ELSEVIER

Parallel Computing 27 (2001) 613–641

PARALLEL
COMPUTING

www.elsevier.com/locate/parco

Cellular automata for polymer simulation with application to polymer melts and polymer collapse including implications for protein folding

B. Ostrovsky [a], G. Crooks [b], M.A. Smith [c], Y. Bar-Yam [d,*]

[a] Sun Microsystems Computer Company, 5 Omni Way, Chelmsford, MA 01824, USA
[b] Department of Chemistry, University of California, Berkeley, CA 94720, USA
[c] Department of Molecular and Cellular Biology, Harvard University, 16 Divinity Ave., Cambridge, MA 02138, USA
[d] New England Complex Systems Institute, 24 Mt. Auburn St., Cambridge, MA 02138, USA

Received 31 March 1999; received in revised form 20 December 1999

Abstract

Cellular automata can be designed that allow the simulation of a large variety of polymer problems including isolated polymers in dilute solution, polymers in high density melts and polymers embedded in media. The two-space algorithm is a particularly efficient algorithm for polymer simulation that is easy to implement and generalize on both conventional serial hardward and Cellular Automaton (CA) Machines. We describe the implementation of this algorithm and two applications: two dimensions (2-D) melts and polymer collapse. Simulations of high density melts in 2-D show that contrary to expectations polymers do not segregate at high density, there is significant interpenetration as there is in 3-D. Polymer collapse is studied in the regime far from equilibrium. Collapse is found to be dominated by migration of the chain ends. The kinetic process of collapse can systematically and reproducibly restrict the possible conformations that are explored during protein folding. This suggests that the kinetics of collapse may help lead to the desired folded conformation of proteins. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Cellular automata; Polymer collapse; Polymer melts; Protein folding; Domain decomposition

[*] Corresponding author.
E-mail address: yaneer@necsi.org (Y. Bar-Yam).

0167-8191/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0167-8191(00)00081-8

1. Introduction to polymer simulation

Polymers are molecules formed out of long chains of atoms that are generally recognizable as a sequence of units called monomers. Biological polymers include proteins, DNA, and polysaccharides. Artificial polymers include polystyrene and polyethylene. Polymers are often found dissolved in liquids. The dynamic properties of polymeric systems are of intense academic and industrial interest [6,7]. The key problem is understanding how their behavior arises from their structure. The high number of conformations of long polymers make conventional computer simulations prohibitive.

The idea of a Cellular Automaton (CA) is to think about simulating the space rather than the objects that are in it [1,9,19,20]. Standard CA which update a single cell on the basis of the value of cells in its neighborhood are not well suited to the description of systems with constraints or conservation laws. For example, if we want to conserve the number of ON sites (e.g. particles) we must establish a rule where turning OFF one site is tied to turning ON another site, corresponding to movement of the particle. Polymeric systems are such systems because they require conserving the number of particles (monomers), as well as the bonds between monomers along the polymer chain which are generally assumed not to break during the course of a simulation. One way to impose particle conservation is to use a lattice gas [8]. We will use a different modification of CA called Margolus dynamics which can be more easily generalized to other conservation laws [19]. The idea is to partition the space into plaquettes. The CA rule describes the update of a plaquette, rather than a single cell. Then, a conservation law that holds in the plaquette (e.g. the number of ON sites) also holds globally. For example, the plaquettes might be 2×2 regions of cells. After each update of the space, the partition of the space into plaquettes is shifted. This restores the cellular periodicity of the space. In a sense, Margolus dynamics is a cellular version of domain decomposition strategies for parallel simulation. Indeed, the CA based simulation strategies we describe in this article can be generalized for the simulation of arbitrary polymer models as domain decomposition strategies [13].

In this article we describe dynamical models in the class of Cellular Automata that represent the highly successful “abstract” models for long (high molecular weight) polymers. The dynamical models are simple to implement, fast in execution on conventional computers, easy to parallelize, and readily adapted to introduce various features of particular problems in the study of polymers. We use these algorithms in two applications: the properties of high density polymer melts in two dimensions (2-D), and the collapse of a polymer from an expanded to a globular form. A third application – the diffusion of polymers through a field of obstacles, is described in other articles.

We will be primarily concerned with the scaling behavior of the properties of long polymers as a function of polymer length N [6,7]. This behavior should not be sensitive to the chemical composition and therefore the abstract model of polymers will be a useful model of the behavior of real polymers that are long enough so that the effect of local structure is unimportant. The scaling theory of polymer structure and dynamics is one of the great successes of simple concepts in understanding complex systems. In brief, according to this theory, any characteristic size of the polymer should scale as $R \sim N^{\nu}$, where $\nu = 3/(d+2)$ is the Flory exponent, and $d$is the dimension of space. The diffusion constant D and internal relaxation time τ are related by $\tau \sim D/R^2$ and the relaxation time scales as $\tau \sim N^z$, where $z = 2\nu + 1$ when hydrodynamics is not included (Rouse relaxation) and $z = 3\nu$ when hydrodynamics is included (Zimm relaxation).

2. Monte Carlo polymer simulation

Conventional simulations of polymer systems are of two types: molecular dynamics, and Monte Carlo [1,2,6,7]. Molecular dynamics simulations are suggestive of 'realistic' Newtonian dynamics of polymers and are implemented by moving all atoms with small steps according to forces calculated from modeled interatomic forces. Monte Carlo represents the dynamics of an ensemble of polymers by steps which take into account thermodynamic transition probabilities. Both techniques give the same results for structure, conformational change and diffusion. All atoms can be moved in parallel (at the same time) in molecular dynamics, which therefore appears to be ideally suited for parallel processing computers. However, with a processor attached to each atom, calculation of the forces requires a large number of communications between processors. Connections between processors are typically the limiting feature of parallel computers. Domain decomposition methods can be used for parallel simulations of both Monte Carlo and molecular dynamics. In this article we focus on Monte Carlo simulations using particularly efficient CA based algorithms.

Monte Carlo dynamics correspond to diffusive dynamics that do not conserve momentum. More formally, the dynamics allows local steps in the space of polymer conformations. Even if these local steps do not follow the same distribution as the real dynamics, the long range motion is consistent with the real dynamics when diffusive motion dominates the system behavior. Such a model is reasonable in the dissipative fluid environment of polymers when we are interested in describing large changes in polymer conformation, or diffusion of the polymer as a whole. It also provides correct results for the ensemble of polymer conformations and thus for average structural properties such as the spatial extent of the polymer. However, it should be noted that the diffusive dynamics corresponds to Rouse relaxation since it does not include correlations in motion of polymer segments arising from the motion of the fluid. The motion of the fluid which couples the motion of different parts of the polymer should be treated using hydrodynamics. These correlations are included approximately in Zimm relaxation but simulation schemes for this type of dynamics have not been developed, and we will not address this question here.

Monte Carlo can be more efficient than molecular dynamics because it, in effect, allows larger time steps in simulations. However, large movements of monomers cause difficulties for parallelization due to the problem of accounting for the joint effects of simultaneous displacements. In solving this problem we will motivate the introduction of our CA models.

In a Monte Carlo simulation of abstract polymer structure and dynamics [1,2], a long chain of monomers is represented by the coordinates of each monomer. There are many different methods for describing the polymer. However, quite generally, a simulation step consists of selecting a monomer, monomer *i*, from the polymer chain and performing a move subject to the constraints: (1) The move does not "break" the polymer connectivity – monomer *i* does not dissociate itself from its nearest neighbors along the chain; and (2) The move does not violate excluded volume – monomer *i* does not overlap the volume of any other monomer *j*.

These two constraints, connectivity and excluded volume, are sufficient to guarantee that the structural properties of a long polymer will be found. In order to study the dynamics of a polymer we must also guarantee that the steps taken are local steps in the space of polymer conformations. This is generally satisfied when monomer steps themselves are local. However, we must also be sure that the polymer cannot pass through itself. For the models we will use it is easy to verify that this cannot happen.

To motivate our dynamic models, we consider the problem of performing a polymer simulation using a parallel machine. In naive parallel processing, a set of processors would be assigned one-to-one to perform the movement of a set of the monomers. Each processor does not know the outcome of the movement of the other monomers, it can only know their position before the current step. With the two constraints (1) and (2) it would be impossible to perform parallel processing in this way since moving different monomers at the same time is likely to lead to dissociation or overlap. Dissociation only restricts the parallel motion of nearest neighbors. However, the excluded volume constraint restricts the parallel motion of *any* two monomers, presenting a fundamental difficulty for parallel processing. A general way to overcome this difficulty is by recognizing that polymer interactions are *local* in *space*. The polymer can coil so as to bring any two monomers into contact, yet, at any particular time, the only possible interactions are between monomers which are nearby in space.

3. Cellular automata for polymer dynamics [1,13,15,17,18]

A CA Margolus dynamics for the simulation of polymers is illustrated in Figs. 1 and 2. Each cell can have two values (ON and OFF). ON cells represent monomers, and a polymer is described by a set of monomers which touch either at corners or on edges of the cells. In a chain polymer (Fig. 1) each monomer has two such neighbors except for the ends which have only one. The general strategy for constructing such a CA begins with the recognition that the update rule only depends on the local spatial conformation. Bonds between monomers should be specified solely by the relative position of the monomers. Thus, we think about a bonded neighbor as a monomer that is closer than a certain distance. Any other (non-bonded) monomer must be farther away. By imposing this as a constraint, we are imposing an excluded volume

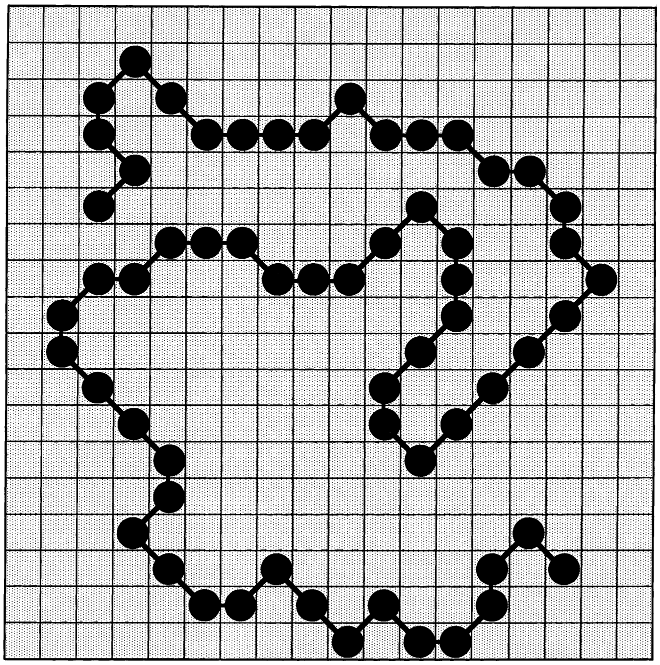

Fig. 1. Illustration of a cellular polymer model. ON cells represent monomers. In this model monomers are considered attached if they are touching by either faces or corners.

between non-bonded neighbors that is larger than a single cell. We call the space around a monomer in which its bonded neighbors are located the bonding neighborhood. It can be shown that in any such model, in 2-D or 3-D, the polymer chain cannot pass through itself.

In Fig. 1 monomers which touch either at corners or on edges of the cells are bonded. The bonding neighborhood is a 3×3 region around each monomer. For a chain polymer, each monomer has two such neighbors except for the ends which have only one.

In updating the polymer, we could use the usual Monte Carlo approach. We would pick one monomer, pick a compass direction (North East West and South (NEWS)) and move the monomer in the selected direction if the constraints of connectivity and excluded volume allow. The connectivity constraint prevents a monomer move from breaking a bond. The excluded volume constraint prevents a non-bonded monomer from entering the bonding neighborhood. Both constraints may be imposed by the condition that the move does not change the monomers in the bonding neighborhood.

For the CA, rather than picking one monomer to move, we select a sub-lattice of cells separated by 3 cells in each direction as shown in Fig. 2. If a monomer is located in one of these cells, we chose a compass direction at random to move it, and move it if permitted by the constraints of connectivity and excluded volume. In the

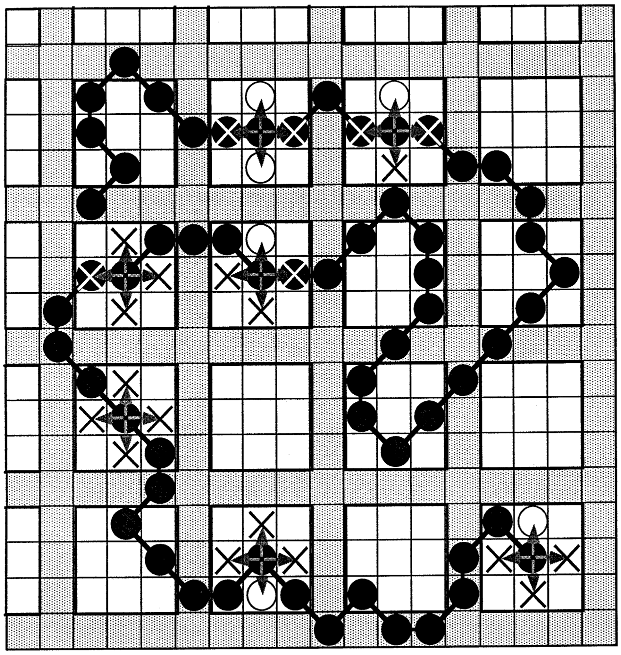

Fig. 2. Illustration of the implemenation of Margolus Dynamics on the Cellular Automaton of Fig. 1. Moves can only originate in the squares in the middle of 3 × 3 neighborhoods separated by buffers. Moves are shown by the arrows. Moves which are not allowed are marked by an "X" in the target square.

figure, steps that are not allowed are indicated by Xs. A movement of a monomer corresponds to an update of the 3 × 3 plaquette around a selected site; the information for the update is contained in a larger 5 × 5 region. All 3 × 3 plaquettes can be updated simultaneously. Finally, a new sub-lattice is picked at random for the next update.

In the simulation of long polymers, it is important to have a model where the behavior of a long chain is realized for moderate numbers of monomers. As for real polymers, the long-chain behavior is reached when the details of the local properties become unimportant. Thus we choose the local dynamics to minimize the influence of local constraints on the dynamics. There are two characteristic types of local dynamical behavior of a polymer – motion perpendicular to the polymer contour, and motion along the polymer contour (which involves local length changes). An effective approach to minimize the influence of local structure is to allow local changes in polymer length. This is not possible in the CA rule just described.

To solve this problem we want to allow the monomers that are bonded to each other to separate by one lattice space. This implies that we should increase the size of the bonding neighborhood to be a 5 × 5 region in 2-D, or a 5 × 5 × 5 region in 3-D. This choice of bonding neighborhood is convenient, but others could be specified as well. As before, we do not allow monomers to violate excluded volume by entering a bonding neighborhood, and we do not allow monomers to break a bond by leaving. A monomer move is accepted if monomers are not removed from nor added to the bonding neighborhood by the move. The larger bonding neighborhood allows more flexibility to the motion because adjacent monomers can move towards and away from each other, enabling local contraction and expansion of the polymer. We call this algorithm the one-space algorithm in order to contrast it with the two-space algorithm discussed next.

4. Two-space algorithm [1,13,15,17,18]

The problem of polymer flexibility also has a second solution – the two-space algorithm – that has some additional advantages. The simplest way to describe the two-space algorithm in 2-D is to consider a polymer on two parallel planes (Fig. 3). The monomers alternate between the planes so that odd-numbered monomers are on one plane and even-numbered monomers are on the other. The neighbors of every monomer reside in the opposite space. The bonding neighborhood is a 3 × 3 region of cells in the opposite space. This is the region of cells in which only its neighbors reside. To construct a polymer we place successive monomers so that each monomer has its nearest neighbors along the contour in its bonding neighborhood. The dynamics is defined, as before, by requiring that a monomer move be allowed only if its movement to a new position (selected at random from NEWS directions) does not add or remove monomers from its bonding neighborhood (Fig. 4).

In this model an additional flexibility is achieved because neighbors can be "on top of each other" so that even the 3 × 3 bonding neighborhood allows local expansion and contraction. Even more interesting, it is possible to move all of the monomers in one space at the same time because both connectivity and excluded

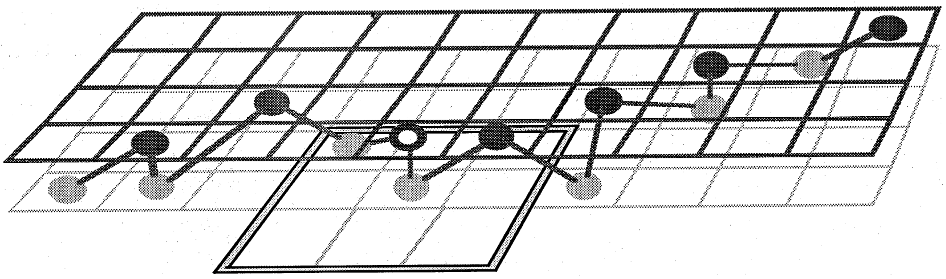

Fig. 3. Schematic illustration of a two-space polymer. In 2-D the two spaces are parallel planes. Monomers on the upper plane are shown as circles with dark shading, monomers on the lower plane are shown as circles with light shading. Along the polymer the monomers alternate spaces so that odd monomers are in one space (the light space) and even monomers in the other space (the dark space). Bonds are indicated by line segments between monomers. Monomers are bonded only to monomers in the other space. The 'bonding neighborhood' of each monomer is a 3 × 3 region of cells located in the opposite plane. The bonding neighborhood of the dark monomer marked with a dot is shown by the region with a double border. The two neighbors of this monomer, both light monomers, are located in the bonding neighborhood.

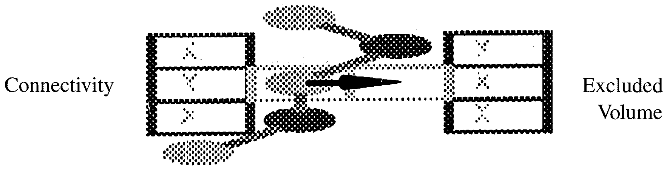

Fig. 4. Illustration of the movement of a monomer in the two-space algorithm. The movement of a light monomer requires checking connectivity and excluded volume in the dark space. The picture illustrates a move where the light monomer is to be moved to the right. To ensure that connectivity is not broken we check that no monomers are left behind. This is equivalent to checking that there are no dark monomers in the three cells marked with Xs on the left. To ensure that excluded volume is not violated is equivalent to checking that there are no dark monomers in the three cells marked with Xs on the right. If there are no monomers in these cells, then no monomers are removed from or added to the bonding neighborhood of the light monomer as a result of the move. In the picture the move is allowed.

volume are implemented through interactions with the other space. Thus 1/2 of the monomers may be updated in parallel.

To show that all the monomers in one space can be moved in parallel, we must show that their motion cannot result in either breaking the polymer or violating excluded volume. Since each monomer move preserves its bonded neighbors, the polymer cannot be broken. Excluded volume is different for two monomers within a space and for two monomers in opposite spaces. For two monomers in opposite spaces, excluded volume prevents monomers from entering each others' bonding neighborhood. For two monomers in the same space excluded volume is just the requirement that two monomers do not move onto the same site. They can be adjacent, since they are not within each others' bonding neighborhood. In a proof by contradiction that two monomers cannot move onto the same site, assume two monomers were to move to the same site. In this state they will have the same bonded neighbors. Since they start with different bonded neighbors, and the algorithm prevents monomers from changing their bonded neighbors, this cannot happen. There is only one exception, which we may avoid (or treat specially). For a polymer of length three the two end monomers have the same neighbor and they are not prevented from landing on the same site.

In order to preserve detailed balance, which is desirable for a Monte Carlo algorithm, we must choose at random which of the two planes to update at each step. The two plane algorithm may be implemented in 3-D by considering the polymer to be in a double space.

5. Advantages of the two-space algorithm and implementation details [13]

There are four ways in which the two-space algorithm excels: step speed (small number of computer operations per step), relaxation rate (small number of monomer steps per polymer relaxation time, as given by the small prefactor τ_1 in $\tau(N) \sim \tau_1 N^z$), inherent parallelism, and simplicity. The two-space algorithm is fast even on serial machines. The simplicity of the two-space algorithm can be appreciated when generalizing it to apply it to various problems.

The direct parallelism of the two-space algorithm lends it to implementation on a variety of computer architectures. Because we can update half of the polymer at a time there are two different ways to implement parallelism: space partitioning and polymer partitioning.

Space partitioning is the usual CA assignment of processors to different regions of space. Polymer partitioning is the assignment of processors to different parts of the polymer. Spatial assignment is particularly convenient when a simulation is performed with a high density of monomers. For example, there is considerable interest in simulations of entangled polymers at high densities (polymer melts). Polymer assignment is convenient when the polymer occupies only a small fraction of the space. This is the case for expanded isolated polymers, or problems that might include a single polymer moving in a static matrix.

Within each of the parallelization schemes, processor assignment may take advantage of full parallelization (fine graining), or task aggregation (coarse graining) may be used. Typically, parallel architectures suited for fine-grained simulations require fast intersite communication and large numbers of processors, while the processing elements themselves do not necessarily have to be very powerful. If the number of processors is too small a compiler may create "virtual processors" allowing the user to treat each site as being assigned to a separate dedicated processor. When coarse graining is used a whole region of lattice or of the polymer is assigned to a single processor, thus requiring fewer more powerful processors.

In fine-grained space partitioning each processor is assigned to one double-space lattice site. When coarse graining is used, a region of lattice is assigned to a processor. Communication is necessary to transfer information about the boundary regions and to transfer monomers across the boundaries of the spatial regions between processors.

Similarly, for polymer partitioning, fine-graining would assign one monomer to a processor. When coarse graining is used a group of monomers is assigned to a processor. The assignment may be of successive monomers along the chain or they may be gathered together under some other rule. In polymer partitioning there is need for interprocessor communication to construct a poster space – a data structure which is indexed by spatial location and represents by occupancy of a site the spatial locations of the monomers – to which the locations of monomers are "posted" by the processors which are keeping track of the locations of specific monomers. Thus far polymer partitioning of the two-space algorithm has not been implemented on parallel architectures, but the algorithm that would be used has been implemented on serial machines.

In the following sections we describe details of implementation of the two-space algorithm for both fine and coarse graining in space partitioning [15].

5.1. Fine-grained space partitioning

Fine-grained implementation assumes each lattice site is assigned a dedicated processor, thus allowing the algorithm to be easily implemented on SIMD machines. The system is updated in three steps:

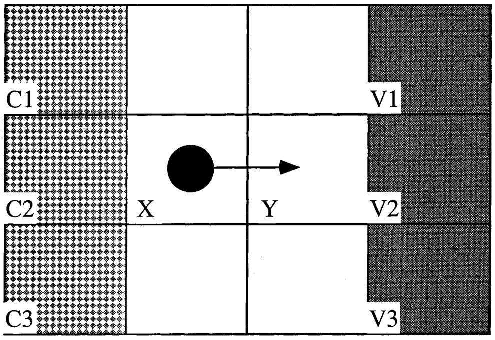

Fig. 5. In fine-grained simulation one processor is assigned to a site. Assume the processor assigned to X has a monomer in its odd plane that has elected to move westward. In order to make the step, processor X has to determine that there is nothing in the even plane of processors C1, C2 and C3 (to preserve connectivity) and nothing in the even plane of processors V1, V2 and V3 (to preserve excluded volume).

1. The even or odd plane is selected at random;
2. For each occupied site one of four compass directions is chosen at random;
3. A Monte Carlo step is performed subject to two constraints in the *other* plane: connectivity and excluded volume (Fig. 4, see also Fig. 5).

For each step a processor (lattice site) requires communication with 6 other processors in order to determine whether the move is allowed. The total number of communications depends on the connectivity of the processors. Assuming single step communication with any of the eight nearest neighbors, the total number of communications is 9. If the same operations are applied to all processors the total number of operations to enable movement of monomers in all four directions is 36. This number can be decreased by gathering the information in two stages as follows (see Fig. 5). First the data from C1, C2 and C3 are combined into C2 and V1, V2 and V3 are combined into V2, totaling four communications per plane update. Then the information may be transferred in three steps to X. Additional optimization is realized by noting that the same information may be used either at X for moving west or at Y for moving east. Thus data is collected in vertical and horizontal strips. The result of vertical (horizontal) data gathering is shifted left and right (up and down). The total number of communications per plane update is $4 + 4 * 3 = 16$, compared to the previous 36. Note that with this optimization only (NEWS) communications are needed. Finally, four more communications are needed to move the monomers for a total of 20. The number of communications is emphasized here because the communication to computation ratio is generally large, and, therefore, minimizing communications is extremely important. In the fine-grained implementation computation requires only random number generation, evaluation of a compound logical expression using communicated data, and clearing a variable when a monomer 'leaves' the site.

The fine-grained approach is especially efficient when dense systems are simulated. A illustrative test simulation performed some years ago is shown in Fig. 6 which compares the performance of the algorithm on a parallel computer (MP-1) vs. a

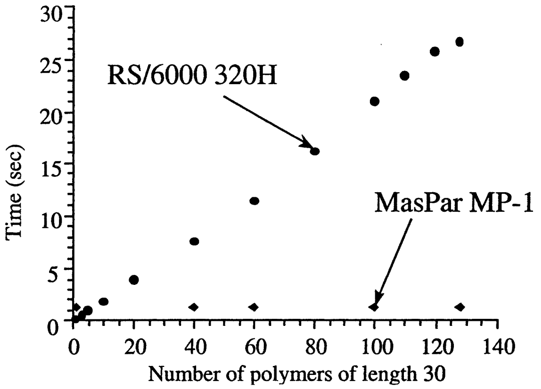

Fig. 6. Illustrative comparison (performed some years ago) of serial (IBM RS/6000 320H) and parallel (MP-1) architectures performing increasing density simulations of 104 space updates of a 128 × 64 lattice. List processing was used on the serial computer, fine-grained space processing on the MP-1.

serial machine (IBM RS/6000) by showing time required to execute 10^4 space updates as a function of a number of polymers of length 30. As the density of the system increases, the advantage of a parallel simulation becomes more and more pronounced.

Another kind of parallelization arises when we consider that the existence of a monomer on the lattice may be represented by only one bit. A lattice plane of the algorithm can be represented as a single-bit plane. It can be seen that more than one system can be simulated simultaneously (see Fig. 7) by programming the algorithm as a number of bitwise operations where one plane consists of the kth bits of all of the variables used to represent the space. This bit-wise parallelism enables an internal form of distributed simulation. For example, if the largest integer type on a computer is 64 bits (as in MP-1), then 64 systems can be updated in a single step. The speedup, however, is much less. We achieved a 12-fold speed up on the MP-1 due to the two factors: Communication time for 64-bit numbers is larger than it is for a 32-bit number, and the generation of a 64-bit random number takes twice as long as for a 32-bit number.

5.2. Coarse-grained parallelism

In architectures with powerful individual processors, task aggregation by coarse-graining enhances the performance. This is particularly important for machines with relatively slow communication and networks of workstations.

For coarse-grained space partitioning, each processor is assigned to a region of the lattice rather than to a single site. The basic structure is shown in Fig. 8 (for simplicity it is assumed that these regions are of rectangular shape). Individual processors are assigned to regions X, N1–N8. The following steps update a space:

1. One of two spaces is chosen at random.
2. All processors synchronize.

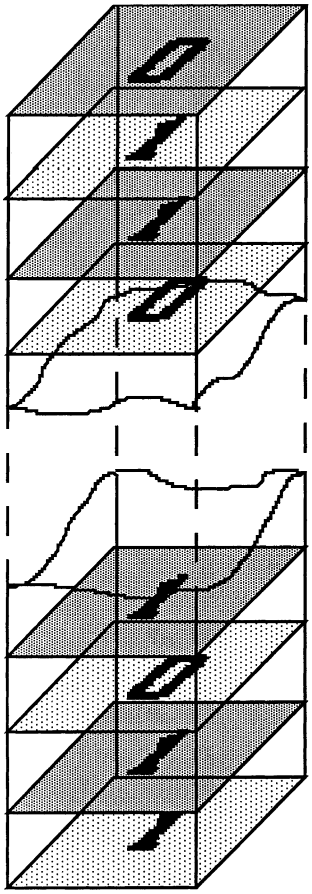

Fig. 7. Bit-plane parallelism on SIMD computers makes use of each bit of the longest data type for an independent simulation (see text).

3. Each processor updates the piece of lattice it is assigned to by either scanning all lattice sites in the region or processing the list of monomers that belong to that region. The former method is efficient for dense systems and is similar in its implementation to fine-graining. The latter better for to dilute systems.
4. Processor X receives information from its 8 neighbors N1–N8 about moves performed on monomers within two lattice units from its boundaries (shown in gray in Fig. 8). In list processing, monomer lists are updated to keep track of monomers entering and leaving the region.

This method does not require synchronization before each *site* update, but rather only before the update of a whole *space*. Therefore this implementation of the algorithm where each processor works independently during the space update is well suited to MIMD architectures. Since communication between processors only describes the boundary regions, the frequency and amount of communication is

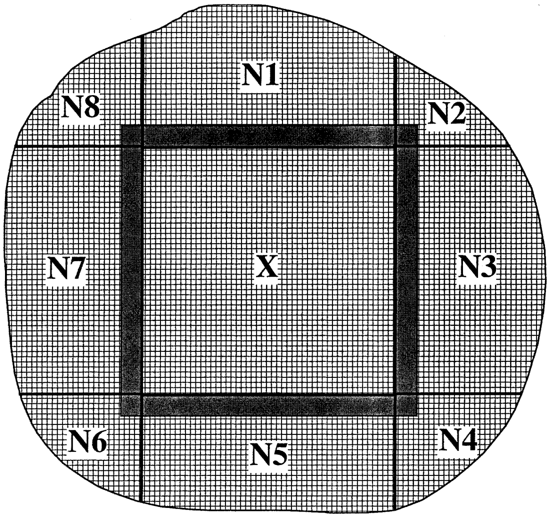

Fig. 8. Schematic drawing of the processor assignment for coarse grained simulations by space partitioning (see text).

reduced. Moreover, all of the information transferred between two processors before each space update may be combined into one message. However, adequately fast synchronization is required. Unlike SIMD, where each step is synchronous, explicit synchronization must be performed between space updates in a MIMD architecture and must be taken into account when performance issues are considered.

6. Tests of the two-space algorithm

To test the two-space algorithm we measure structural and dynamic properties. The simulations we perform for these tests are in 2-D. We measure the characteristic size of the polymer as given by the radius of gyration $R_g(N;t)$:

$$\begin{aligned} R_g(N;t)^2 &= \frac{1}{N}\sum_i (r_i(t) - r_{cm}(t))^2, \\ r(t)_{cm} &= \frac{1}{N}\sum_i r_i(t). \end{aligned} \tag{1}$$

To initialize the simulation we start from a straight polymer. The test shown in Fig. 9 consists of a two-dimensional polymer simulated with $N = 140$ monomers. We see that after a few steps the polymer fluctuates around an average polymer size that we can calculate as a time average, $R_g = 18.39$, indicated by the horizontal line. It is better to leave out the first part of the simulation in calculating the average.

To see how $R_g(N)$ varies with N, we use a log–log plot (Fig. 10) which gives a straight line for large N. Thus it follows a power law behavior where the slope of the line is the value of the exponent. The exponent is in agreement with the expected result $R_g(N) \sim N^\nu$, $\nu = 0.75$ is the Flory exponent in 2-D.

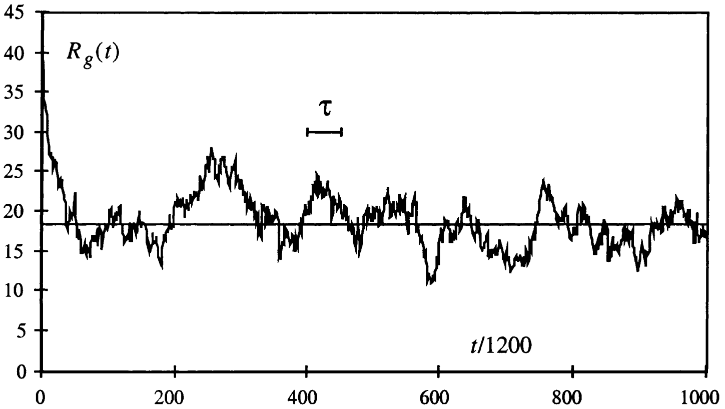

Fig. 9. Plot of the characteristic polymer size, the radius of gyration, $R_g(t)$ as a function of time in a Monte Carlo simulation of the two-space algorithm. The two-dimensional polymer simulated has $N = 140$ monomers. The simulation starts from a completely straight conformation which has an unusually large size. After relaxation, the radius of gyration fluctuates around the average value, $R_g = 18.39$ indicated by the horizontal line. The characteristic time over which the polymer conformation relaxes τ is the correlation time of the radius of gyration indicated by the horizontal bar. The values plotted of the radius of gyration are sampled every 1200 plane updates. There are about 50 samples in a relaxation time.

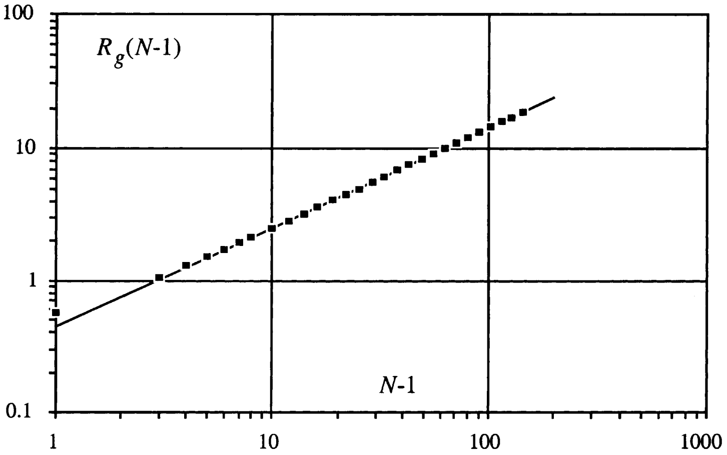

Fig. 10. Plot of the average radius of gyration as a function of polymer length for the two-space algorithm in 2-D. The average values are obtained by simulations like that shown in Fig. 9 using 100,000 samples and without including the first 100 samples. The horizontal axis is the number of links, $N - 1$ in the chain. The line in the figure is fitted to the data above $N = 10$ and has a slope of 0.756. This is close to the exact asymptotic scaling exponent for long polymers, $\nu = 0.75$.

The second test is to evaluate the dynamics of relaxation of the polymer. We can see from Fig. 9 that there is a characteristic time over which $R_g(N; t)$ fluctuates shown by the horizontal bar on the plot. The values plotted of the radius of gyration are

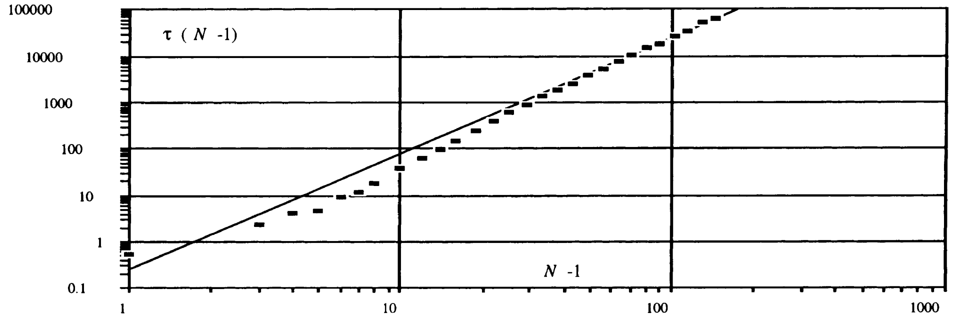

Fig. 11. Plot of the relaxation time τ of polymers as a function of the number of links $N-1$ for the two-space algorithm in 2-D. The line in the figure is fitted to the last four points that are relaxation times for polymers longer than $N=100$. The slope of this line is 2.51, which is consistent with the exponent expected for Rouse relaxation, $2\nu+1=2.5$.

sampled every 1200 plane updates. There are about 50 samples in a relaxation time. Over shorter times, $R_g(N;t)$ values are correlated. Over longer times they are independent. This characteristic time is the relaxation time, $\tau(N)$. To find a value for the relaxation time we study the correlation of $R_g(N;t)$ (for simplicity the dependence on N is not indicated):

$$A[R_g(t)](\Delta t)=\frac{\left\langle (R_g(t+\Delta t)-R_g)(R_g(t)-R_g)\right\rangle}{\left\langle (R_g(t)-R_g)^2\right\rangle},\qquad (2)$$

$$R_g=\left\langle R_g(t)\right\rangle.$$

The averages are over time. This measures the relationship between $R_g(t)$ and $R_g(t+\Delta t)$, as a function of Δt. The behavior of the correlation function can be readily understood. For $\Delta t=0$ it is one. For large Δt, when the value of $R_g(t+\Delta t)$ is independent of $R_g(t)$, the average in the numerator is the product of the averages of the two factors independently. Since the average of either factor is zero, the correlation function is zero. $\tau(N)$ is the time at which the correlation falls to $1/e$.

A plot of the characteristic relaxation time $\tau(N)$ as a function of the polymer length is shown in Fig. 11. $\tau(N)$ increases with length and for long enough polymers it agrees with $\tau(N)\sim N^z$, $z=2.5$. This is the Rouse prediction expected to apply to long polymers when the fluid motion (hydrodynamics) is not included, and therefore is the correct result for our simulations.

7. Application to 2-D melts

A unique feature of polymer solutions in 3-D compared to other states of condensed matter is the existence of a semi-dilute regime of polymer concentration in which chains interpenetrate and become entangled. This happens when the volume fraction c of the space occupied by a polymer exceeds the overlap volume fraction c^* which can be arbitrarily small for long polymers ($c^* \sim N/R_g^d \sim 1/N^{d\nu-1}$, where N is the number of monomers, $R_g \sim N^\nu$ the radius of an isolated chain, ν the Flory exponent, and d is the dimension of space). In contrast, it is assumed that interpenetration does not take place in 2-D. The chains are believed to become progressively segregated into compact disks as the concentration is increased above c^* [3]. We show by extensive computer simulations that the 2-D melt behavior cannot be so simply described [13,14]. The actual behavior, whose precise nature remains undetermined, is intermediate between segregation and interpenetration.

As melt density increases, the radius of gyration R_g of a polymer decreases. This can be understood in various ways. In 3-D the decrease in size results from interpenetration which prevents (screens) interactions between monomers of the same polymer. This results in a shrinkage of the polymer to a size characteristic of a random walk, $R_g \sim N^\nu$, $\nu = 1/2$, which is smaller than the random walk with excluded volume, where $\nu = 0.6$. Thus, for interpenetrating polymers the structural properties correspond to polymers without excluded volume. This is quite different from what is thought to occur in 2-D. In 2-D polymers are believed to form collapsed disks implying that the excluded volume plays an essential role in determining the size of the polymer. However in 2-D there is a coincidence of two theoretical exponents for the radius of gyration. Both a collapsed disk and a random walk have the same value of $\nu = 1/2$. This opens the possibility that more subtle effects can play a role in the polymer structure.

We performed detailed computer simulations of 2-D polymer solutions on a 128×64 lattice with periodic boundary conditions. The maximal concentration (area fraction) allowed by the two-space algorithm is $\rho_{\max} = 3/4$, because of local excluded volume constraints. Therefore the concentration in the simulation was defined as $c = \rho/\rho_{\max} = (4/3) \times QN/(128 \times 64)$, where Q is the number of polymers on the lattice. Runs in the concentration range $0.163 < c < 0.651$ and for chain lengths $50 < N < 333$ were performed.

Figs. 12 and 13 display various properties of the system discussed in the following paragraphs. At high densities R_g follows a power law scaling as a function of N. The exponent can be seen from Fig. 12(a) which shows a nearly constant R_g^2/N consistent with either random walk or disk. As a function of density in Fig. 12(b) the polymers decrease rapidly in size.

A first indication that the polymers do not become compact disks arises by considering the polymer size at maximal density. In Fig. 12(b) the horizontal axis is the concentration c, so the maximal possible value corresponds to unity. At high density R_g^2/N must equal a universal constant. This constant can be calculated for the compact disk as well as for the random walk. For a compact disk at the maximal density $\rho_{\max}$ the number of monomers in the polymer is the area of the disk times the maximal density

$$N = A\rho_{\max} = \pi R_{\max}^2 \rho_{\max}, \tag{3}$$

where $R_{\max}$ is disk radius, N is the length of the polymer and A is the area. The radius of gyration R_g can be expressed as

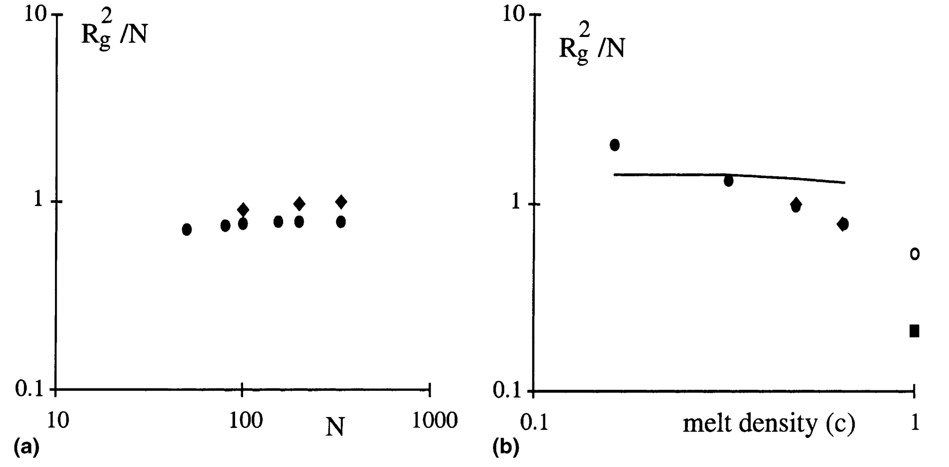

Fig. 12. Average radius of gyration of polymers in a 2-D melt. The scaling $R_g \propto N^{1/2}$ predicted both for segregated compact disks and random walks, is apparent when R_g^2 is divided by chain length N. (a) Shows that the N dependence becomes very weak. (l c =0.65; u c =0.48). (b) Shows the concentration dependence (l N = 200; u N = 333, which nearly overlap). There are two points shown at concentration c = 1. The open circle (*m*) is an extrapolation of the measured values. The solid square (*n*) shows the value expected for compact disks. The extrapolated value from the simulation is approximately 2.5 times higher. The nearly horizontal line shows the values expected from a random walk. It varies slightly with density. The line is approximately 2.5 times higher than the extrapolated value.

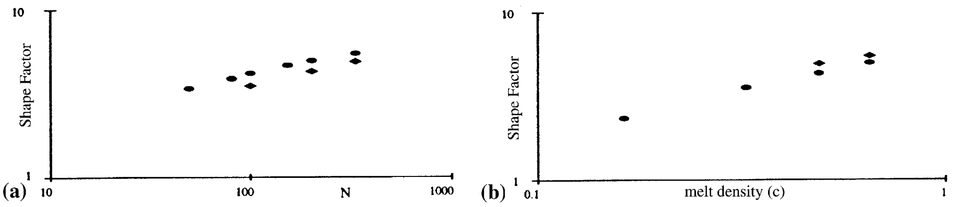

Fig. 13. The shape factor S is plotted (linear axes) as a function of the same parameters as used in Fig. 9. The shape factor measures directly polymer interpenetration. It is expected to be unity for completely segregated disks and this is the lowest possible value. The increasing shape factor as a function of polymer density indicates that significant interpenetration occurs and increases as a function of density.

$$R_g^2 = \frac{1}{N}\int_0^{R_{\max}} \rho_{\max} r^2 2\pi r \, \mathrm{d}r. \tag{4}$$

From Eqs. (3) and (4) it follows that for compact disk

$$\frac{R_g^2}{N} = \frac{1}{2\pi\rho_{\max}} = 0.21. \tag{5}$$

For random walks the radius of gyration can be written as $R_g = \sqrt{N\sigma}$, where σ is the root mean square distance traveled in an elementary step. Thus,

$$\frac{R_{\rm g}^2}{N} = \sigma^2 = P_1 + 2P_{\sqrt{2}}, \tag{6}$$

where P_1 is the probability of moving along a horizontal or vertical lattice direction and $P_{\sqrt{2}}$ is the probability of moving along diagonals. These probabilities were calculated during the simulations and are weak functions of the density. At low densities $\sigma^2 \sim 1.4$ while at the highest densities we studied $\sigma^2 \sim 1.3$. From Fig. 12(b) we see that the measured extrapolated $R_{\rm g}^2/N$ is distinct from the values expected for either compact disk or random walk. Remarkably, it lies half-way (on a logarithmic scale) between the two expected values.

In order to make further progress there is need for a measure of the polymer properties that can clearly distinguish the degree of polymer interpenetration. For this we compare the area to the perimeter of the region occupied by the polymer. The shape factor S is defined to be the ratio of the perimeter squared of the polymer divided by its area, $S = P^2/A$. S should be much smaller for segregated disks than for interpenetrating polymers. To measure S, we use an algorithm [16] that fills in the area surrounding each polymers until all space is filled. Then the perimeter of the space associated with each polymer is counted, as is the area.

We normalize the shape factor plotted in Fig. 13 so that the minimum possible value is unity. The shape factor increases as a function of density implying polymer interpenetration. Fig. 14 shows a frame taken from a simulation of polymers of length $N = 333$ at the highest density $c = 0.65$. Distinct individual polymers are highlighted in each of the six illustrations. There is clear indication that the polymers are sometimes compact and sometimes expanded. When time lapse pictures are taken it can be seen that individual polymers change from expanded to contracted.

In conclusion, contrary to expectations that polymers will segregate into compact disks at high densities, the chains adopt both compact and expanded conformations.

8. Application to polymer collapse

The flexible polymer coil to globule transition (polymer collapse) is an extensively investigated and fundamental aspect of the properties of polymers in dilute solution. When polymers are dissolved in liquids there are essentially two possible structures: either the polymer collapses into a compact structure, or the polymer is expanded. The transition (collapse) is driven by changes in the affinity of the monomers for each other compared to their affinity for the solvent. A polymer in its expanded state is said to be in a good solvent. A compact polymer is said to be in a poor solvent. The transition is called the θ-point. When a protein is folded and unfolded in solution it is crossing the line between compact and expanded structures. DNA also undergoes a transition between compact and expanded forms in order to allow transcription or replication.

Analytic arguments as well as Monte Carlo simulations have investigated the equilibrium structure of various model polymers as a function of the effective

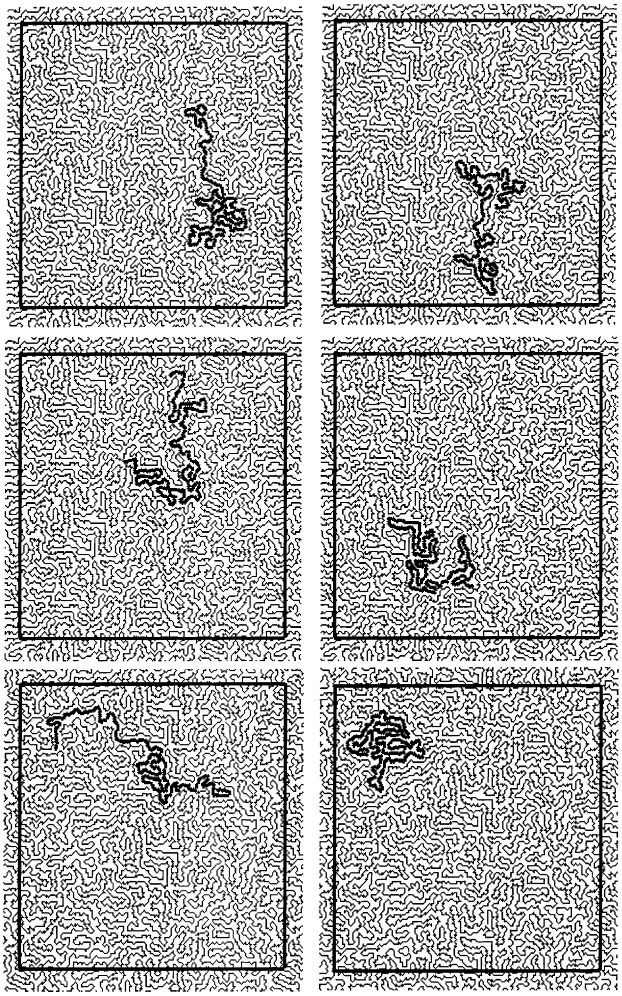

Fig. 14. A frame "snapshot" from a 2-D melt simulation on the MP-1 parallel computer showing polymers of length 333 at a density of 0.65. Each picture highlights a distinct polymer. Contrary to expectations that polymers will segregate into compact disks at high densities, the chains adopt both compact and expanded conformations. This is consistent with the analysis of averaged quantities described in Figs. 12 and 13.

temperature or relative solvent affinity through the transition [4]. The study of simple model polymer collapse is not relevant to the problem of determining the end structure in biopolymer collapse – what is generally considered the protein folding problem. However, the *kinetics* of the coil to globule transition may be relevant according to the molten globule concept [10]. According to this picture, in its most general form, initially a protein collapses into a globule which then reorganizes into a specific structure. It is the initial stage of the collapse that may be modeled by the kinetics of the coil to globule transition. Despite the extensive investigations of the equilibrium properties of collapse, little is known about kinetics of this transition.

The collapse of a polymer is controlled by the difference of the temperature $\Delta T = \theta - T$ from the θ-point temperature. The lower the temperature the more rapid the collapse, and the more important the kinetic effects. The process of collapse involves many encounters between monomers that form weak bonds to each other, like hydrogen bonds. Some of these might break free and others form instead. The larger ΔT is, the smaller is the probability that a bond will break. The bonds that are formed build larger and larger aggregates. The local formation and breaking of a single bond is less relevant than the formation of larger clusters. The possibility of breaking up a cluster becomes less likely for large clusters because the total bonding energy is large. This means that we can treat the kinetics of collapse by considering irreversible bonding without bond breaking. More formally, it is possible to prove[4] that the parameter that actually controls the collapse for long polymers is $\Delta TN^{1/2}$. Thus we can always think about the process of collapse as if it occurs for large values of ΔT, and bond breaking is unimportant.

We investigate the behavior of polymer collapse using the two-space algorithm [1,5,11–13]. Starting from an initial expanded (equilibrium) conformation, polymer collapse is simulated by eliminating the excluded volume constraint. We discuss below why excluded volume should not be essential during collapse, even though it is necessary for the original polymer conformation. Once the excluded volume constraint is eliminated, the usual monomer Monte Carlo steps are taken. Monomers are no longer prevented from entering the neighborhood of another monomer, however, they continue to be required not to leave any neighbors behind. This enables monomers of the same type (odd or even) to move on top of each other. Once a monomer moves onto another monomer they loose separate identity and become an aggregate that moves as a unit. We keep track of the mass M of an aggregate, which is the total number of monomers that reside on the same site. A diffusion constant is assigned to the aggregate according to $D \sim 1/R \sim 1/M^{1/d}$. This is the diffusion constant according to hydrodynamics (Stokes' law) in 3-D, and in 2-D this is the diffusion constant of a polymer trapped at a fluid interface. By incorporating Stokes' law into the collapse we have incorporated the primary effect of hydrodynamics when there are aggregates present. We implement the diffusion constant by controlling the probability of making a hop when an aggregate is selected to move. We choose the time scale by setting to one the probability that a single monomer will move when chosen.

The polymer dynamics are then simulated by selecting an aggregate at random (monomers are included as aggregates of mass 1) and moving the aggregate in one of four compass directions with a probability given by the diffusion constant and only if the connectivity constraint allows – the aggregate does not leave any neighbors behind. In order to move the aggregate with a probability given by its diffusion constant, a random number ranging between zero and one is compared with the diffusion constant. The monomer is moved only if the random number is smaller than the diffusion constant.

Time is normally measured in a Monte Carlo simulation by choosing N monomers to move in one time interval. During collapse, a time interval consists of choosing a number of aggregates that is equal to the number of remaining aggregates. Since the number of aggregates can change during the time interval, it is arbitrarily taken to be the number at the end of the time interval. When the number of moves exceeds the number of aggregates a new time interval is started.

A sequence of frames from a simulation in 2-D is shown in Fig. 15. Most striking in these pictures is that the ends of the polymer have a special role in the collapse. The ends diffuse along the contour of the polymer eating up monomers and smaller aggregates until the two end aggregates meet in the middle. Along the contour, away from the ends, the polymer becomes progressively smoother. The polymer becomes more and more like a dumbbell. We call this process end-dominated collapse. A similar behavior has been confirmed for various other abstract polymer models that include excluded volume in 3-D [1,12].

To understand these observations we consider qualitatively the process of polymer collapse. We realize that encounters of monomers that are distant from each other along the contour of the chain are unlikely because they are also, on average, distant from each other in space. We can therefore consider aggregation as primarily a local process, where a monomer forms an aggregate with neighboring monomers along the contour. This aggregation is, however, inhibited by the existing bonds. Aggregation occurs when two monomers that are near each other in space move close enough to form a new bond. The easiest aggregation would occur if a monomer could move to aggregate with one of its neighbors, however the neighbor on the other side prevents this, because stepping away from the other neighbor would break an existing bond. Without curvature in the chain, the monomer is unable to move to aggregate with either neighbor, because it is bonded to the neighbor on the other side. If there is some curvature, then monomers can aggregate. The aggregation would cause the curvature to decrease and further aggregation becomes more difficult. The same argument applies if we consider a monomer moving to bond to its second or third neighbors along the contour. These problems do not occur at the ends of the polymer. The ends, because they have only one neighbor, can move to aggregate with the monomers that are near them along the contour. Thus, during collapse the aggregates at the ends grow more rapidly than aggregates along the contour, and eventually the polymer looks like a dumbbell. This is what was found in the simulations.

In order to understand this more fully we develop an analytic (scaling) theory that compares the growth of the end mass $M_0(t)$, with that of the average aggregate mass not including the ends $M(t)$. In a small time interval each end aggregate has a probability proportional to its diffusion constant $D_0(t)$ of collecting more mass by moving toward and accreting its immediate neighbor aggregate. This neighbor has an average mass $M(t)$ and is a distance a away, where a is approximately a monomer–monomer distance and does not depend on time. Thus, on average $M_0(t)$ grows according to

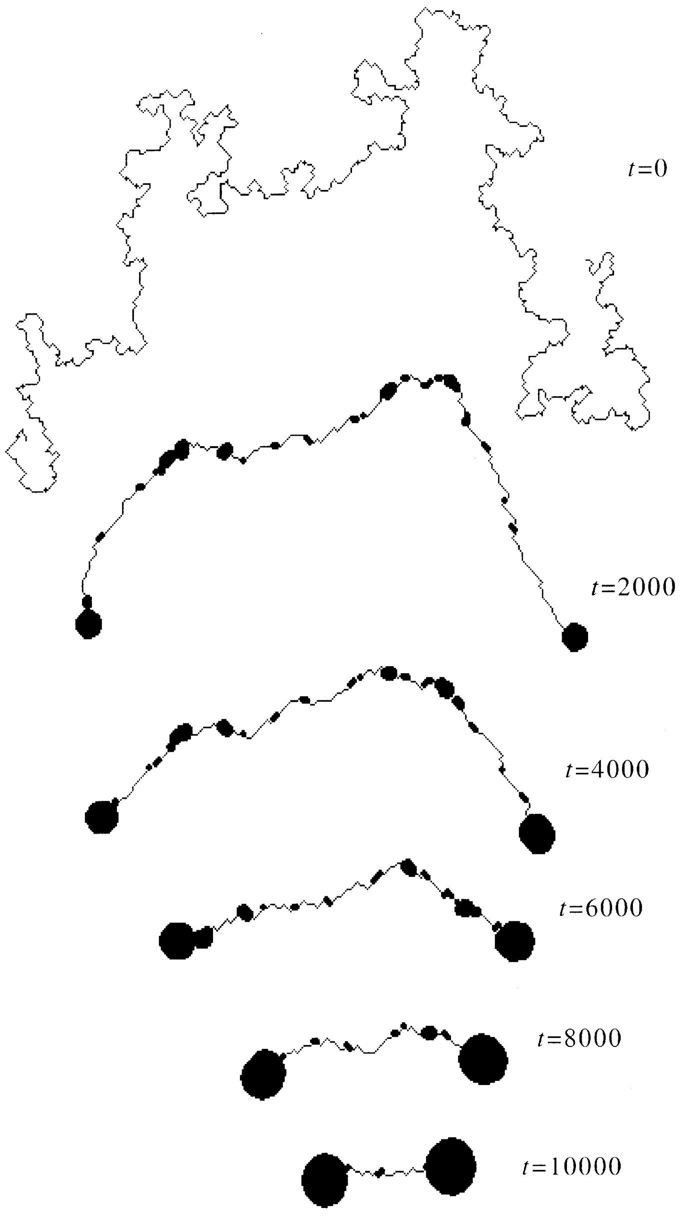

Fig. 15. Frames ‘snapshots’ of the collapse of a single polymer of length $N = 500$ monomers in 2-D. Each aggregate is shown by a dot. The area of the dot is the mass of the aggregate. This does not reflect the excluded volume of the aggregates which is zero during collapse. Successive snapshots are taken at intervals of 2000 time steps – approximately 1/6 of the collapse time. The initial configuration is at the top. The pictures demonstrate the end dominated collapse process where the ends diffuse along the contour of the polymer accreting small aggregates.

$$\frac{\mathrm{d}M_0(t)}{\mathrm{d}t} \propto M(t)D_0(t)/a^2. \tag{7}$$

We assume that the two quantities $M(t)$ and $M_0(t)$ follow a power law scaling

$$M_0(t) \propto t^{s_0}, \quad M(t) \propto t^s. \tag{8}$$

Inserting this and $D_0(t) \propto 1/M_0(t)^{1/d}$ into Eq. (7), we ignore prefactors and set the exponents on both sides equal. Solving for s_0 we obtain

$$s_0 = (s+1)d/(d+1). \tag{9}$$

The two exponents are equal when $s = s_0 = d$. We will find that s is much smaller than d, and therefore s_0 is much larger than s – the end mass grows faster than the mass along the contour.

The value of s may be estimated by considering collapse of a polymer with fixed ends at their average equilibrium separation. This removes the dynamics of the end motion from the problem. The collapse of this fixed end polymer would result in a straight rod of aggregates with an average mass $M = N/R$, where N is the number of original monomers and $R \sim N^v$ is the average end-to-end distance of the original polymer, which is also the length of the resulting rod. Since we have eliminated the special effects of the ends, the time over which the collapse occurs can be approximated roughly by the usual dynamics of a polymer with $\tau \sim N^z$, where $z = 3v$ when hydrodynamics is included (Zimm relaxation), or $z = 2v + 1$ without hydrodynamics (Rouse relaxation). We think about the fixed-end polymer collapse as a simple model for the collapse of the original polymer along the contour away from the ends. We are thus assuming that the polymer can be approximated locally by polymer segments whose ends are pinned. Over time, progressively longer segments are able to relax to rods. The average mass of the aggregates at a particular time t is then determined by the maximal segment length $N(t) \sim t^{1/z}$ that relaxes by time t. This means that

$$M(t) \sim N(t)/R(t) \sim N(t)^{1-v} \sim t^{(1-v)/z} \tag{10}$$

or

$$s = (1-v)/z. \tag{11}$$

The value of s obtained from this argument depends on the value of v and z, but for all reasonable values of these exponents, it is small. For example, for $v = 0.6$ and $z = 2v + 1$ (Rouse dynamics) we obtain $s = 0.18$, for $z = 3v$ (Zimm dynamics) we obtain $s = 0.22$. Using Eq. (9) we also find that s_0 is much larger than s.

We analyze the simulations to compare with the scaling argument. The results, shown in Fig. 16, were obtained in both 2-D and 3-D. In 2-D for polymers of length $N = 1000$ (averaged over 500 samples), and $N = 500$ (1000 samples). In 3-D for polymers of length $N = 500$ (500 samples), and $N = 250$ (1000 samples). It is apparent from the figure that the collapse of longer polymers follows precisely the same curves as the collapse of the shorter polymers but extend the curves to longer times. This is consistent with the picture of end-dominated collapse, where the only effect of a longer polymer is increasing the length of time till the end aggregates meet in the

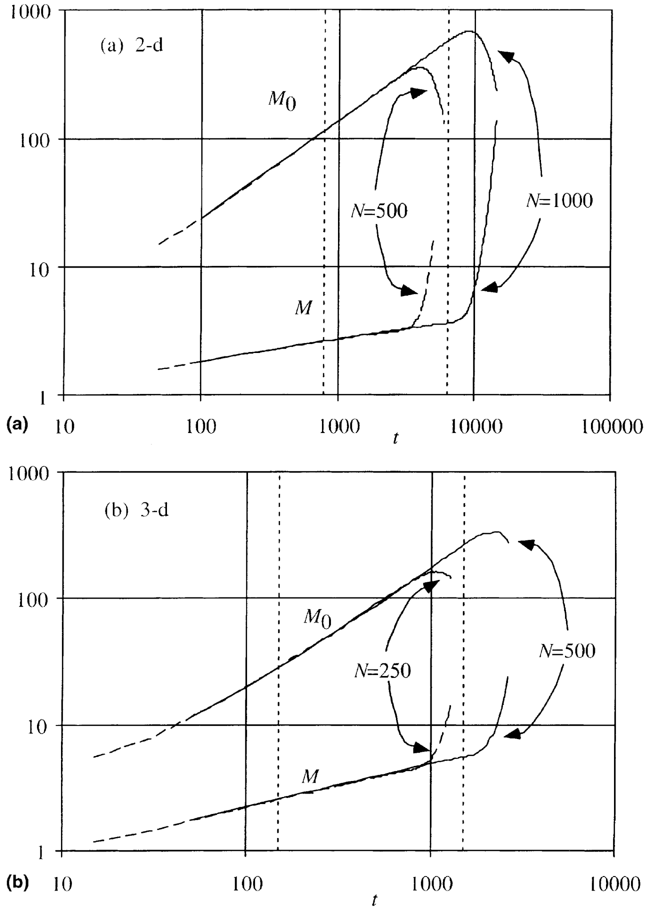

Fig. 16. Plot of the time evolution during polymer collapse of the average total mass of the polymer ends $M_0(t)$ and of the average mass $M(t)$ of aggregates not including the ends. (a) Shows collapse in 2-D of polymers of length $N = 1000$ (averaged over 500 samples), and $N = 500$ (1000 samples). (b) Shows collapse in 3-D of polymers of length $N = 500$ (500 samples), and $N = 250$ (1000 samples). Longer polymers follow the collapse of the shorter polymers but extend the curves to longer times. This is consistent with the picture of end-dominated collapse, where the only effect of a longer polymer is increasing the length of time till the end aggregates meet in the middle. Scaling exponents fitted to are given in II and discussed in the text.

middle. Both $M(t)$ and $M_0(t)$ follow power law scaling. Exponents are given in Table 1. The value of s obtained from the simulation is in qualitative but not quantitative agreement with the analytic argument. Since our derivation of s relied upon additional simplifying assumptions and depends upon v and z we can accept the disagreement without abandoning the basic argument. The 3-D result would be in agreement if we used $v = 0.5$ and $z = 3v$ but this must be a coincidence. The relationship given by Eq. (9) between s_0 and s is satisfied, and s_0 is larger than s.

Table 1
Power law exponents for the scaling of end mass (s_0) and mass along the contour (s) during polymer collapse[a]

	s	s_0	$s_0(s)$ (Eq. (9))	$s_0 - s_0(s)$
2-D	0.154 ± 0.001	0.7734 ± 0.0006	0.7695 ± 0.0006	0.004 ± 0.001 (0.5%)
3-D	0.337 ± 0.002	0.982 ± 0.005	1.003 ± 0.002	-0.021 ± 0.005 (2%)

[a] The first column gives the dimension of space. The second and third columns are fitted to the simulation results between the dashed lines in Fig. 16. Fits were chosen to minimize standard errors. Errors given are only statistical – they reflect the standard deviation of the simulation data around the fitted line. The simulation results are compared to the scaling relation, Eq. (9), in columns four and five.

In our simulations, during polymer collapse, we neglected excluded volume. We know that excluded volume is relevant to the initial polymer conformation in good solvent. Moreover, excluded volume is also relevant to the final collapsed state of the polymer – without excluded volume the polymer collapses to a point. However, excluded volume does not enter in the scaling argument leading to the relationship between s and s_0. This relationship depends only on the dimension d of the space. Thus, we do not expect excluded volume to affect qualitatively the behavior of collapse. On the other hand, the value of s in Eq. (11) is dependent on the values of the exponents v and z. This means that we can expect the precise values of s and s_0 to be somewhat more sensitive to the presence of an excluded volume. Simulations with other model polymers confirm that the excluded volume affects the value of the exponents (in 3-D s changes from about 1/3 to 1/2, and s_0 from 1.0 to 1.1), but not the overall behavior.

We can analyze our results to give the scaling of the collapse time $\tau_c(N)$. This is the time that passes until each end aggregates has half of the mass of the polymer

$$M_0(\tau_c(N)) \propto N. \tag{12}$$

Substituting Eq. (8) we obtain

$$\tau_c(N) \sim N^{1/s_0}. \tag{13}$$

From the simulations we find that $1/s_0 = 1.293 \pm 0.001$ in 2-D and 1.018 ± 0.005 in 3-D (errors are statistical). Thus the collapse time is predicted to scale linearly with polymer length in 3-D. This indicates that kinetic effects through end-aggregation accelerate the collapse from the usual relaxation time scaling of $\tau(N) \sim N^z$, where $z \approx 2$.

We can extend these arguments to consider the behavior of polymer collapse away from the polymer ends along the polymer contour [5]. A completely straight segment of polymer does not allow aggregation because no monomer can move to bond with another monomer. In contrast, highly curved regions are more flexible and monomers in these regions may aggregate. Aggregation in a curved region reduces the contour length and the polymer becomes straighter, smoothing the rough fractal polymer structure. We therefore expect that the scaling exponent will increase over time. At long enough times the scaling will approach that of a straight line ($r \sim l$). However, this smoothing occurs first at the shortest length scales. In effect the polymer structure becomes consistent with a progressively longer persistence length. Assuming scaling, we anticipate that the polymer end-to-end distance for a polymer segment away from the ends of contour length l will follow the dynamic scaling formula

$$r = lf(t/l^z). \tag{14}$$

The universal function $f(x)$ is a constant for large values of its argument (long times), so that $r \sim l$ and scales as $x^{(1-\nu)/z}$ for small values of its argument, so that $r \sim l^\nu$ at $t \to 0$. The characteristic time at which crossover occurs is given by $\tau \sim l^z$. The dynamic exponent z is assumed to be consistent with conventional Zimm relaxation, $z = 3\nu$ Finally, we can also rewrite this scaling relation in terms of the number of monomers n in a polymer segment. Since the average mass along the

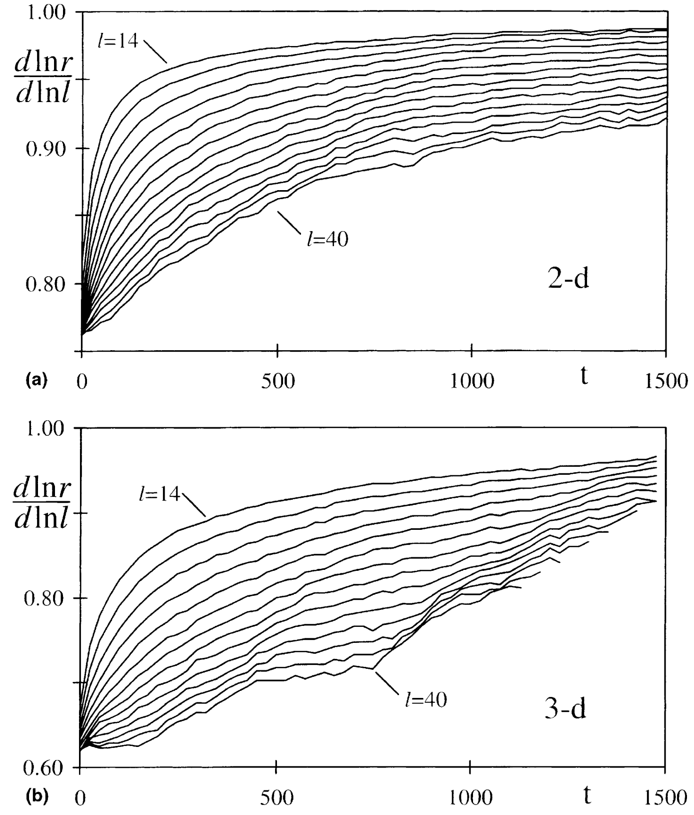

Fig. 17. Plot of the scaling exponent $d\ln r/d\ln l$ where r is the internal polymer segment end-to-end distance and l, is the segment contour length in both (a) two dimensions (2-D), for $t = 0$, 25, 50, 100, 200, 400, 800, 1600 and (b) three dimensions (3-D), with $t = 0$, 25, 50, 100, 200, 400, 800. In both cases the polymer contained 500 monomers and results were averaged over 200 collapses. Slopes are average slopes over segments of length of 20.

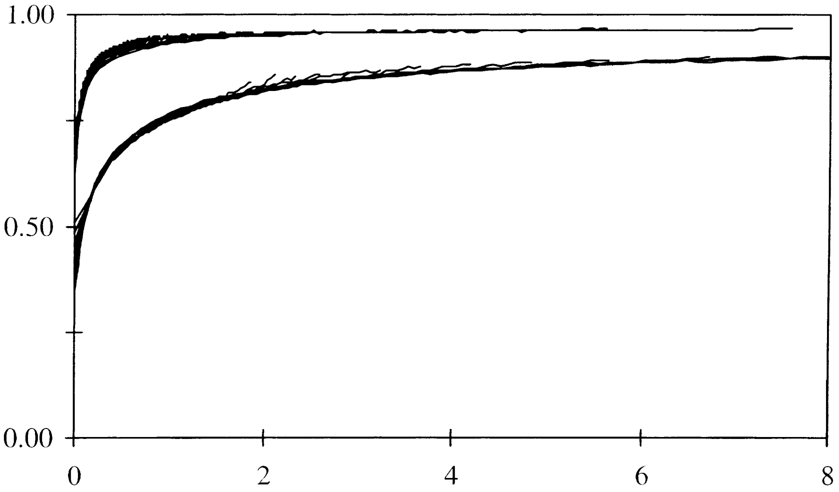

Fig. 18. Plot of the rescaled end-to-end polymer segment distance, r/l as a function of the rescaled time, $t/l^z d$ and 3-D are shown. The coincidence of the curves is consistent with validity of the universal scaling relationship, Eq. (1).

contour is $M \sim n/l$ and M follows power law scaling $M \sim t^s$ – we substitute $l \sim nt^{-s}$ in Eq. (14) to obtain $r(n,t)$.

Fig. 17 shows the derivative, obtained from finite differences, of contour length (l) versus end-to-end distance (r) for both 2-D and 3-D simulations, as a function of time for different segment contour lengths. For all segment lengths the derivative starts at approximately v, consistent with $r \sim l^v$ for a self avoiding random walk. As time progresses the polymer becomes smooth resulting in a slope that approaches 1 as the collapse proceeds. The rate of collapse becomes progressively slower as l increases. The scaling relation, Eq. (14), predicts that the relaxation time will scale with l as l^z. Fig. 18 shows the data following rescaling. $r/l$is plotted against the rescaled time, t/l^z. The generally good coincidence of the different curves confirms that the simulation obeys Eq. (14). An attempt to use an asymptotic scaling exponent, $r \sim l^u$, with $u = 0.95$, led to a visibly poorer fit, as did small variations in the exponent z.

9. Implications for protein folding

Can we relate our discussion of polymer collapse to the problem of protein folding? Unlike a general uniform collapse of the polymer, the end-dominated collapse proceeds by an orderly process of sequential monomer encounters. These encounters build up the aggregate compact structure (globule) in a manner that is not entirely random. A consequence of this orderly kinetic process is that the resulting globules are selected from a limited subset of all possible globules. This may simplify the process of arriving at a specific folded protein structure. End-dominated collapse is consistent with the molten globule model for the kinetics of folding. It suggests that there is a fast initial process of forming a compact globule followed by a rearrangement of the globule to form the final folded protein. The rearrangement process should take more time than the collapse. Unlike the collapse, this process requires bonded segments of polymer to move around each other, which is a much more difficult dynamic process. The significance of the end-dominated collapse is that by preselecting the initial compact globule, the rearrangement process is shortened and does not necessarily explore all possible compact conformations of the protein before settling in the desired state.

It is not easy to see what the precise nature of the globules that are formed by end-dominated collapse. However, we can note that they are likely to be formed out of two parts corresponding to the aggregate formed from one end, and the aggregate formed from the other. A more subtle feature of this process is that the globule is likely to contain fewer knots than would be generally found in a globule. This is because the diffusive end motion tends to unknot the polymer, since the ends are passed through any knots rather than closing or tightening them. To discuss this formally would require defining knots in a polymer with free ends, which is feasible but tricky.

We note that it has yet to be demonstrated experimentally that kinetics plays a significant role in protein folding, or in DNA aggregation. There is an interesting consequence of end-dominated collapse that has relevance to experimental tests. End-dominated collapse represents a significant departure from the usual rule of thumb that linear and ring polymer dynamics are similar. The simulations indicate that ring collapse should be significantly slower than linear polymer collapse. This is one of the ways that the predictions may be tested by experiment.

10. Conclusions

We have designed and tested a number of CA algorithms for parallel polymer simulation. One of these, the two-space algorithm, is a special algorithm that partitions the polymer into two spaces. Since interactions and explicit constraints act between the spaces, all elements within a space can be updated in parallel. This is particularly useful for treatment of the excluded volume interactions. Other advantages in implementation of this simple algorithm were discussed.

Scaling behavior of polymers in the two-space algorithm were tested and confirmed to satisfy the known scaling of long polymers. It was then applied to study the behavior of high density polymer melts in 2-D and polymer collapse. In both cases important new results are reported.

The behavior of polymers in 2-D melts, is shown to consist of both expanded and compact configurations. This is counter to the conventional wisdom which assumes that polymers will be compact in 2-D melts.

In the study of polymer collapse it is shown that the polymer ends play a surprisingly important role. Because they can aggregate more easily than other parts of the chain, the ends form rapidly growing aggregates that diffuse along the contour and accrete monomers and other smaller aggregates. The orderly process of sequential monomer encounters suggests the compact structure (globule) is formed in a manner that is not entirely random. In considering the role of the kinetics of collapse in protein folding it can be suggested from these results that the orderly kinetic process may simplify the process of reliably arriving at a specific folded protein structure.

References

[1] Y. Bar-Yam, Dynamics of Complex Systems, Perseus Press, Cambridge, 1997.

[2] K. Binder, (Ed.), Applications of Monte Carlo Methods in Statistical Physics, second ed., Springer, Berlin, 1987.

[3] I. Carmesin, K. Kremer, Static and dynamic properties of two-dimensional polymer melts, J. Phys. France 51 (1990) 915.

[4] H.S. Chan, K.A. Dill, Polymer principles in protein structure and stability, Annu. Rev. Biophys. Biophys. Chem. 20 (1991) 447.

[5] G. Crooks, B. Ostrovsky, Y. Bar-Yam, Mesostructure of polymer collapse and fractal smoothing, Phys. Rev. E 60 (1999) 4559–4563.

[6] P.-G. de Gennes, Scaling Concepts in Polymer Physics, Cornell University Press, Ithaca, 1979.

[7] M. Doi, S.F. Edwards, Theory of Polymer Dynamics, Oxford Science, Oxford, 1986.

[8] G.D. Doolen, (Ed.), Lattice Gas Methods for Partial Differential Equations, Addison-Wesley, Reading, MA, 1990.

[9] D. Farmer, T. Toffoli, S. Wolfram, (Eds.), Cellular Automata, North-Holland, Amsterdam, 1984.

[10] E. Freire, Thermodynamics of partly folded intermediates in proteins, Annu. Rev. Biophys. Biomol. Struct. 24 (1995) 141.

[11] B. Ostrovsky, Y. Bar-Yam, Irreversible polymer collapse in 2 and 3 dimensions, Europhys. Lett. 25 (1994) 409.

[12] B. Ostrovsky, Y. Bar-Yam, Motion of polymer ends in homopolymer and heteropolymer collapse, Biophys. J. 68 (1995) 1694.

[13] B. Ostrovsky, M.A. Smith, Y. Bar-Yam, Applications of parallel computing to biological problems, Annu. Rev. Biophys. Biomol. Struct. 24 (1995) 239.

[14] B. Ostrovsky, M.A. Smith, Y. Bar-Yam, Polymer interpenetration in 2D high density melts, Int. J. Polym. Complex Fluids (4) (1995).

[15] B. Ostrovsky, M.A. Smith, M. Biafore, Y. Bar-Yam, Y. Rabin, N. Margolus and T. Toffoli, Massively parallel architectures and polymer simulation, in: Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing, 1993, p. 193.

[16] M.A. Smith, Cellular automata methods in mathematical physics, Ph.D. Thesis, MIT, Cambridge, 1994.

[17] M.A. Smith, Y. Bar-Yam, Y. Rabin, C.H. Bennett, N. Margolus, T. Toffoli, Cellular automata simulation of polymers, in: E.B. Sirota, D. Weitz, T. Witten, J. Israelachvili (Eds.), Complex Fluids, MRS Symposium Proceedings, vol. 248, 1992, p. 483.

[18] M.A. Smith, Y. Bar-Yam, Y. Rabin, B. Ostrovsky, C.H. Bennett, N. Margolus, T. Toffoli, Parallel processing simulation of polymers, J. Comput. Polym. Sci. 2 (1992) 165.

[19] T. Toffoli, N. Margolus, Cellular Automata Machines: A New Environment for Modeling, MIT Press, Cambridge, 1987.

[20] S. Wolfram, (Ed.), Theory and Applications of Cellular Automata, World Scientific, Singapore, 1986.