LATTICE POLYMER AUTOMATA

Steen RASMUSSEN[a,b] and Joshua R. SMITH[b,c]

[a]TSA-DO/SA MS M997 and CNLS MS B258
Los Alamos National Laboratory
Los Alamos, NM 87545
U.S.A.
[b]Santa Fe Institute
1660 Old Pecos Trail
Santa Fe, NM 87505
U.S.A.
[c]Physics and Media Group
MIT Media Laboratory, 20 Ames Street
Cambridge MA 02139
U.S.A.

From
Berichte der Bunsen-Gesellschaft für physikalische Chemie 98, 1185-1193 (1994) No. 9

Abstract

We present a new style of molecular dynamics and self-assembly simulation, the Lattice Polymer Automaton [LPA]. In the LPA all interactions, including electromagnetic forces, are decomposed and communicated via propagating particles, "photons". The monomer-monomer bond forces, the molecular excluded volume forces, the longer range intermolecular forces, and the polymer-solvent interactions may all be modeled with propagating particles.

The LPA approach differs significantly from both of the standard approaches, Monte Carlo lattice methods and Molecular Dynamics simulations. On the one hand, the LPA provides more realism than Monte Carlo methods, because it produces a time series of configurations of a single molecule, rather than a set of causally unrelated samples from a distribution of configurations. The LPA can therefore be used directly to study dynamical properties; one can in fact watch polymers move in real time. On the other hand, the LPA is fully discrete, and therefore much simpler than traditional Molecular Dynamics models, which are continuous and operate on much shorter time scales. Due to this simplicity it is possible to simulate longer real time periods, which should enable the study of molecular self-organization on workstations; supercomputers are not needed.

1 Introduction

1.1 The Vision

Because of the complexity of the physical chemistry of biopolymer dynamics and self-assembly, and the limited simulation capabilities current computers give us, minimal models of molecular interaction are needed to simulate molecular self-organization in an explicit manner. Using current techniques, simulating the formation and dynamics of micelles, liposomes and other nanoscale structures requires heavy use of supercomputers, because most current Molecular Dynamics tools are designed to study the detailed behavior of macromolecules and molecular aggregates. The ability to model and make intensive simulation studies of polymer self-organization, which occurs on time scales inaccessible to more detailed models, should provide insight into the emergence of a variety of mesoscopic physical phenomena including proto-life forms.

In this paper, we present the conceptual framework, implementation, and first tests of a "minimalist" 2D polymer simulation tool.

Required first of all is an explicit representation of the system's functional elements – the interacting molecules. Without an explicit representation of the functional elements, autonomous construction of novel structures and properties is not possible, and the system will not be self-programmable. Earlier, more abstract approaches, studying the dynamics of self-programmable matter (rather than "real" matter) [9] [7] [8] have had a variety of problems ranging from the question of validity[1] to problems concerning the conservation of the newly constructed functional structures in the systems[2]. In particular, the problem of how newly formed functional structures may be conserved has lead us to a closer study of how the laws of Physics support the spontaneous emergence of boundaries.

The particular choice of conceptual framework for this simulation test bed is not obvious and several frameworks might work. Some of the considerations behind our choice are discussed in the two subsections below. Other examples of discrete polymer algorithms can for instance be found in [2] and [5].

1.2 Discrete Fields and Mechanical Models: Basic Considerations

Cellular Automata may be viewed as computational models of physical systems. They embody *discrete* field models: space and time are divided into a lattice of cells of finite size, each of which occupies one of a discrete set of states. The state reflects "what is going on" in that region of space.

To clarify the claim that Cellular Automata embody discrete field models, we can contrast the CA data structure with that of a mechanical model (i.e. a model provided by Newtonian, Lagrangian, Hamiltonian, Quantum, or Statistical mechanics). In a CA description, the fundamental entities are points in spacetime. If we imagine a CA model as a database, it would consist of a record for each point in space. A record (x, y) has attributes such as "particle present" or "particle absent," and also topological attributes, for example pointers to cells whose coordinates place them adjacent to (x, y)— or the topology may be given by an array the records are stored in. Whether two adjacent particles are free or bound to each other depends on the local rules.

In a pure mechanical description, the fundamental entities are particles, which have attributes such as "position". A "mechanical" database would contain one record for each particle. The header would be an artificial label, i, used to distinguish the records from one another. Attributes such as coordinate values would be recorded in the fields below.

In a CA, geometrical relationships between the entities are reflected directly in the data structure. If the molecule at cell (x, y) is 2 units away from the molecule at cell (x', y'), then cell (x, y) is two pointers (array slots) away from (x', y'). This also means that, continuing with the example above, there is no way to represent a situation in which two molecules occupy the same state—the particle register in cell (x, y) can

[1] Since it is not the constructive dynamics of interacting model-molecules we trace in these more abstract models, are we then justified in infering anything about the dynamics of real molecules?

[2] The abstract self-programmable systems we have studied earlier[7] [8] have had difficulties in constructing or inventing the notion of a functional boundary, or "membrane". This is, however, not an inherent problem with the level of abstraction, but rather a problem with the particular representations.

only be set to "molecule present" once. In the mechanical database, by contrast, there is no intrinsic constraint that prevents all the molecules from occupying the same state. The coordinates of every molecule i could be set to the same value (x_0, y_0). Futhermore, a mechanical representation makes spurious distinctions between physically identical configurations, leading to the overcounting problem known as the Gibbs' paradox. With a discrete field model, there is a one-to-one correspondance between logical states and distinct physical states.

In a CA the volume of space being simulated and the spatial resolution of the simulation — in other words, the total number of cells — determines the computational complexity of the problem. The number of particles in the system is irrelevant, unlike in mechanical models. Since the transition rule is the same at all cells, empty or otherwise, it is just as easy to simulate 10^6 particles as it is to simulate 10. In a sparsely populated CA, most of the computational resources will therefore be spent simulating empty space, resources that a mechanical model would not waste.

The computational complexity of algorithms based on the pure mechanical data structure is typically n^2, where n is the number of monomers. This is because the data structure places no a priori communication constraints on molecules. For example, the algorithm will expend just as much time calculating the perhaps trivial effect of a distant monomer m_j on m_i as it will calculating the effect of a nearby monomer m_k. Even though it would only require constant time to compute a new configuration using a version of a molecular dynamics model with one processor per particle, each processor would need to communicate with $n - 1$ others. A CA also computes a new configuration in constant time, but each processor requires only a constant amount of communication, since cells talk only to their neighbors. On the assumption that a processor can communicate simultaneously with only a finite number of others, a pure mechanical Molecular Dynamics technique clearly will not scale up with n.

1.3 Lattice Gas Automata as Models of Physics

Our model is inspired by Lattice Gas Automata, LGA, one of the first Cellular Automata to be used for accurate modeling of complex physical systems[10]. In LGA microscopic conservation laws encoded in the local rules produce physically realistic dynamics one level higher. From a microphysics that conserves particle number and momentum emerge continuous density and velocity fields that are solutions of the Navier-Stokes equations of fluid flow. Similarly, we will see how a microphysics that respects certain key constraints leads to the emergence of higher-order structures and dynamics, namely polymers and polymer dynamics, that posses properties not found on lower levels.

Like most CA, the LGA evolves synchronously. A cell's new state is a function of its current state and the state of its six neighbors. However, there are two stages in the update of the Lattice-Gas dynamics: a transport phase and a scattering phase. In the transport phase, each particle jumps to the neighboring cell indicated by its velocity; in scattering, the particles, having just arrived at new cells, change velocity according to symmetric binary and triple collision rules that conserve particle number and momentum. In information processing terms, interprocessor communication occurs in the transport step, and computation happens in the scattering step.

The standard LGA universe consists of a two-dimensional, triangular lattice of cells, each of which is connected to its nearest neighbors to form a hexagonal neighborhood. (Six-fold symmetry, in addition to the conservation laws, is necessary for the emergence of the Navier Stokes equations.[11]) Periodic boundary conditions may be set by defining the cell connectivity properly. Other types of boundary conditions, for example, forbidden sites, are defined by identifying certain cells as boundary cells, and updating them differently.

An important strength that Lattice Gasses inherit from the Cellular Automata framework is that boundary conditions and/or internal lattice sites/areas may be made arbitrarily complex without affecting the computational complexity of the algorithm. So with properly chosen boundary conditions, Lattice Gases are capable of efficiently simulating phenomena such as fluid flow in porous media.

Because a Lattice Gas model is discrete and uses computational resources extremely efficiently, it can be orders of magnitude faster than other techniques (in particular, those which rely on floating point numbers) for simulating fluid flow. The discreteness also means that there are no numerical difficulties such as roundoff error. It should be noted, however, that not all fluid flows, (as for instance high Reynolds number flows) at present can be simulated in a satisfactory way using LGA techniques.

2 Lattice Polymer Automata

2.1 The basic LPA concept

The Lattice Polymer Automaton is based on the assumption that all molecular interactions can be modeled by mediating particles. We model both matter and fields as "information particles" that propagate locally along the edges of a lattice and interact with one another at nodes, as in the Lattice Gas model. Unlike a standard LGA, the LPA uses several different types of particles, so the structure of a node is more complicated than the simple six bit register required for a minimal LGA.

The transmission of the force particles between the monomers enables an update of each individual monomer using only local information; after the transport step, each lattice site can be updated independently. The force-communicating particles propagate locally, that is, between neighboring lattice sites.

A variety of molecular interactions may be formulated by chosing the mediating particles properly. An LPA polymer must obey a connectivity constraint and an excluded volume constraint: bonded monomers must remain bonded, and monomers may not penetrate the excluded volume around a polymer. An LPA update rule must cause the monomers to move without violating these constraints.

The models to be discussed are formulated on a 2D hexagonal lattice and use three types of particles. One of these particles, the "attracton," enforces the connectivity constraint, by communicating the information that allows the monomers to maintain the coordination needed to avoid breaking. Another particle, the "repellon," implements the excluded volume constraint, by communicating a repelling force that prevents more than one monomer from occupying the same site. Since the solvent is not modeled explicitly in this version of the LPA, we introduce a third particle, the "kickon." The kickon is a phantom particle that induces a random velocity in a monomer and thus in an implicit way models the solvent. By choosing a hexagonal lattice initially the solvent may at some later point be introduced explicitly using a 2D LGA of fluid flow.

2.2 The LPA Data Structure

An LPA cell has several registers which serve to represent the state of a monomer. The velocity register can take 8 values: 0 means that no monomer is present, 1-6 indicates a velocity from East through South East, and 7 indicates a stationary monomer. This representation scheme is used because only one monomer is allowed per lattice site—thus the full six bit scheme normally used to represent particle velocities in a LGA is not needed.

Each cell has two registers for representing bond directions, which point at two of its nearest neighbors. Thus if a cell's bond vectors are correctly oriented, a monomer "knows" the location of its neighbors—it need not do any communication in order to locate them. (Communication must of course occur if the monomers are to behave collectively as a polymer. The communication step occurs later when we adjust the bonds to track the motion of neighboring monomers.) A bond register may take the value 0, which means a bond is not present, or a direction from 1-6 if a bond is present.

The repellon register is used to implement the polymer's excluded volume, that is, to prevent collisions between monomers that are not bonded together. It represents force particles called repellons, using the usual 6 bit LGA data structure. This register can be used in several different ways—there are many sensible repellon update rules, two of which we will describe below.

The final two registers hold attractons, the message particles that moving monomers use to keep their neighbors—in particular their neighbors' bonds—informed about their new whereabouts.

Each cell has pointers to its six neighbors: `e`, `ne`, `nw`, `w`, `sw`, and `se`. The lattice has toroidal boundary conditions, which are implemented by setting the pointers appropriately. Once the pointers are in place, one never need worry about boundary conditions, because the lattice data structure actually has the topology of a torus. In a fine-grained parallel implementation with one cell per processor, the pointers would become connections between processors.

Each cell also has a pointer called `other` which points to its corresponding cell in a second layer of cells that has the same topology as the first. (As in most CA, a second layer of cells is needed as "scratch space" to represent the previous state while the next state is being computed.) One may follow an `other` pointer to the second layer, and then follow the `other` pointer in the second layer back to the original cell in the first layer.

A cell contains one additional pointer that is particular to the serial implementation, and that would not appear on a parallel machine. This is the `next` pointer, which snakes through all the cells in the lattice. When an operation has to be done synchronously on all cells, we trace through this list. Since all update steps happen synchronously, the order in which the cells are updated does not matter, as long as all cells are reached. On a parallel machine, all the cells would be updated simultaneously. Since the cells can be updated in any order, it is practical for sparse simulations on serial machines only to update occupied cells. This can be accomplished by following the bond vectors in order to determine which cell to update next.

2.3 The LPA Update Cycle

2.3.1 Scheduling

The polymers are updated in two distinct phases. Therefore two types of monomers—call them "green" and "red"—are defined. Green monomers can only bond with red monomers, and vice versa. All the green monomers are updated synchronously in the first phase, and the red monomers are updated in the other. The advantage of this interleaved update scheme, which was introduced by Chopard[2], is that the neighbors of a moving monomer are guaranteed to be stationary. This simplifies other aspects of the update algorithm significantly.

2.3.2 The Solvent (kickons)

First the velocities of all the monomers are randomized. We think of the random impulses as being provided by the phantom particles called kickons. Since these kickons appear from nowhere, our model does not conserve momentum. It should be possible to replace the kickons by impulses from solvent particles, which would be simulated explicitly as a LGA.

Next some of the velocities are reset to zero by consulting a look-up table indexed by the velocity, bond, and color registers. In the phase in which only green particles may move, all the red velocities will be zero after this step. The bond directions of the green particles, coupled with the knowledge that the red will be stationary, determines the positions of the neighbors. Thus this table is also used to implement the no-break constraint, by halting those green monomers whose kickon-suggested moves would break the chain.

2.3.3 Excluded Volume (repellons)

At this point, the repellons come into play to implement the excluded volume. We have implemented several different repellon schemes. One of these yields an excluded volume of radius 2, and one yields an excluded volume of radius 1.

In the radius 2 scheme, repellons are created at the location of each monomer, in all directions except along bond vectors. The repellons propagate one step. Then they "split," in a step that might be described as a lattice version of Huygens' principle. If the lattice directions are indexed by $i = 0, \ldots, 5$, then each repellon in direction i becomes three repellons, in directions $i-1, i, i+1$ (modulo 6). Now additional repellons are created at the monomers, as before, and all the repellons propagate one step before splitting again. The polymer is now surrounded by a "field" of repellons, as shown in figure 1.

To prevent collisions the monomers must interact with the generated repellons. A table indexed by monomer velocity and repellon register is used to halt those monomers headed for a violation of the excluded volume constraint. If a monomer has a velocity in direction i, and a repellon in the same cell has come from that direction, that is, if there is a repellon in the direction $i+3$ (modulo 6) then the velocity is reset to zero. So another particle can move to within distance 2 of the polymer, but then cannot come any closer. Thus there is a "statistical" force in directions radially out from the polymer, since intruding monomers are jostled randomly by kickons, but cannot travel inward. This means that (un-bonded) monomers tend to move away from one other if the distance

Figure 1: Six stills (a – f) from the 15 step update cycle of the LPA with radius 2 repellons. (a) Initially the kickons randomize the monomer velocities. After zeroing the suggested monomer velocities that would break bonds, the repellons (the excluded volume particles) are created, transported and then scattered (using a discrete version of Huygens principle) at radius 1 from the polymer. (c) New repellons are again created at each monomer, transported out to radius 1 and scattered (Huygens). At the same time the repellons located at radius 1 are transported out to radius 2 from the polymer. Now the monomer velocities suggested by the kickons are checked for excluded volume violations (repellons) and zeroed if in violation. (d) The repellons are then anhilated, the legal monomer moves are executed and then the bond directions are adjusted to track the neighbors. (e) The moved monomers finally transmit an attracton along their new bond directions which will ensure that the stationary, neighboring monomers can adjust their bond directions accordingly (f) which completes an update cycle.

between them is less than 3 lattice sites; if they have an initial separation of 2 or 1, they never move closer.

In the radius 1 scheme, a monomer sends a single particle (a "scouton") in the direction of its velocity. If another monomer— or scouton—is present at that lattice site, it reflects back to the monomer to prevent the move. This rule creates an excluded volume of radius 1.

2.3.4 The Bonds (attractons)

At this point it is guaranteed that none of the remaining proposed moves would violate the two constraints, so the monomers can now be prepared to move according to their velocities. Before monomer m_i, which we will suppose is green, may be moved, its bond directions must be adjusted so that they will again be correct after it moves. This is accomplished with a look-up table indexed by the monomer velocity and bond directions. Then the monomer can be moved, which actually means copying its velocity, bond, and color registers into the cell (in the other layer) pointed to by the velocity.

After m_i moves, the bond vectors of its stationary neighbors are no longer oriented correctly. So m_i (along with all the other green monomers) sends the message particles called attractons along its bond vectors to its red neighbors, which adjust their bonds accordingly.

A refresh step which copies the modified information in the other layer back to the original one, plus erasure of the repellons, completes the first phase of the update cycle. In the second phase, the green particles will be stationary and the red will move. Smith *et al.* [5] have remarked that this sort of interleaved update scheme violates detailed balance, but that this can be respected by chosing stochastically which set of monomers to update. However, this two step update scheduling scheme in a significant way simplifies the update algorithm as a whole.

3 Static and Dynamical properties of LPA

In this section we discuss some of the dynamical properties that emerge at the polymer level from the LPA rules. Different LPA formulations yield different polymer properties and particular polymer properties can be obtained by proper design of the propagating force particles and their local interpretations.

3.1 Polymer Elasticity

Here we present a simple equilibrium thermodynamics model of emergent polymer elasticity. As we shall see, a polymer behaves as an entropic spring: it experiences a restoring forces proportional to extension, because of the decrease in entropy required to stretch the spring. This sort of elasticity is an emergent property that arises from the statistics of large numbers of linked monomers, even in the complete absence of intrinsic elasticity, that is, pairwise elasticity between monomers.

We can derive an expression for $f(r)$, the force exerted by the polymer on its ends when it is anchored with end-to-end separation r. Suppose that the end-to-end vector $\mathbf{r}$ is the result of a random walk. Then the probability of end-to-end length r will be

$$p(r) = Ae^{-\beta^2 r^2}$$

where

$$\beta = \frac{d}{2Na^2}$$

d being the dimensionality of the space the polymer inhabits and a the bond length [3]. The constant A can be found by normalization. Now we can write the entropy

$$S(r) = -k_B \ln p = k_B \beta^2 r^2 - c$$

and take the derivative with respect to r

$$\frac{\partial S}{\partial r} = 2k_B \beta^2 r$$

Then the Helmholtz free energy can be expressed in terms of the entropy:

$$F = U - TS$$

where U is the internal energy of the system. We can now write down an expression for the force:

$$f = \frac{\partial F}{\partial r} = \frac{\partial (U - TS)}{\partial r} = \frac{\partial U}{\partial r} - T\frac{\partial S}{\partial r}$$

so

$$f = \frac{\partial U}{\partial r} - 2k_B T \beta^2 r$$

There are two force terms. The first may be identified as intrinsic elasticity—it is a manifestation of the energy that comes directly from the state of the polymer, for example from attractive or repulsive interactions between the particles. The second represents emergent elasticity. It is the term that is due to the configurational entropy of the polymer. The physical origin of this force is that the monomers are being shaken vigorously, so that the polymer explores its configuration space. The statistics of this space therefore dictate what happens to the polymer, determining, for example, what length it will (probably) assume after a certain amount of shaking, and so on. Because there are fewer "long" configurations available, the shaking results in a restoring force.

3.2 Dynamical and Scaling properties of LPA

Some of the dynamical properties of the LPA are seen in figure 2, which shows the timeseries of R_0, the end-to-end distance, $< R_0 >$, the cumulative average of R_0, R_g, the radius of gyration, $< R_g >$, the cumulative average of R_g, and the distance the center of mass has diffused from its initial location. The radius of gyration is given by $R_g = \sqrt{I/M}$, where the total polymer mass is $M = \sum_i m_i$, and m_i is the mass of the i'th monomer. The moment of inertia of the polymer is $I = \sum_i m_i r_i^2$ and r_i is the distance from the center of mass of the polymer to the i'th monomer.

In figure 2 the dynamics of the polymer relaxation from its initial stretched (linear) state is shown. Note how the polymer, after in this particular run initially being caught in a local minimum, quickly relaxes and then engages in oscillations with a frequency of approximately 10^3 updates. Thus, an elastic force does indeed emerge, as the simple theoretical analysis suggests. It is clearly seen that the fluctuations in R_0 are much larger than the fluctuations in R_g which also follows directly from their definitions. Also note that even after approximately $10 \cdot N^{5/2}$ updates ($N = 13$) the cumulative averages of R_g—and in particular of R_0—are still varying substantially.

It is known [3] that R_0 and R_g for ideal polymers should scale as $N^{3/4}$ in the theoretical limit. Our LPA polymers are stiffer than an ideal polymer; only angles $\geq \frac{2}{3}\pi$ are allowed (recall section 2.3); so we should expect a scaling exponent which is closer to (but still smaller than) unity.

In figure 3 the scaling behavior of $< R_0 >$, $< R_g >$ (cumulative averages) for the radius 2 excluded volume rule and the radius 1 rule are shown. The simulated polymers range from length 5 to 56. By performing least squares on a log-log "plot," we estimate that for the radius 2 rule, $R_0 \propto N^{.934}$ and $R_g \propto N^{.914}$; for the radius 1 rule, $R_0 \propto N^{1.048}$ and $R_g \propto N^{0.962}$. Since the statistics are much better for R_g than for R_0, R_g should give a better estimate of the limit value of the scaling exponent. Note that both scaling exponents are close to unity; the exponents for the radius 1 rule are apparently somewhat larger than those for the radius 2 rule. The estimated scaling exponent for the radius 1 rule is a few percent larger than unity, which in principle should not be possible, but this deviation

Figure 2: The dynamics of a polymer consisting of 13 monomers; R_0 is the end-to-end distance, $< R_0 >$ is the cumulative average of R_0, R_g is the radius of gyration, $< R_g >$ is the cumulative average of R_g, and *Diff* is the distance the center of mass has diffused from its initial location.

Figure 3: The scaling of $< R_0 >$ and $< R_g >$, the cumulative average of the end-to-end distance and the cumulative average of the radius of gyration respectively, for radius 1 and radius 2 repellon LPA of length 5 to 56.

is not statistically significant given the way R_0 is computed (recall the discussion of figure 2). It is likely that in the limit of large number of monomers N, the scaling behavior of the two rules will converge, both to an exponent ≤ 1.

The simulation time used to obtain the R_0 and R_g scaling values is given by $10 \cdot N^{5/2}$, which should give the initially linear polymer time to reach a stady state.

4 Informational Physical Chemistry

4.1 Channel Capacity of a Polymer

One interesting feature of discrete computational models such as the LPA is that all information flow may be traced explicitly. At any time step one may write down exactly how many bits are being communicated between pairs of monomers, and also how many bits are being created or erased at each update step.

If we assume at each step in the update cycle that all the possible messages a cell might receive from its neighbors—that is, all its possible successor states—are equally likely, then the bit accounting of an LPA update cycle for a central monomer or an end-monomer can easily be calculated.

Let us consider the bit accounting for three typical substeps of the LPA update cycle. In one of these information is created, in one information is erased, and in one information is only transmitted, but neither created nor destroyed[3].

To proceed, we need to recall the properties of the data structure of a monomer, as well as the steps in the update cycle. Please see section 2.3. It should be noted that the information estimates based on these assumptions give an *upper bound* for the amount of information that a (central) monomer can receive/transmit/destroy in each step of the update cycle.

In the initial kickon creation step, for every monomer, either no kickon is created, or a single kickon in one of the 6 lattice directions appears. In our present model, these 7 possibilities are equally likely. Thus $\log_2 7 = 2.807$ *bits* are created at each monomer site in this step; this is the entropy of the probability distribution describing the kickon register. Since there should be no correlations between the kickons, each monomer has become less correlated with its neighbors after this step, and more correlated with the phantom "heat bath," in reality the random number generator behind the kickons. In other systems, correlations with the heat bath are often referred to as noise.

In the "clobber" step, some of this noise (information stored in the monomer about the phantom heat bath) is erased; the kickon register will become more correlated with the bond direction registers. Before the clobber step, the kickon and two bond registers are equally likely to be in any of their $7 \cdot 6 \cdot 5 = 210$ allowed states. After the clobber step, the kickon register is in the "no kickon present" state unless the angle between the bonds was $\frac{2\pi}{3}$ and the kickon was centered between the bonds; there are only 6 such states. Therefore the entropy of the kickon probability distribution has fallen to $-6 \cdot \frac{1}{210} \log_2 \frac{1}{210} - \frac{204}{210} \log_2 \frac{204}{210} = 0.261$ *bits* from 2.807 *bits*; 2.546 *bits* of noise from the solvent have been erased. The remaining 0.261 *bits* of noise would cause correlations between monomers to decay, were it not for the communication step analyzed below, which

[3] Note that a complete LPA implementation which modeled the solvent explicitly would be physically and logically reversible, so none of its steps would create or destroy information.

increases correlation between the monomers. Without a communication step to strengthen correlations, the monomers would quickly cease to be recognizable as a polymer.

In the attracton transport step, a monomer that has just moved sends messages to its stationary neighbors informing them of its new whereabouts. So a stationary monomer receives one of 3 possible messages from its left (mobile) neighbor, and one of 3 possible messages from its right (mobile) neighbor. Assuming once again that all these messages are equally likely (the maximum information case), the monomers receive $\log_2 9 = 3.170$ *bits* in this step. Unlike the two steps discussed above, in which information was created and destroyed, this step involves the communication between the monomers that ultimately is what allows them to remain correlated in the higher order structure known as a polymer.

It would be interesting to see whether these a priori calculations of the upper bound on the communication rate between monomers do indeed limit the observed mutual information between pairs of monomers. The nature of the interactions between the monomers determines a maximum rate of communication between them, a channel capacity; the correlations between monomers, as measured by mutual information, should respect this channel capacity limit.

The particle communication and the local use and deletion of the communicated information in the LPA makes it very clear which bits, or bit combinations, cause which dynamical actions. This provokes one to reflect on "real" physical molecular interactions from an informational point of view. One might wonder what the channel capacity of a real polymer is.

4.2 Operational Semantics of Polymer Interactions

In the LPA, the local use — or "interpretation" — of the different kinds of communicated information defines the *operational semantics* of the information. The solvent particles (the kickons) are for instance interpreted differently by the monomers depending on the context the monomer currently is in. If the monomer is free, the kickon will induce a movement in its direction, but if the monomer is polymerized the effect will depend on whether such a movement would violate bond restrictions (the attractons) and/or a part of the excluded volume (the repellons). Thus, the "meaning" of a kickon depends on the context in which it occurs. In particular, this observation can be stated in the following way: the interpretation of the information depends on which hierarchical level [1] the communicating objects belong to. In this system we have two levels: (i) the level of the free monomers and the solvent, and (ii) the level of the polymers. In general, new hierarchical levels support new means of communication, both within new levels and between old and new. Note that different levels in general need not be part of a strict hierarchy.

This interpretation of the information depending on the context (e.g. hierarchical level) of the interacting objects has profound consequences for how higher order (≥ 3) emergent structures can self-assemble in formal self-programmable or constructive dynamical systems.

To bring it back to Physics: a semantic analysis of the interactions (communication) between objects in a physical system will uncover which properties of the interaction (communication) cause which actions, in particular which properties at a certain level cause the emergence of higher order structures. In fact, we already perform a kind of semantic analysis of the physical interaction when we "explain" physical phenomena.

5 Discussion

We are developing a modeling and simulation tool for an investigating of the formation, the dynamics, and the interactions of polymers.

The two first main steps in this development are:

> To discretize all force interactions through propagating informational particles, and
>
> to design a local interpretation of the communicated information that allows the movement of an extended (non-local) object, the polymer, by local rules.

We have in detail presented how this can be done.

5.1 LPA Applications

In the present form, the LPA could be used to simulate the free flow of polymers in a solvent by making the "kickon flow" anisotropic, for example by making kickon creation in one of the 6 directions more likely than in the other directions.

It should also be possible to study the flow of polymers in porous media with the current formulation. In addition to adjusting the distribution of kickons, stationary objects on the lattice need to be introduced. This can for instance be done by fixing periodic polymers (where the polymer ends are joined) on the lattice and making them inert to the kickons.

The dynamics of polymers in an external field could also be simulated in the current formulation of the LPA. By allowing another phantom particle—like the kickon, but with a strictly directional action— to act on certain predefined monomers (simulated charged parts of the polymers), the simulation of polymers in an external field is possible. By setting the boundary conditions of a porous medium as described above, it should also be possible to simulate the dynamics of polymers in a gel, which also has many potential applications (e.g. DNA, RNA and protein electropheresis).

Thus, it should be possible to use the LPA directly in the study the dynamics of polymer systems in a variety of industrial and laboratory processes.

5.2 Alternative Formulations of the LPA

In the design and experimental phase of the first two steps mentioned above, we tried a variety of alternative formulations. These include a parallel update scheduling algorithm without two "update-colors" on the polymers. A completely parallel updating scheme complicates the update cycle significantly. The main reason is that the light-cone in a completely parallel update (the number of lattice sites that has to be taken into consideration in order to update a single site) grows tremendously. In principle, each monomer must have information about each other monomer in the polymer to complete an update. For the two-step parallel updating scheme the light cone for each monomer is limited to the nearest neighbors. We have also tested algorithms with expandable bonds. These imply an updating cycle with more steps than the one currently in use.

5.3 Long Range Forces, Polymerization and Hydrophilic/Hydrophobic Interactions

Since long range forces, polymerization and hydrophilic/hydrophobic interactions are important additional properties of a more complete LPA system and since their formalization is non-trivial we shall touch upon some of the major issues associated with the representation of these properties, although they have not yet been integrated in the LPA. We have, however, constructed and tested long range force fields and are currently experimenting with different, simple algorithms to obtain LPA polymerization and hydrophilic/hydrophobic properties.

Long range forces can of course also be carried by particles (photons). There are several ways to generate discrete electromagnetic fields by means of propagating particles [13]. Since the field due to an impulse charge is a function only of distance, ($\sim 1/r^2$ in 3D and $\sim 1/r$ in 2D) it is possible to generate an electromagnetic field in an update time which is proportional to the distance from the source. Note that this force particle dynamics is occurring at a different time scale than the movements of the molecules (monomers). In real time the field propagation occurs with the speed of light as do the repellon and attracton dynamics (recall section 2.3).

The monomer dynamics induced by these long range fields can be derived from the resulting local field which is calculated by (truncated) vector addition of six different local monomer fields, one field for each direction, which have been generated in the field propagation steps. This resulting force is then treated in the same way as the kickons (the implicit solvent particles, recall section 2.3). It should be noted that this is not the only way to generate long range fields through a particle propagation.

Molecular self-assembly is easily obtained using classical cellular automata and movable finite automata [4]. For the LPA polymerization can for instance be obtained by allowing oppositely charged end-monomers (monomers with one free bond) to ignore each other's repellons and thus form bonds once the attracting forces have brought the polymer ends within distance one. Polymerization can also occur when the polymer dynamics is driven by diffusion alone, but then the process is much slower, since it is not actively directed by a potential (a field).

Obtaining hydrophobic- and hydrophilic- interactions requires at least two different kinds of monomers, let us call them A and B. Such an interaction can be obtained either with or without an explicit representation of the solvent (water) particles.

One way of obtaining hydrophobic-like properties for the B-monomers or polymers without an explicit representation of the solvent is to let the B-monomer communicate information about its presence to its neighborhood via a repellon-like mechanism. The non-bonded B monomers can use this information to attract each other by preventing the kickons from moving them apart, and by repelling the A-monomers.

5.4 LPA and Computational Performance

Although Molecular Dynamics simulations traditionally used a mechanical description, the last few years of development in this area seems to point towards a hybrid description adopting some of the properties of local computational models. Modern, large scale, Molecular Dynamics simulations on massively parallel computers have discretized space into cells and only consider molecular interactions from molecules within the same or between neighboring cells. Instead of assigning one molecule to each processor, each processor is responsible for a discrete part of space (a cell) with one or more molecules, whose number (may) change over time as a function of the dynamics.

One of the currently fastest Molecular Dynamics codes [6] uses approximately $0.26 \cdot 10^{-6}$ sec/(particle-update) on the CM-5 Connection machine having approximately $20 \cdot 10^{6}$ particles. Each particle update corresponds to some femto (10^{-15}) seconds real time and a typical simulation runs for some thousand updates. The (current) upper limit for the maximal time steps size in Molecular Dynamics simulations where all atoms are explicitly represented seems to around a pico (10^{-12}) second, so the real time simulated in a typical Molecular Dynamics simulation does not exceed the nano (10^{-9}) second range.

Due to the complete discreteness and absence of floating point operations the LPA should be faster in the same environment. And since the real time per update in the LPA range from 10^{-9} - 10^{-1} seconds, depending on which molecules we are simulating, it becomes possible to trace some of the molecular self-organizing processes which occur on timescales as long as hours, as for instance the formation and self-replication of micelles and liposomes [12].

5.5 Self-Organization

The main thrust of our efforts are now concentrated on the introduction of LPA long range forces, polymerization as well as hydrophobicity and hydropilicity.

With the integration of one or more of these properties we can initiate systematic studies—on a work station—of what we originally started out to study: the dynamics of some of the most fascinating—and least understood—processes in subcellular and (pre-) biological systems, the processes that assemble and self-organize the interactions between a range of key biomacromolecules.

We also believe that the LPA can become a simple vehicle for the more conceptual study of successive emergence of higher level structures in formal dynamical systems.

References

[1] N.A. Baas. Emergence, hierarchies and hyperstructures. *Artificial Life III, proceedings, ed. C.G. Langton, Addison-Wesley/Santa Fe Institute Studies in the Sciences of Complexity*, 1994.

[2] B. Chopard. A cellular automata model of large-scale moving objects. *Journal of Physics A*, 23:1671–87, 1990.

[3] P.G. deGennes. *Scaling Concepts in Polymer Physics.* Cornell University Press, Ithaca, New York, 1979.

[4] H. Hotani *et al.*. Microtubule dynamics, liposomes and artificial cells. *Nanobiology*, 1(1):61–74, 1992.

[5] M.A. Smith *et al.*. Parallel-processing simulation of polymers. *Computational Polymer Science*, 2:165–171, 1992.

[6] P.Lomdahl *et al.*. 50 gflops molecular dynamics on the connection machine 5. *Proceedings of Supercomputing 93 (IEEE Computer Society Press)*, 1993.

[7] S. Rasmussen *et al.*. The coreworld: Emergence and evolution of cooperative structures in a computational chemistry. *Physica* **D**, 42(1), 1990.

[8] S. Rasmussen *et al.*. Dynamics of programmable matter. *Artificial Life II, ed. C.G. Langton* et al. *Addison-Wesley/Santa Fe Institute Studies in the Sciences of Complexity*, pages 211–254, 1992.

[9] W. Fontana. Beyond digital naturalism. *Artificial Life 1*, 1:211–227, 1994.

[10] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56:1505–1508, 1986.

[11] B. Hasslacher. Discrete fluids. *Los Alamos Science*, Special Issue(15):175–217, 1987.

[12] P.L. Luisi. The chemical implementation of autopoiesis. *Nato-Maratea meeting 1993, Self-reproduction of supramolecular structures; Proceedings to appear.*

[13] S. Rasmussen and J.R. Smith. Long range forces in lattice gasses. *preprint*, 1994.