. ◆:g_1, ▲: g_2, ●: g_3.

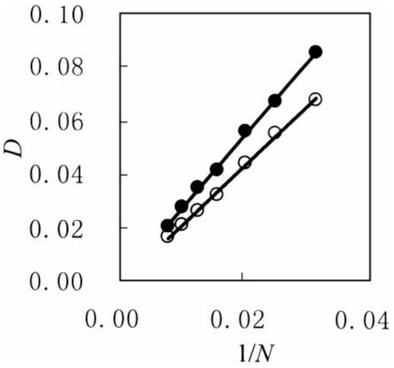

FIGURE 7 Self-diffusion coefficients: ○: RW chains; ●: SAW chains.

Figure 7 shows the diffusion coefficient obtained by the present algorithm and the conventional single segment move algorithm. The diffusion coefficient is defined by

$$D = \lim_{t \to \infty} \frac{g_3(t)}{4t} \quad (5)$$

Figure 7 shows that for both RW and SAW chains $D \sim N^{-1}$.

The more strict check is the Rouse normal modes, which are defined by:

$$X_p(t) = \frac{1}{N} \sum_{n=1}^{N} R_n(t) \cos \frac{(n - 0.5)p\pi}{N} \quad (6)$$
$$p = 1, 2 \cdots N - 1$$

By Rouse theory, the time correlation function of the normal modes decrease exponentially:

$$\frac{\langle X_p(t) X_p(0) \rangle}{\langle X_p(0) X_p(0) \rangle} = \exp\left(\frac{-t}{\tau_p} \right) \quad (7)$$

τ_{p} is the relaxation time [15]:

$$\tau_p \sim \sin^{-2}(p\pi/2N) \quad (8)$$

As the Rouse theory don't consider the excluded volume effect, we only analyze the RW chain results. By fitting the simulated correlation function of the normal modes we get the relaxation time τ_p. Figure 8 shows the τ_ps for chains of 32, 64 and 128 segments satisfy formula (8) well. At last, we examine the correlation function of the end-to-end distance $P(t) = R_N - R_1$:

$$\phi(t) = \frac{\langle P(t) \cdot P(0) \rangle}{\langle P(0)^2 \rangle} = \sum_{p=1,3,\ldots} \frac{8}{p^2 \pi^2} \exp\left(\frac{-tp^2}{\tau_1} \right) \quad (9)$$

If time is rescaled using the Rouse relaxation time τ_1, the simulated correlation function of the end-to-end distance of different length polymer satisfy Eq. (9) as shown in Fig. 9.

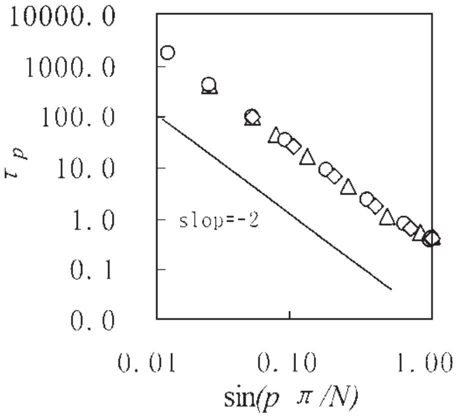

FIGURE 8 τ_p vs. $\sin(p\pi/2N)$ ◇: $N = 32$; △: $N = 64$; ○: $N = 128$.

CONCLUSION AND DISCUSSION

It is generally accepted that only those algorithms, which generate Rouse dynamics in the free draining limit, can be used as the polymer dynamic Monte Carlo algorithm. The dynamic behavior of the present algorithm is in accordance with Rouse theory, so this algorithm can be used to study polymer dynamics.

Dynamic Monte Carlo simulation has some inherent difficulties. An important one, we believe, is that by using the sequential Monte Carlo simulation process to mimic the real parallel process, the cooperative move of the system should be appropriately considered. However, for a long time the cooperative move algorithms have not been sufficiently used to simulate the polymer dynamics. Recently some researchers begin to stress this problem, Haire *et al.*'s [14] and our work reveal that the cooperative algorithm can also reproduce the polymer dynamics correctly. Azuma *et al.*'s work [11] reveal that for the situation where the tensile force plays an important role, the cooperative algorithm must be used. We believe that cooperative algorithm will play a more important role in the study of polymer dynamics.

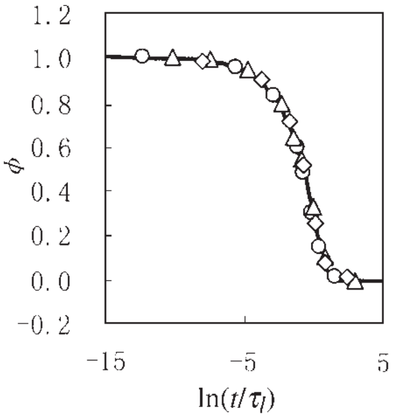

FIGURE 9 End-to-end vector correlation function ◇: $N = 32$; △: $N = 64$; ○: $N = 128$. Solid line: calculation by Eq. (9).

An important difference between molecular dynamics simulation and the dynamic Monte Carlo simulation is that in molecular dynamic simulation the molecules move mechanically guided by the dynamic equations, but in dynamic Monte Carlo simulation, the move mode of the molecule contain the researchers' understanding of the dynamics. It is interesting to reflect the physical picture behind the different dynamic Monte Carlo algorithms. For the conventional single segment move algorithm, the motion of polymer seems to be the hops of the individual segments, but for the present cooperative algorithm, the motion of the polymer is more like the wriggle of worms.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Project No.20025618, 20236010) and the Education Committee of Shanghai City of China.

References

[1] Binder, K. and Paul, W. (1997) "Monte Carlo simulation of polymer dynamics: recent advances", *J. Polym. Sci. Polym. Phys. Ed.* **35**, 1.

[2] Verdier, P.H. and Stockmayer, W.H. (1962) "Monte Carlo calculations on the dynamics of polymers in dilute solution", *J. Chem. Phys.* **36**, 227.

[3] Hilhorst, H.J. and Deutch, J.M. (1975) "Analysis of Monte Carlo results on the kinetics of lattice polymer chains with excluded volume", *J. Chem. Phys.* **63**, 5153.

[4] Carmesin, I. and Kremer, K. (1988) "The bond fluctuation method: a new effective algorithm for the dynamics of polymers in all spatial dimensions", *Macromolecules* **21**, 2819.

[5] Lu, J.M. and Yang, Y.L. (1991) "Dynamic Monte Carlo simulation of high concentration multichainsystem: bond fluctuation model and cavity diffusion algorithm", *Sci. China, A* **11**, 1226, (in Chinese).

[6] Feng, J., Cai, J., Liu, H.L. and Hu, Y. (2000) "Monte Carlo simulation for microphase separation of diblock copolymers", *J. East China Univ. Sci. Tech.* **26**, 421, (in Chinese).

[7] Pakula, T. (1987) "Co-operative relaxations in condensed macromolecular system I. A model for computer simulation", *Macromolecules* **20**, 679.

[8] Murat, M. and Witten, T.A. (1990) "Relaxation in bead-jump polymer simulations", *Macromolecules* **23**, 520.

[9] Reiter, J. (1990) "Monte Carlo simulations of linear and cyclic chains on cubic and quadratic lattices", *Macromolecules* **23**, 3811.

[10] Nelson, P.H., Hatton, T.A. and Rutledge, G.C. (1997) "General reptation and scaling of 2*d* athermal polymers on close-packed lattices", *J. Chem. Phys.* **107**, 1269.

[11] Azuma, R. and Takayama, H. (1999) "Bond fluctuation method for a polymer undergoing gel electrophoresis", *Phys. Rev. E* **59**, 650.

[12] Qin, Y., Liu, H.L. and Hu, Y. (2002) "Cooperative motion algorithm for dynamic Monte Carlo simulation of polymer", *Chem. J. Chinese Univ.* **23**, 1447, (in Chinese).

[13] Qin, Y., Liu, H.L. and Hu, Y. (2001) "Dynamic Monte Carlo simulation of polymer: cooperative motion algorithm in 2-dimensional triangle lattice", *J. Functional Polym.* **14**, 417, (in Chinese).

[14] Haire, K.R., Carver, T.J. and Windle, A.H. (2001) "A Monte Carlo lattice model for chain diffusion in dense polymer systems and its interlocking with molecular dynamics simulation", *Comput. Theor. Polym. Sci.* **11**, 17.

[15] Verdier, P.H. (1966) "Monte Carlo studies of lattice model polymer chains. I. Correlation functions in the statistical-bead model", *J. Chem. Phys.* **45**, 2118.