CELLULAR AUTOMATON SIMULATION OF POLYMERS

M. A. SMITH, a Y. BAR-YAM, b Y. RABIN, c N. MARGOLUS, a T. TOFFOLI, a AND C. H. BENNETT d
a MIT Laboratory for Computer Science, Cambridge, MA 02139
b ECS, 44 Cummington St., Boston University, Boston MA 02215
c Department of Physics, Bar-Ilan University, Ramat-Gan 52900, Israel
d IBM T. J. Watson Res. Ctr., Yorktown Heights, NY 11973

ABSTRACT

In order to improve our ability to simulate the complex behavior of polymers, we introduce dynamical models in the class of Cellular Automata (CA). Space partitioning methods enable us to overcome fundamental obstacles to large scale simulation of connected chains with excluded volume by parallel processing computers. A highly efficient, two-space algorithm is devised and tested on both Cellular Automata Machines (CAMs) and serial computers. Preliminary results on the static and dynamic properties of polymers in two dimensions are reported.

INTRODUCTION

Polymers vary from "simple", usually synthetic, linear chains of identical monomers to highly complex chains consisting of sequences of amino-acids that form the building blocks of living organisms. Realistic dynamical simulations of polymers for study of such problems as protein folding are expected to be one of the major scientific undertakings in upcoming years. An ability to take advantage of massively parallel computer architectures with up to 10^5 processors would dramatically improve the effectiveness of simulations. In order to illustrate the difficulties in simulating polymer dynamics and their resolution, it proves quite fruitful and enlightening to consider abstract polymer models.

One of the basic paradigms of polymer science is that many of the dynamical and structural properties of sufficiently long macromolecules can be understood within the framework of an abstract polymer model. In this model the polymer consists of elements which are connected in sequence and avoid intruding on each other's space. There are only two essential parameters, the length and the excluded volume. Even for this simple model where analytical tools such as mean-field, scaling and renormalization group methods provide a basic understanding of physical properties, computer simulations are needed to check analytical results in simple systems and study complex systems of dense, grafted, branching, matrix-embedded polymers, etc. For more realistic models, computer simulations provide the only hope of obtaining detailed information. The central difficulty in simulating the behavior of polymers is the large number of individual components necessary to effect the conformation of a long macromolecule.

What methods should one use to simulate complex high molecular weight polymers? Conventional simulations are of two types: molecular dynamics, and Monte-Carlo. 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 dynamics represent 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 the limiting feature of parallel computers and resolutions of communication problems only apply to particular computer architectures. The general approach to parallel processing dynamics we have developed can be used for both molecular dynamics and Monte Carlo simulations. Since Monte-Carlo does not require the specification of artificial forces in the abstract model, we describe the approach in the language of Monte-Carlo step dynamics for this model.

In Monte-Carlo simulations of abstract polymer structure and dynamics, a long chain consisting of attached monomers is represented by the coordinates of each monomer. A simulation step consists of selecting a monomer <i> from the polymer chain and performing a move. There are many different methods for describing the local structure of the polymer and
the process of each move. However, quite generally, each move is required to preserve the constraints that arise from the two parameters of abstract polymers, their length and excluded volume. These constraints are:

1. The move does not "break" the polymer connectivity - monomer $<i>$ does not dissociate itself from its nearest neighbors along the chain (NNCs).
2. The move does not violate excluded volume - monomer $<i>$ does not overlap the volume of any other monomer $<j>$. 

In naive parallel processing, a set of processors is assigned one-to-one to perform the movement of a set of 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 has generally been considered impossible to perform systematic parallel processing on polymer dynamics since moving different monomers at the same time is likely to lead to dissociation or overlap. The former constraint only restricts the parallel motion of nearest neighbors and thus can be overcome using simple algorithms. In contrast, the "non-local" excluded volume constraint restricts the parallel motion of any two monomers, presenting a fundamental difficulty for parallel processing.

**CELLULAR AUTOMATON DYNAMICS**

We propose a general way to overcome this difficulty 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. As a general rule, monomers sufficiently distant in space from each other can be moved in parallel without any possibility of interfering in that step. Two monomers can move independently if the distance between them is greater than $2l + r$, where $l$ is the step length and $r$ is the excluded volume radius. This can be realized by partitioning space as in Fig. 1. In each shaded region, of linear dimension $(l+r)$, we select a monomer (whose center lies in that shaded region). Then we move all the selected monomers independently. Finally, we shift the shaded regions before the next selection process. This brief description specifies entirely the modification of an existing non-parallel algorithm into a parallel one. However, note that there is an additional computational task: the tabulation of monomers in each region, which must be updated with each move. This update can be performed efficiently and in parallel.

The idea of space oriented dynamics is manifest in the very general category of dynamical models known as Cellular Automata (CA)\(^6\) of which the best known example is Conway's "Game of Life"\(^7\). An automaton consists of a rule for specifying the state of a system after a time interval (defined to be one unit of time) in terms of its previous state. In a simplified form, dividing space into cells, and considering each cell to have only two possible values ON and OFF, the state of a cell is determined by the condition of its neighborhood at the previous time. Different rules for determining the state of the cell describe different dynamics. The Cellular Automaton is a sufficiently general construction that it is believed to capture the essence of dynamical systems for formal investigations\(^8\),\(^6\). Specific automata have been used to model systems of interest in Biology, Chemistry and Physics. CA are ideal for simulation on parallel computers. Moreover, there exist Cellular Automata Machines\(^9\) which are computers specifically designed to simulate a cellular automaton for study and analysis.

Two recent efforts have been directed at using automata in the description of polymers or strings. Chopard\(^10\) showed for the first time how extended objects such as strings could be simulated. Koelman\(^11\) adapted a lattice gas CA dynamics by incorporating a non-automaton dynamical step which attached monomers together to form a polymer.

Standard CA 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 we must establish a rule where turning off one site is tied to turning on another site. A modification of CA developed to describe such systems is Margolus dynamics\(^9\), where the rule updates a whole neighborhood rather than a single cell. Then, a conservation law which holds in the neighborhood update (the number of ON sites) also holds globally.

We have developed a CA Margolus dynamics for the simulation of polymers. For example, in two dimensions: 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. 2) each monomer has two such neighbors except for the ends which have only one. We select a sub-lattice of cells (illustrated by circles in Fig. 2) separated by 3 cells in each direction. If a monomer is located in one of these cells, it can be moved up, down, right, or left as permitted by the constraints of maintaining connectivity and excluded volume. Permitted movements are
Figure 1: Illustration of an abstract polymer composed of monomers which are connected to neighbors and do not overlap "excluded" volumes. The motion of monomers is restricted so that they do not detach from neighbors or overlap with any other monomer. Moving any two monomers at the same time can lead to inadvertent overlapping. However, if only one monomer is selected from each shaded region the monomers selected can be moved at the same time without chance of accidental overlap. The shaded regions can then be shifted so all monomers can be moved.

Figure 2: Illustration of a cellular polymer model which can be moved using Margolus Cellular Automaton dynamics. Monomers are considered attached if they are touching by either faces or corners. Moves are only originating in the squares marked by central circles in the middle of 3x3 neighborhoods separated by buffers and possible moves are given by the arrows. Moves which are not allowed due to excluded volume are marked by an "x" in the target square.

Illustrated by arrows. Movements not allowed by excluded volume are indicated by Xs in Fig. 2. A movement corresponds to an update of the 3x3 plaquette around a selected site; the information for the update is contained in a larger 5x5 region. All 3x3 plaquettes can be updated simultaneously. Finally, the selected sublattice (circles) is shifted before the next set of update steps (preferably randomly to avoid possible correlations in the motion).

In the simulation of long polymers, since we are interested in the asymptotic behavior where local properties are unimportant, and since the relaxation time increases dramatically with the number of monomers, it is important to have a model where the high molecular weight limit is realized for moderate numbers of monomers. As for real polymers, the long-length 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 and motion along the polymer contour (which involves local length changes in the long polymer). An effective approach to minimize the influence of local structure is to allow local changes in polymer length. In order to enable monomers to be more flexible in their motion we have generalized the CA dynamics just described.

In a generalized formulation of polymer dynamics, we define a region V of cells around each monomer as its "bonding neighborhood". A polymer is constructed where each monomer has its NNCs in the region V about it. The dynamics is defined simply by requiring that the motion of a monomer be allowed only if its movement to a new position (selected at random from a movement region M around it) does not change its NNCs. This preserves both connectivity (preventing loss or change of a neighbor) and excluded volume (preventing the addition of a neighbor). In the original CA dynamics (Fig. 2) V was the 3x3 region around the monomer. Using the general formulation V can be larger, for example a 5x5 region. NNCs no longer need to be adjacent in space: the bonded neighbors of a monomer are defined solely by being within V. Then monomers can separate by one space and remain NNCs, as in the
fluctuating bond method.\textsuperscript{12,13} This enables local length changes, improving the simulation of high molecular weight polymers. For parallel processing, monomers can be moved independently at a distance where two monomers can be moved towards each other without one entering the other's bonding neighborhood V.

**TWO-SPACE ALGORITHM**

The consideration of more flexible polymer dynamics leads us to a second general class of dynamics which enables a different partitioning of the polymer for simulations. The simplest way to describe this dynamics in two dimensions is to consider a polymer on two parallel planes (see Fig. 3). The monomers of the polymer alternate between the planes so that odd-numbered monomers are on one plane and even-numbered monomers are on the other. The polymer and its dynamics are described as in the preceding paragraph except that the NNCs of a monomer are located in the other plane and excluded volume with respect to non-NNC monomers is only enforced with respect to monomers in the other plane. Simply stated, using the language of the preceding paragraph, the bonding neighborhood of a monomer in one plane is located entirely in the other plane. Using this dynamics an additional flexibility is achieved because neighboring monomers can be "on top of each other" so that even a 3x3 neighborhood dynamics allows local expansion and contraction. More interestingly, it is possible in this dynamics to move all of the monomers in one plane at the same time without concern for their interference because both connectivity and excluded volume are implemented through interactions with the other plane. This is an alternative to the concept of partitioning space, through a special partitioning of the polymer. The ability to update 1/2 of the monomers simultaneously implies that this dynamics is efficient for implementation on both CA machines and other parallel processing computers such as Hypercubes and Connection Machines. It may even be possible to take full advantage of conventional vector processors.

To show that all the monomers in one space can be moved independently, we must show that their motion cannot result in either breaking the polymer or violating excluded volume. Since each monomer is moved preserving its NNCs, the polymer can not be broken. Excluded volume is different for two monomers within a plane and for two monomers in opposite planes. For two monomers in opposite planes, the excluded volume is the neighborhood volume and preventing a change in number of NNCs also prevents violation of excluded volume. For two monomers in the same plane excluded volume is just the requirement that two monomers do not move onto the same site (they can be adjacent since they are never NNCs). 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 NNCs and, since our algorithm explicitly prevents any two monomers from having the same NNCs, the above situation can not arise.

![Fig. 3: Schematic illustration of a two-plane polymer used in two-plane polymer dynamics described in the text. Monomers on the upper plane are shown as filled circles, monomers on the lower plane are shown as open circles. Monomers are attached only to monomers in the other plane. The attachment neighborhood of each monomer is a 3x3 region of cells located in the opposite plane. A lightly shaded region indicates the attachment neighborhood of the black monomer marked with a white dot. Its two neighbors are located in the attachment neighborhood. Attachments are indicated by line segments between monomers.](image-url)
The two-space algorithm may be used in three dimensions by considering the polymer to be in a double three-dimensional space. The double-space polymer dynamics can be mapped onto a single-space dynamics with an unusual implementation of excluded volume. This difference of the local interaction does not affect the asymptotic structural and dynamical behavior, which therefore is in the same universality class as other abstract polymer models. Thus the second general algorithm we have described is a highly efficient technique for simulations of abstract polymer problems. However, the polymer partitioning used is not obviously applicable to the dynamics of realistic polymer models, which may still be parallelized by the use of the previously described space partitioning.

RESULTS

We have performed test simulations of the two-space algorithm on the present state-of-the-art Cellular Automata Machine, CAM–6, and on SPARC stations. The results of both simulations are presented in Fig. 4 where we plot the radius of gyration of a polymer as a function of its contour length and compare with the exact result in two dimensions, \( R \sim L^{0.75} \). We also used the algorithm to calculate the longest relaxation time of the chain. For simplicity of analysis only the SPARC data is presented in Fig. 5, and compared with the Rouse model prediction, \( \tau \sim L^{2.5} \).

Finally, we would like to comment on the efficiency of the proposed implementation of CA algorithms on parallel and serial machines. All of the algorithms discussed above scale with the number of moves to relax a polymer, \( M_{\tau} \sim L \tau \sim \alpha L^{2v+2} \), where \( 2v+2 \) equals 3.5 in two dimensions, and 3.2 in three dimensions; \( \tau \) is the relaxation time, \( L \) is the length of the polymer, and \( \alpha \) depends on polymer local structure. In a conventional algorithm the calculation time scales as \( M_{\tau}(t_m + L t_x) \); \( t_m \) is the time to execute a move and \( t_x \) is the excluded volume test time. In space partitioning this becomes \( M_{\tau}(t_{m,x,s} + t_c V/V_p)/P \); \( P \) is the number of processors, \( t_{m,x,s} \) is the time to perform a move and excluded volume test and maintain the space partitioning, \( t_c \) is the time to check whether a partition is occupied, and \( V/V_p \) is the ratio of simulation volume to filled volume (the second term may often be eliminated by coarse graining). In the CA two-space polymer model on a CAM the time scales as \( M_{\tau} t_m/(PV_p/V) \) and on connection machine or hypercube as \( M_{\tau} t_m/P \). In a CAM, \( t_m \) is of order 1 (one time step per move), while for other parallel machines \( t_m \) is substantially larger, though empty space need not be processed. Therefore, when the new generation modular 3-dimensional CAM-8 becomes operational with 4 million cells and 25 million cell updates per second per module it will be ideal for simulation of high density polymer melts where conventional computers are approaching their limit.\(^{14} \)
Figure 5: Simulations of the relaxation time $\tau$ of a polymer as a function of polymer length $L$, testing the two-space algorithm. Simulations were performed on a SPARC station. The asymptotic fit indicated by the dashed line was obtained as $\tau \sim 0.12 L^{2.55}$ consistent with the Rouse model exponent of 2.5. The small prefactor indicates the efficiency of the two-space

Figure 4: Simulations of the radius of gyration of a polymer $R_g$, as a function of the polymer length $L$ (one less than the number of monomers), testing the two-space algorithm on a workstation (o) and on CAM-6 (*) the current generation Cellular Automata Machine (soon to be replaced by CAM-8). An asymptotic fit indicated by the dashed line is $R_g \sim 0.8 L^{0.751}$ consistent with the exact exponent 0.75. Agreement with the asymptotic values are reached for remarkably small polymers of length $L=2$. 
algorithm. Polymers of length 30-50 already approach the asymptotic behavior.

This research is supported in part by the US-Israel Binational Science Foundation.

REFERENCES: