Cell-based models

Tamulonis, C.V.T.

Citation for published version (APA):
Let’s begin our overview of cell-based models by presenting a simple motivating example: a model of \textit{in vivo} cancer growth inhibited by the immune system and external pressure, proposed by Qi et al. (5). There are five types of cell in the model: normal ($N$), cancerous ($C$), dead ($D$), effector (immune system response cells, $E_0$) and a cancer cell bound by an effector ($E$). The basic rules of the model are specified using reaction laws:

\[
\begin{align*}
C & \xrightarrow{k_1} 2C \\
C + E_0 & \xrightarrow{k_2} E \xrightarrow{k_3} E_0 + D \\
D & \xrightarrow{k_4} .
\end{align*}
\]  

(2.1)

where the $k_i$ are the reaction rates. The first reaction depicts cancer cell proliferation, the second is the response of the immune system and the third is the dissolution of dead cells.
Chapter 2 – Lattice-based Models

One way to approach this system would be to design a system of PDEs, typically reaction-diffusion equations, that implemented the above reactions and then solve the system, yielding a cell density field across the domain. This approach would assume that the cells are infinitesimal in size compared to the tumor itself and behave as molecules in a chemical reaction. This approach has the advantage of not requiring that we specify the precise mechanics of cell collisions and cell movement, allowing us to concentrate on the phenomenon of interest, i.e. cancer growth. Analytical equations can also be studied to determine the general properties of the system, rather than relying on extrapolation from a series of simulations.

The alternative is to not take the system to the continuum limit, in which individual cells and their behaviors are lumped into macroscopic equations, but to consider a system composed of individual cells. Each cell is pre-programmed to behave according to rules specified by the modeler, however once the simulation is running the cells are left to their own devices. The macroscopic features of the system are therefore calculated implicitly by the virtual cells as the simulation runs, as opposed to being codified into system-wide differential equations and then integrated.

These types of models are called agent-based or individual-based and are used in a wide range of subjects including physics, social science, economics and biology. When the “agents” are cells, then the term cell-based model is used. Agent- or cell-based models are most often used for complex systems composed of individuals whose behavior is well described by simple rules, but from which it is difficult to derive macroscopic rules for the behavior of the system as a whole. In other words, systems in which “the sum is greater than its parts”.

In cell-based models more details of the system must be included. We must decide how to represent the cells in the model and what geometry the cells may assume. And we must also specify what happens when cells collide when moving, or what happens if a cell is under mechanical strain or when a cell adheres to another cell. This chapter and the following are concerned with these basic issues. We introduce multiple techniques for modeling cell geometry and basic cell mechanics. We start with the simplest models and gradually add more detail with each model presented. As we progress, we present elementary physical concepts related to cellular mechanics and introduce biological systems well suited for cell-based modeling. Although there is no single technique suitable for all systems, we have tried to present a
Our review is divided into two chapters based on the representation of the spatial domain. In this chapter, we review techniques that are based on a lattice representation of space, in which objects are only allowed to occupy fixed positions. In Chapter 3 we discuss lattice-free methods, in which objects can move continuously through space.

2.1 A Lattice Model of Cancer Growth

The first issue we will address when designing a cell-based model is how to keep the cells separate. Since we are assuming that the cells are of finite size, then each cell has its own exclusive volume that cannot intersect with another cell’s exclusive volume. This must be explicitly taken care of by the modeler, and there are multiple techniques of doing so depending on how the cells are represented.

Perhaps the simplest solution is to use a regular lattice for representing space and representing each cell as a single point in the lattice. For example, returning to our cancer example, Qi et al. (5) assume that the cancer grows on a two dimensional plane. This is common for cancer models, since tumors are often roughly radially symmetric and can be represented by a single slice though the center. The plane is divided into equally sized squares that form a regular grid. The squares are occupied by cells and are called sites. The basic rule is that each site can hold at most one cell, and all the other rules are designed to respect this. Therefore, we don’t have to write explicit rules for cell collisions and we can concentrate on just the behaviors we are really interested in, in this case cancer growth limited by external pressure and the immune system.

Initially each lattice site is occupied by a normal cell, except for five cells at the center of the domain. The model progresses in time by visiting each lattice site and transforming the site using transition rules. A timestep is complete once the transition rules have been applied to all the sites. The transition rules for this particular example are based on the reactions in Eqs. 2.1, and are functions of the cell type occupying the site and the cell type of its nearest neighbors, i.e. the cells immediately above, below and to the sides. If the visited cell is:

- cancerous (type C) and at least one of its neighbors is a healthy cell,
then the cell will proliferate with probability $k_1'$. Proliferation consists of picking a random neighboring normal cell and replacing it with a cancer cell. If the cell does not proliferate, then it may be bound by an effector cell with probability $k_2$, forming a cell-effector complex $E$.

- bound by an effector (type $E$) then it cannot proliferate and it will die with probability $k_3$.
- dead (type D), it is replaced by a normal cell with probability $k_4$.
- normal (type N), then nothing happens.

In addition there are two rules controlling growth. The cancer growth is limited by the amount of nutrients available and so the rate of growth is inversely proportional to the number of cancer cells in the system $N_C$: $k_1' = k_1 (1 - N_c/\phi)$ where $\phi$ is the total number of cancer cells the system can support. Growth is also affected by the internal pressure of the tumor as measured by the cell density $d = N/R^2$ where $N$ is the total number of cancer, dead and bound cells and $R$ is the average radius of the cancer. The cancer can only grow outwards if there is sufficient internal pressure, which is proportional to the cell density. If the density is greater than some critical value $d_C$, then when a cancer cell divides the daughter cell can replace any normal cell neighboring the mother cell. Otherwise, the daughter cell can only be placed in the interior of the cancer, i.e. in a direction facing the center.

By performing a few hundred simulation steps, the initial five cell cancer grows until the maximum size $\phi$ is reached (Fig. 2.1a). The macroscopic feature of interest is the size of the tumor. Plotting the size of the simulated tumor shows good agreement with the expected growth of real tumors according to the Gompertz curve (Fig. 2.1b).

### 2.2 General Characteristics of Lattice Models

Qi et al.’s (5) model has typical elements of lattice models. Each lattice site is associated with some variable of state, in this case the cell type occupying the site. The transition rules are functions of a local neighborhood, in this case the four adjacent sites above, below and to the sides, called the first nearest neighbors or the von Neumann neighborhood (Fig. 2.2). Another common neighborhood is the eight-site Moore neighborhood or the second nearest
2.2 – General Characteristics of Lattice Models

Figure 2.1 – Simulating cancer growth using a lattice-based model. (a) A simulation of a growing tumor in 2D. (b) Comparing the growth of the simulated tumor with that of mouse carcinoma KHT. The jagged line represents the growth of the model tumor. The smooth line represents the typical Gompertz growth of the mouse carcinoma. Adapted from Qi et al. (5).

neighbors, which also include the adjacent sites in the diagonal direction. At each step of the simulation, every site’s state is updated by applying a transition rule based on the site’s current state and the state of its neighbors.

The sites at the top/bottom/left/right edges of the lattice do not have bottom/top/right/left neighbors and special rules may be conceived for these, or alternatively it can be assumed that the lattice wraps around such that the left neighbors of the sites on the left edge of the lattice correspond to the sites on the right edge, and so forth for the top, bottom and right edges. In this case the model is said to have periodic boundaries.

The sites in the model can either be all updated simultaneously – a synchronous update – or one by one such that the lattice sites that have not yet been updated can already see the new states of those sites that have already been updated – an asynchronous update. In Qi et al.’s model asynchronous updates are performed. Finally, the model has both local rules and global rules. Local rules are functions of a site’s local neighborhood, whereas global rules are functions of the entire system. For example, the rules for proliferation
Figure 2.2 – Common lattices and neighborhood templates. (a) The regular square lattice is the most common. (b) The regular hexagonal lattice is often used to reduce anisotropic effects (see e.g. §2.6.1). (c) The four site von Neumann neighborhood. The black site can only “read” its four closest neighbors. (d) The eight site Moore neighborhood. Adapted from Deutsch and Dormann (6).
described above are global rules because they depend on the total number of cells and the dimensions of the tumor. The rules cannot be applied without looking at the whole lattice. On the other hand, the rules for cell death and cell-effector dissolution are local rules because they do not depend on any site outside of the local neighborhood.

2.3 Cellular Automata

A lattice model that uses only local rules is called a cellular automaton (CA). In computer science, an automaton is an idealized machine that has an internal state that changes according to the input fed to it. For each input value given, the automaton may transition to a different state according to some internal rules. In the case of cellular automata, each site houses an automaton whose input is the state of the other automata in its neighborhood. Note that the automata can only interact directly with their close neighbors, but a single automaton can indirectly affect the entire system by initiating a chain of local interactions. For example, assume that there are only two possible states \{0, 1\} for each site and the transition rule is such that if both neighbors have the same state, then the state is set to 0, otherwise the state is set to 1. This is the so-called “Rule 90” (7) that demonstrates how a very simple system can generate a complex structure. By starting the simulation with a single site set to 1 and using Rule 90 to evolve the system, a curious recursive pattern of triangles called the Sierpinski Triangle appears, a fractal geometry that “looks” the same regardless of how far you were to zoom out. Rule 90 is one of the “simplest” complex systems: a simple rule and a single differentiated site are sufficient to set off a riot of interactions that form a complicated pattern that is not an obvious result of the underlying rule.

Given the conceptual similarity between CA and real cells, there are naturally many CA models of biological systems (introductions to cellular automata applied to biological systems can be found in e.g. 8, 9, 10, 11, 12). Consider, for example, the following automaton proposed by Young (13) for the formation of animal coat patterns. The automaton uses a two-dimensional lattice with periodic boundaries and synchronous updates. There are two types of cells: (a) undifferentiated cells (UC) and (b) differentiated cells (DC) that are pigmented. Each DC produces two morphogens – chemical agents that have the ability to transform cells into different types. One morphogen acts as activator, which turns UC into DC, and the other as an inhibitor,
which turns DC back into UC and prevents UC from differentiating. The morphogen with the highest concentration wherever the cell is located has the dominant effect. The combined effect of the pair is measured by subtracting the inhibitor concentration from the activator and the result is called the morphogen field \( w(d) \) where \( d \) is the distance from the DC cell.

As the morphogens diffuse away from the DCs they decay, producing a decreasing gradient of concentration from the source cell outwards. A key element of these activator-inhibitor models is that the activator is produced in larger quantities but decays quickly, giving it a shorter range, whereas the opposite is true for the inhibitor. Therefore the activator dominates at short range, whereas the inhibitor dominates at longer ranges. Young used the simplest morphogen field that fulfills this requirement:

\[
  w(d) = \begin{cases} 
    w_1 & 0 \leq d < R_1 \\ 
    w_2 & R_1 \leq d \leq R_2 \\ 
    0 & \text{otherwise} 
  \end{cases}
\]

where \( w_1, w_2 \) are the respective activator and inhibitor strengths, and \( R_1, R_2 \) are their respective ranges. Overlapping fields from neighboring DCs are additive, so that the total morphogen field \( W(r) \) experienced by the cell at position \( r \) is given by:

\[
  W(r) = \sum_{i \in \mathcal{N}} w(|r_i - r|)
\]

where \( \mathcal{N} \) is the set of DCs within a circular neighborhood of \( r \). At each time step, the cells change state depending on the value of \( W \) at their location:
2.3 – Cellular Automata

- If $W(r) = 0$, then the cell does not change state.
- If $W(r) < 0$ then the cell becomes a UC.
- If $W(r) > 0$ then the cell becomes a DC.

Initializing the lattice with randomly distributed DC cells and running for just a few steps, stable patterns reminiscent of animal skins emerge (Fig. 2.4).

It is sometimes possible to derive a system of PDEs which are equivalent to the CA model, when we assume that the CA sites become infinitesimally small. This is advantageous since the dual CA and PDE models offer a microscopic and macroscopic perspective of the same system, respectively, and the link between the two scales is firmly established. In this case, Young’s (13) CA model is equivalent to a simplified reaction-diffusion system of the form:

$$\frac{\partial M}{\partial t} = \nabla \cdot D \cdot \nabla M - KM + Q$$

(2.2)

where $M = M(x, t)$ is the morphogen concentration and the terms on the right represent diffusion, chemical reactions and production respectively.
Reaction-diffusion models such as this are an important class of models in theoretical biology.

2.4 The Importance of Locality

Although many models are dubbed “cellular automata”, in reality many of them use global rules and asynchronous updates, so do not have the strict locality and independence of “pure” CA. This is more than a semantic quibble, since these properties lend significant computational advantages.

CA models with local rules are ideal for parallelization on computing clusters composed of hundreds of networked computing nodes. The CA domain can be split up into pieces and distributed to the nodes. Since only local interactions are required between elements, communication between the nodes is limited to sharing the state of the nodes located at the edges of each piece of the domain. No node needs to look at arbitrary positions in the domain and no particular piece of data is shared by all the nodes. This reduces the computational complexity and communication overhead of distributed implementations of the model program. It is also an important consideration for parallel implementations with shared memory, as random memory access is highly penalized in massively parallel computers. Perhaps more importantly, models with asynchronous updates are generally difficult to parallelize. This is because these models are inherently serial, since the application of a transition rules one site at a time causes the application of one rule dependent on application of the previous. In general, parallelization requires that a problem be decomposable into independent sub-problems, and so the parallelization of asynchronous CA is often a challenge. These considerations are especially important today compared to the recent past, as increases in computing power come in the form of multiple computer cores working in parallel, instead of faster single cores.

Another positive point is that models which rely only on local rules generally have linear computational complexity also written $O(n)$, where $n$ is the number of elements in the system. This means that the computing time required for a simulation is directly proportional to the number of elements in the system. In other words, double the number of cells and you double the the running time as well. This is because for each element we only have to look at a small subset of elements in the local neighborhood. If, however, the model were to use a global rule that required it, for each element, to look at
every single other element in the system, then the complexity of the model would become quadratic $O(n^2)$ and doubling the number of cells in the system would result in a four fold increase in simulation running time. Not all global rules necessarily imply increased computational complexity, however. Well implemented global rules can conserve the linear complexity of a model, although global rules do in general complicate parallel implementations of models.

When designing a computational model, it is important to keep the computational complexity as low as possible. Fortunately, cell-based models will typically have linear complexity, because the cellular systems they simulate are inherently local. Cells function as independent units that typically only interact with their immediate neighbors, or communicate over long distances through diffusable signals, itself a local process. Hence biological systems at the cellular level are well suited for modeling since, in principle, we can model ever larger systems without a disproportionate increase in computing time.

2.5 A Model of Embryogenesis

Some models walk a thin line between the CA paradigm and more general lattice-based models. Longo et al.’s (14) model for the thinning of the *Xenopus laevis* blastocoel roof is a good example of how the simplicity of pure CA can often be too limiting to be practical, especially in the case of mechanical interactions such as contact and collisions.

*Xenopus laevis* is a widely studied model organism in developmental biology. Like most animals, *Xenopus* embryos undergo a process called gastrulation during which the embryonic cells rearrange themselves to form three distinct cell layers (only two in simpler animals). Gastrulation in *Xenopus* is rather complicated involving many types of cell migration such as involution and convergent extension, which we will elaborate on later. Longo et al. (14) developed a model for a specific process of *Xenopus* gastrulation called blastocoel roof thinning. The process occurs at a phase when the original egg cell has already divided multiple times forming the blastula, which in *Xenopus* has a spherical shape. The bottom hemisphere of the blastula is solid and filled with cells, whereas the upper hemisphere has a large cavity called a blastocoel, which is filled with fluid. The portion of the blastula lining the top of the blastocoel is called the blastocoel roof (BCR) and it is composed of
several layers of cells. As the blastula develops, the BCR gradually thins and extends as the cell layers merge forming just two layers, a phenomenon aptly called blastocoel roof thinning (Fig. 2.5).

In the model the BCR is represented as a two-dimensional regular square grid 84 cells wide and 3-5 cells high. Initially the domain is differentiated vertically by three distinct layers. The top layer is one cell deep and is composed of superficial cells, the middle layer is composed of deep inner cells and is 1 - 3 cells deep and finally the bottom layer is composed of bottom deep cells and is one cell deep. The dynamics can be divided into two separate but related features:
Intercalation. Inner deep cells are allowed to intercalate by inserting themselves into an upper or lower row of inner deep cells or bottom deep cells. Once an inner cell migrates into the deep bottom layer, however, it is trapped and not allowed to move upwards.

Extension. A row that receives an intercalating cell is required to shift either to the left or to the right to make room for the cell. Cells cannot intercalate into the superficial layer, however the superficial layer extends over the bottom layers whenever these extend due to intercalation.

The rules are simple, but in practice they can be difficult to implement if one wants to adhere strictly to CA local interactions. For example during movement 1 in Figure 2.6, an inner deep cell moves to the row below it. In order for the move to occur, however, the lower row must shift to the left by one cell to make room (movement 2). This is simple to do from a programming standpoint: just move the green cell that’s in the way to the beginning of the row a then move the red cell to its place. This simple solution however is not a local interaction as it affects sites that are relatively far from each other. It would be possible to implement such a move in purely CA terms: the red cell would have to signal to the green cell that it intends to take its place, the green cell would then have to signal to the cell on its left to move over, and this signal would have to propagate through the whole row until the last cell in the row got the message and moved one cell over. Then one at a time the green cells would move over until a space opened up for the red cell to move into. Needless to say, such a program would neither be efficient nor elegant and such explicit programming of the model behavior would defeat one of the main purposes of a CA model, which is to show that simple rules can generate complex behavior. Regardless, the model does capture various quantitative aspects of the BCR thinning process and stands as a good example of how a simple model can be informative and predictive, even if we cannot pigeonhole its modeling approach.

The BCR model illustrates the difficulty of implementing a lattice-based model that includes cell migration using ad-hoc rules. Systems in which local events can trigger a large-scale instantaneous reaction are not naturally expressed using strict CA. However, by relaxing the strict CA paradigm, two sub-classes of CA models exist that are well suited for this sort of problem:
Chapter 2 – Lattice-based Models

Measuring the dilution of the bottom cell layer

An embryo explant seeded on an FN substrate containing labeled bottom deep cells and unlabeled intercalating cells was monitored over time (Marsden and DeSimone, 2001). The number of cells that intercalate into the bottom deep layer in the first hour of intercalation was counted, and this number was converted to a dilution rate based on the total number of cells in the bottom deep layer.

Running the simulation and collecting data

The simulation was run in the NetLogo environment (Wilkensky, U. (1999). NetLogo. http://www.ccl.northwestern.edu/netlogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL). Simple algorithms were generated to collect data such as total thinning time, temporal FN values, and cell locations. The data were exported to Microsoft Excel for analysis.

Statistical analysis

Differences in mean data between samples were compared using Student’s t tests, and differences were considered significant at \( P < 0.05 \).

Results

To build the CA simulation, we defined the cells, cell layers, and tissue organization in the simulation and incorporated rules for cell behaviors based on independent experimental work published in the literature and our own unpublished experimental observations. To test the predictive capability of the CA model, we compared the spatial patterns of cellular rearrangement and tissue layer morphology, as well as the temporal dynamics of BCR thinning predicted by the model to those measured in independent experimental studies. To further assess the predictive value of the CA model and establish the robustness of the computational approach, we simulated implantations of labeled donor cells into recipient BCRs and tracked the lateral dispersions of these cells throughout the virtual BCR. We compared the predicted results to those generated by performing the same experimental manipulation in vivo. Finally, we used the CA simulation to test the hypothesis that FN deposition and fibrillogenesis at BCR cell surfaces are influenced by differential cell adhesion and cell residency times in the BCR.

Building the simulation

The NetLogo modeling tool was used to develop the computer simulation of the BCR thinning process. Net...
Lattice Gas Cellular Automata (LGCA) models and the Cellular Potts Models (CPMs).

### 2.6 Lattice Gas Cellular Automata

Lattice Gas Cellular Automata (LGCA)\(^1\) were originally conceived to model fluid flow as an alternative to the often intractable numerical integration of Navier-Stokes equations. LGCA models are a discrete form of molecular dynamics models. Simulations are performed on the scale of individual “particles” that move in discrete steps along a regular lattice. The particles move from site to site through *velocity channels*, which connect nearest neighbors to each other. On a two-dimensional square lattice\(^2\), each lattice site has four velocity channels, one for each of its nearest neighbors

\[ \mathcal{N} = \{(-1,0), (0,1), (1,0), (0,-1)\} \]

corresponding to the cardinal directions \{N,E,S,W\}\(^3\). The velocity of a particle corresponds to the channel it is in, such that a particle residing in the south channel has velocity \((1,0)\), \((0,1)\) in the east channel and so forth. Channels can hold at most one particle and each particle must reside in some channel, therefore sites can hold at most four particles at any given time. The state of a site \(s\) is simply a list of 4 binary values \(s = \{\eta_1, ..., \eta_4\}\) that indicate whether a channel is occupied or not.

The dynamics of LGCA models are split into two parts: interaction and propagation. During propagation the particles simultaneously move into neighboring sites according to the velocity channel they occupy. After the propagation step, a particle residing in the north channel at site \((i,j)\) will move to the north channel at site \((i-1,j)\), a particle in the south channel will move to \((i+1,j)\), and so forth. Following propagation, the interaction step consists of reconfiguring the state of each site so that particles may change channels due to, for example, the result of a collision or some other process. Note that with propagation alone, each particle would simply move along a straight path forever. In specifying interactions between particles, we can customize their behavior to model particular phenomena.

---

\(^1\)See Deutsch and Dormann (10) and Deutsch (15) for extended introductions of LGCA.

\(^2\)In practice LGCA models only work well for hexagonal lattices, but for the sake of clarity we use a square lattice.

\(^3\)Note we are using matrix indices \((i,j)\) as opposed to Cartesian \((x,y)\) coordinates.
2.6.1 A Model of Adhering Cells

Although largely superseded by Lattice Boltzmann models for modeling fluid flow, LGCA models have found various applications in biology (10). The LGCA particles can be interpreted as motile cells and the interaction step can be implemented to implement specific cell behaviors. Consider, for example, a simple model of aggregating adhesive cells (15). Each cell is modeled as a single particle that is subject to the general rules outlined above. The interaction step consists of applying the following procedure simultaneously to each site \( r \), with state \( s(r) \) and cell count \( \rho(r) \):

1. A local gradient \( G \) is calculated to determine the direction of highest cell concentration:
   \[
   G = \sum_{v \in N} v \rho(r + v)
   \]  
   (2.3)

2. For every possible state \( s_i \) with \( \rho \) cells a weight is calculated:
   \[
   W_i = \exp(\alpha G \cdot J(s_i))
   \]  
   (2.4)

   where \( \alpha \) is the aggregation strength and \( J(s_i) \) is the flux of the state \( s_i \), which measures the net direction of cell movement, and is defined by
   \[
   J(s_i) = \sum_{v \in N} v \eta(s_i, v)
   \]  
   (2.5)

   where \( \eta(s_i, v) = 1 \) if channel \( v \) is occupied in state \( s_i \) and 0 otherwise.

3. A new state is chosen randomly with probability proportional to the weights \( W_i \).

Note that the probability distribution is designed to increase the probability of choosing states whose net flux points towards the highest concentration of cells in the neighborhood, thereby promoting cell aggregation.

Although this example uses a square lattice, the above can also be extended to a hexagonal lattice with six velocity channels. Figure 2.7 shows some results of applying this model, using both types of lattice. The advantage of the hexagonal lattice is that is largely avoids anisotropiuc effects by allowing movement in the diagonal direction in one-step, as opposed to the two steps required using the four-site von Neumann neighborhood. The
2.6 – Lattice Gas Cellular Automata

Figure 2.7 – Example simulations of adhesive cells using a LGCA model. The cells, initially randomly distributed in the domain, form small islands that coalesce and grow over time. In the top row a square lattice is used resulting in pronounced anisotropic effects. In the bottom row a hexagonal lattice is used, which greatly reduces the anisotropy. Adapted from Deutsch (15).

eight-site Moore neighborhood would also allow for movement in the diagonal direction, however a diagonal step in this case would correspond to $\sqrt{2}$ units, translating into anisotropic velocity. In practice, hexagonal lattices are used for LGCA models to reduce anisotropy.

2.6.2 A Model of Gliding Myxobacteria

Myxobacteria are Gram-negative bacteria that live in topsoil. They are flexible and rod-shaped, around 1 µm in diameter and 7 µm long\(^4\). Myxobacteria live vegetatively, consuming and multiplying, until their nutrient supply runs low. If there is sufficient cell density in the environment (detected via a quorum sensor in the form of a diffusible A-signal), the cells begin to aggregate and form a fruiting body, which produces heat resistant spores.

\(^4\)For reviews see Kaiser, Dworkin (16, 17).
Myxobacteria propel themselves using a combination of a “slime jet” mechanism located at each end of the cell (A motility) and type IV pili, which attach to neighboring cells and act as grappling hooks. They glide along their long axis when in contact with a substrate and they generally do not turn, but periodically reverse their direction. When observed in vitro in submerged agar culture, the first phase of aggregation is characterized by a dark band of bacteria at the edge of the swarm. Soon a pattern of parallel waves that oscillate backwards and forwards emerges (Fig. 2.8). The waves consist of dense bands of parallel cells that are aligned with the direction of the wave propagation. When the waves collide the cells reverse, producing back-and-forth oscillation. The first aggregates appear where two waves collide on the outer band, forming a “traffic jam” that halts the cells. As the cell density increases in the jams, the cells begin to form streams that swirl around the initial aggregate, drawing in more cells and increasing the aggregate size. Aggregates that are close by merge to form ever larger structures. Finally, once cell density within the aggregate surpasses a given threshold the bacteria begin to differentiate into spores.

Wave propagation, streaming and spore differentiation all appear to depend on C-signaling. The C-signal is localized on the outer membrane of the cells and is activated with when a cell’s anterior pole contacts another’s posterior pole. When the C-signal is triggered, the cell will tend to follow the cell ahead of it by disabling its random reversals and moving ahead behind the other cell. This allows the cells to form continuous waves and streams as one by one cells fall behind each other to “form ranks”. Therefore, Myxobacteria exhibit swarming behavior dictated by direct contact between neighboring cells.

Mark Alber’s lab has developed various models of Myxobacteria covering different aspects of fruiting body formation using the LGCA framework (18, 19, 20, 21). The basic two-dimensional model of gliding myxobacteria uses a hexagonal lattice. As with the other models presented, each cell is represented as a single particle. However, each particle has its own capsule shaped $3 \times 21$ site neighborhood meant to represent the myxobacterium shape (Fig. 2.9(b)). The model can also be extended to three-dimensions by assuming a “pea-pod” shaped neighborhood for each cell (Fig. 2.9(b)). The neighborhood is centered on the particle and oriented according to the velocity channel the particle occupies, i.e. the direction the particle is moving in. Neighborhoods are allowed to overlap, as real myxobacteria cells can crawl
Figure 2.8 – The myxobacteria development cycle proposed by Kaiser (16). Under normal conditions myxobacteria feed cooperatively by swarming, which allows them to glide more effectively. Under starvation conditions, parallel waves of high cell density form. Occasionally, when waves collide (especially at the high density swarm edges) a “traffic jam” occurs, leading to the formation of a stationary aggregate. As waves wash over the aggregate it increases in size to form a fruiting body. Once a threshold size and density is reached, bacteria in the center differentiate into spores. During this process, cells coordinate their movement through C-signaling. From Kaiser (16).
Chapter 2 – Lattice-based Models

over one another, however, following the basic LGCA rule, two cells moving in the same direction cannot occupy the same site.

The interaction step implements a scheme which imitates the alignment effect of C-signaling, triggered by anterior-posterior cell-cell contact. The posterior and anterior of a virtual cell consist of 7-site zones located at either end of the rod-shaped neighborhood (Fig. 2.9(a)). The signal strength felt by a cell is given by the number of overlapping anterior sites of the cell and posterior sites of neighboring cells. During the interaction step, a cell's direction is either maintained or is shifted one step clockwise or counterclockwise. The weight and corresponding probability of each choice is given by:

\[ W = \exp(\beta C(\theta')) \]  

(2.6)

where \( \beta \) is a parameter and \( C(\theta') \) is the C-signal the cell experiences at angles \( \theta' = \theta + \Delta \theta, \Delta \theta \in \{+\pi/6, 0, -\pi/6\} \).

Using the three-dimensional version of their LGCA model extended with C-signaling, Sozinova et al. (18) were able to simulate multiple stages of fruiting body formation including traffic jams, streams, swirling aggregates, mounds and sporulation within the body. The model demonstrated that C-signaling and cell movement alone are sufficient to explain all the phenomena.

2.7 The Mechanics of Cell Aggregates

LGCA models were originally designed for compressible gasses. Lattice sites hold discrete particles that are meant to represent individual molecules. Gliding bacteria can, roughly speaking, also be viewed as loose, hard particles and so LGCA for bacterial systems is an adequate, if somewhat coarse, modeling approach. On the other hand, eukaryotic multicellular systems are characterized by their strong cohesion and here the gas analogy breaks down. Because they are cohesive, multicellular systems tend to behave more like liquids or solids.

Steinberg (23) first postulated that cell aggregates behave like liquids by drawing an analogy between cells and molecules, since both are discrete units that form cohesive aggregates and are mobile inside the aggregate. Despite acting on different time and space scales, Steinberg believed that these two
Figure 2.9 – Multi-site representation of Myxobacteria cells in a LGCA model. (a) In 2D a cell is modeled as a rod shaped neighborhood, where the sites at the tips of the rod are used for detecting overlap with neighboring rods. Note that the neighborhoods may overlap, as only the central point is actually represented in the system (from 22). (b) In 3D, a myxobacteria cell is modeled as a “pea-pod” shaped neighborhood (from 18). (c) A 3D simulation of aggregation. Top panel is top view, bottom is side view (from 19).
essential features, which explain many features of the behavior of liquids, could also explain similar features of cell aggregates.

A drop of some liquid \( A \) suspended in another fluid of type \( B \) assumes a spherical shape when not subject to any external forces. This is because each molecule \( A \) in isolation has free cohesive energy, which is reduced when other \( A \) molecules bind to it. Inside the drop, the \( A \) molecules have a minimum free energy because they are surrounded by other \( A \) molecules. However, the \( A \) molecules at the surface are exposed to both \( A \) and \( B \) molecules, a configuration which has a higher energy associated with it. Systems of particles tend to organize such that the global energy of the system is reduced to a minimum, which corresponds to more stable configurations, and so the \( A \) molecules tend to organize such that there are as few molecules at the surface as possible. The free energy of the molecules at the liquid-medium interface generates a constant surface tension that causes the surface of the liquid to shrink to a minimum, thereby minimizing the free energy of the droplet and reducing it to a sphere.

Like a liquid drop, individual eukaryotic cells tend to round-up when suspended. This is most likely due to the active contractility of the cellular cortex driven by myosin molecular motors, which produces an effective surface tension, rather than passive free energy minimization. In addition, when a cell is subject to external strain or stress, viscous and elastic responses are observed. A cell therefore seems to have characteristics of both viscous liquids and viscoelastic solids.

Cohesive eukaryotic cell aggregates also tend to round-up. Steinberg (23) attributed this to the same physical principle of minimization of free-energy in liquids, which holds for mobile and cohesive units of any kind. In the case of individual cells, the free surface energy is due to unbound adhesion molecules embedded in the cell membrane, whereas motility is due to local cell membrane fluctuations that allow the cells to move gradually through the aggregate (24). Cells therefore are subject to both internal surface tension, which causes them to round-up in isolation, as well as interfacial tension between cells, which causes the aggregate as a whole to round-up.

The strength of cell-cell adhesion is measured as the *work of cohesion*\(^5\) and is the amount of free energy expended on joining two adhesive surfaces. In addition to explaining the rounding up of cell aggregates composed of one

\(^5\) Although the term cell-cell adhesion is used in biology, in physics the term adhesion is used for particles of different types, whereas cohesion is used for particles of the same type
cell type, Steinberg predicted that cell aggregates composed of two different cell types A and B will self-organize depending on on the relative adhesion strength between like cells ($W_A, W_B$) and different cells ($W_{AB}$), in a process called cell-sorting. He identified three possibilities, assuming type A cells adhere more strongly than type B cells:

1. If $W_{AB} \geq \frac{W_A + W_B}{2}$ then the lowest energy configuration of the system corresponds to a checkerboard pattern where contact between cells of the same type is minimized.

2. If $W_A > W_{AB} > W_B$ then the two cell types will segregate, forming a two-tier aggregate with the strongly adhering type A cells in the center either totally or partially engulfed by the less strongly adhering type B cells.

3. If $W_A > W_B > W_{AB}$ then the two cell types will segregate into two separate aggregates.

Steinberg (23)’s theory was dubbed the Differential Adhesion Hypothesis (DAH). Note that the above predictions are also valid for systems of two immiscible fluids (Fig. 2.10).

Since its formulation, the DAH and the multiple predictions stemming from it have been tested both experimentally and computationally (25, 24, 26). The most convincing evidence that cell aggregates behave as viscous immiscible liquids comes from experiments of cell sorting and engulfment (25, 24), which show that cell aggregates with less free surface energy do indeed engulf aggregates with higher surface energy (Fig. 2.10).

This model does not hold up when aggregates are compressed however (27, 28, 29). When compressed between two plates for a short period of time, around a few minutes, cell aggregates generate an elastic resistance force proportional to the aggregate’s deformation. When the pressure is released the aggregate quickly returns to its original shape. This response is consistent with that of a solid elastic body, not a purely viscous liquid. Viscous liquids respond to deformation with a resistance force proportional to the rate of deformation, and so when deformation stops no restorative force is exerted. During fast deformations the cells within the aggregate were observed to deform and so cell deformation is likely the source of the aggregate’s elasticity (27).
On the other hand, when compressed for a longer period of time (on the scale of hours) the resistance force gradually dissipates as the aggregate slowly conforms to the applied pressure in the manner of a highly viscous material. This is due either to the cells slipping past each other and rearranging or the cells conforming to their deformed shape. In both cases the inner stress of the aggregate is reduced. Therefore, cell aggregates have properties of both viscous, immiscible liquids and also of viscoelastic solids (Fig. 2.11).

In the following section we present a modeling framework that has proven highly effective in modeling the viscous liquid behavior of cell aggregates, including sorting and engulfment, and whose mechanics have a stronger physical basis – a departure from the phenomenological models of the previous sections.
2.7 – The Mechanics of Cell Aggregates

Figure 2.11 – Two models of cell aggregates. (left) In the viscoelastic solid model, cells deform during compression but do not slip past each other. (right) In the viscoelastic liquid model, cells initially deform but then rearrange to reduce their stress. (From 27)
2.8 The Cellular Potts Model

Cell-sorting and engulfment in cell aggregates were first observed over a century ago, and a computational model of cell sorting by (30, 31) is an early example of cell-based modeling. The model is lattice-based, each cell being represented as a single site. The transition rules were motivated by the DAH, but were not based on any quantitative physical law and so the models proved to be only partially representative of the real processes (32). Two decades later, an alternative model was presented by Graner and Glazier, Glazier and Graner (26, 33) based on the large-Q Potts model used for studying cellular patterns found in soap foams and metals. The authors adapted the foam model to allow for biological cells of different types and to allow the volume of the cells to be controlled. The model was named the Cellular Potts Model (CPM), reflecting its origins.

In the CPM each cell is represented as a set of lattice sites and is assigned a number $\sigma$ (called the spin) that uniquely identifies it. In addition, each biological cell has a type $\tau \geq 0$. Therefore each lattice site $(i, j)$ is part of one cell $\sigma(i, j)$, which in turn has a type $\tau(\sigma(i, j))$ (Fig. 2.12). There are usually only a couple of cell types, plus one special type $M$ that represents the medium in which the cells are immersed.

2.8.1 The Modified Metropolis Algorithm

The CPM is based on a modified Metropolis algorithm for its dynamics. The Metropolis algorithm is part of the larger family of Monte Carlo stochastic algorithms that use random number generation and statistics to approximate deterministic problems. The Metropolis algorithm is used to generate a sequence of random samples following a certain statistical distribution. In the case of the CPM, random configurations of the model lattice are generated according to the Boltzmann distribution – the probability of the system being in a given energetic state. In the Boltzmann distribution, states with increasing energy are exponentially less probable of appearing. Therefore, by successively applying the Metropolis algorithm according to the Boltzmann distribution, on average the system will tend towards states with less and less energy.

The total energy of the CPM lattice is given by a function called the Hamiltonian ($\mathcal{H}$). Once this function is defined a simulation step consists
of iteratively applying the following procedure:

1. Randomly choose a lattice site \((i,j)\).

2. Randomly choose a neighbor \((i',j')\) within the next-nearest neighborhood of \((i,j)\)\(^6\).

3. Calculate the change in energy to the system \(\Delta \mathcal{H}\) caused by setting \(\sigma(i,j) \leftarrow \sigma(i',j')\).

4. Apply the following transition rule:
   
   - if \(\Delta \mathcal{H} \leq 0\) then set \(\sigma(i,j) \leftarrow \sigma(i',j')\)
   - if \(\Delta \mathcal{H} > 0\) then set \(\sigma(i,j) \leftarrow \sigma(i',j')\) with probability
     \[ P = \exp(-\Delta \mathcal{H}/kT) \]

   where \(k\) is the Boltzmann constant and \(T > 0\) is a simulation parameter called the temperature.

\(^6\)Usually any point in the system could be used to for the swap. Since only local changes are explored, this method is called the modified Metropolis algorithm.
For low temperatures, the algorithm tends to minimize the total energy of the system, as changes to the system that lower the overall energy are always allowed, whereas changes that increase the total energy have a low chance of occurring. These random inputs of energy are permitted as they allow the system to explore alternative paths that may lead to lower energetic states, thereby avoiding getting stuck in local minima. In the case of real biological cells, these small spurts of energy can take the form of thermal undulations on the membrane and active protrusions produced from the cytoskeleton.

### 2.8.2 The CPM Hamiltonian

The Hamiltonian in every variation of the Cellular Potts model has at least two components: bond energy and volume energy $H = H_{\text{vol}} + H_{\text{bond}}$. The volume energy component assumes that the cells are elastic volumes that resist being deformed. In the case of the 2D models, the cells have elastic areas instead. The area energy takes the form:

$$H_{\text{vol}} = \lambda \sum_{\sigma \neq M} (a(\sigma) - A_T(\sigma))^2$$

where $\lambda$ is the stiffness of the cell area, $a(\sigma)$ is the equilibrium area of the cell and $A_T(\sigma)$ is the current area of the cell. Note that if the cell is at its equilibrium size it contributes no energy to the Hamiltonian, but if the cell is compressed or stretched the cell’s energy increases quadratically.

Bond energy results from contact between two cells or a cell and the medium, and is proportional to the size of the interface between the two. The bond energy of the system is given by:

$$H_{\text{bond}} = \sum_{\text{neighbors}} J\{\tau(\sigma(i,j)), \tau(\sigma(i',j'))\} \left\{1 - \delta_{\sigma(i,j),\sigma(i',j')}\right\}$$

where $J(\tau_1, \tau_2)$ is the bond energy resulting from two neighboring sites belonging to cells of types $\tau_1, \tau_2$. The second factor eliminates contributions from neighboring sites within the same cell.

### 2.8.3 Modeling Cell-sorting Using the CPM

As an initial demonstration of the model, assume we initialize a square domain with a square array of cells of random sizes (Fig. 2.13(a)) all of the same...
type \( l \) (for “light”). The bond energies are set such that cell-medium bond energy is higher than the cell-cell bond energy i.e. \( 0 < J_{l,l} < J_{l,m} \). As we run the simulation, the Metropolis algorithm minimizes the total energy of the system by favoring cell-cell bonds over cell-medium bonds, which translates into the square aggregate assuming a round shape and therefore reducing the length of the cell-medium interface. Each cell-cell bond also contributes energy to the system, however, so each cell tends to have a minimum interface with its neighboring cells (Fig. 2.13(b)). The tendency for the cell boundary to shrink is counterbalanced by the area-constraint on each cell that resists compression.

Now a second cell type \( d \) (for “dark”) is added and half of the cells in the system are randomly changed from “light” to “dark”. The bond energies are set such that \( 0 < J_{d,d} < J_{l,l} < J_{d,l} < J_{l,m} \). Note that bond energy is inversely proportional to adhesive strength, as lower energy bonds are more stable and break less often. Continuing the simulation, the light and dark cells gradually segregate from each other because the bond energies are such that contact between cells of the same type is preferred. The lighter cells eventually surround the dark cells because dark-medium contact is less favorable than light-medium contact, and the dark cells form the aggregate at the center because dark cell-dark cell adhesion is the strongest (Fig. 2.13(c)).

2.8.4 Extensions of the CPM Hamiltonian

This initial application of the CPM by Graner and Glazier demonstrated the viability of the DAH through simulation and is a prime example of an elegant and powerful lattice-based model. It also laid the groundwork for a host of new models, each with its own extensions of the basic formulation. The framework has been so successful that its original authors propose renaming it the Graner-Glazier-Hogeweg (GGH) model to recognize its evolution into a new class of models, rather than a spin off the previous large-Q Potts model (Hogeweg developed a large number of extensions for the original model was in large part responsible for the model’s success).

One of the main advantages of the CPM, along with ease of software implementation, is that it provides a straightforward “interface” for adding new phenomena via the Hamiltonian. Anything that can be quantified as a scalar number can be controlled by adding a term to the energy function. Behavior that comes with a high energy cost is avoided, whereas behavior
Figure 2.13 – Some initial simulations using the CPM. Adapted from Glazier and Graner (33)
that reduces the total energy of the system tends to occur. An interesting aspect of this is that one does not have to specify exactly how the constraints are to be met. One specifies what is considered to be “good behavior” and “bad behavior” and the cells tend to behave “well”, although sometimes with unexpected results leading to more constraints to close certain energetic loopholes. Here we discuss a few important extensions of the original model that are of general interest. For a more complete inventory of CPM extensions see e.g. Balter et al. (35) and Merks and Glazier (36).

Perimeter Constraint

In the cell sorting example above, the cell interfaces tended to be minimized and this effectively resulted in the length of the cell boundaries also being minimized. Both behaviors are the result of using positive bond energies, which penalize cell-cell contact. However, in some cases real cells tend to maximize their contact with each other, corresponding to negative bond energy. Setting negative bond energies between cells in the CPM results in uncontrolled growth of the cell boundaries, as the boundary can grow without increasing the cell volume by distorting its shape or forming wrinkles. To avoid this unrealistic behavior, a new term can be introduced that is similar to the area term, except it is a function of the cell perimeter:

\[ H_{\text{sur}} = \lambda_s \sum_{\sigma \neq M} (s(\sigma) - S_\tau(\sigma))^2 \]

Anisotropic Differential Adhesion and Cell Shape

Convergent extension describes a process in which an “array” of cells converges laterally such that the columns intercalate, causing the number of rows to increase and the array to extend. In the specific case of Xenopus, convergent extension involves three steps. First the cells change shape from roughly cuboidal into an elongated form along one axis. Elongation is followed by alignment, during which the cells align such that their long axes are roughly parallel, forming stacks. These stacks then merge, like shuffling a deck of cards, causing the tissue to elongate (Fig. 2.14).

Zajac et al. (37) proposed an analytical model for this process based on the notion of anisotropic differential adhesion in which the local bond strength between two elongated cells is a function of their orientation and position
Figure 2.14 – During convergent extension, parallel stacks of cells converge laterally, extending the tissue in the perpendicular direction. Adapted from (33).

at the point of contact. Adhesive bonds between stacked parallel cells are the strongest, whereas cells that are aligned head to toe bond weakly. Zajac et al. (34) then followed up on their analytical model with an extended CPM that included both elongated cells and anisotropic differential adhesion. Cell shape was controlled by introducing an energy term that is a function of the moment of inertia $I$ of the cell. The moment of inertia is a scalar value that is dependent on the cell shape and its axis of rotation. For example the moment of inertia of an ellipse around its center is $I = \frac{4}{3}(a^2 + b^2)$, where $A$ is the area and $a, b$ are the lengths of its semi-axes. For objects composed of discrete points such as CPM cells, the moment of inertia is given simply by

$$I = \sum (x - \bar{x})^2$$  \hspace{1cm} (2.7)

where $x$ are the lattice sites belonging to the cell and $\bar{x}$ is the center of mass of the cell. Then, together with the area constraint, the shape of the cell can be controlled by adding the term

$$H_1 = \lambda_1 \sum_{\sigma \neq M} (I(\sigma) - I_0)^2$$  \hspace{1cm} (2.8)

to the Hamiltonian, where $I_0$ is the target moment of inertia of the cells.
Anisotropic differential adhesion was modeled by altering the bond energies. So far we have assumed that the bond energies are constant for each pair of cell types. Zajac et al. (34) defined the bond energies as functions of the cell types and the relative position of the contact boundary to the cell centers in polar coordinates $r, \theta$.

\[
J(r, r') = J_{\tau(\rho), \tau'(\rho')} - \Delta(r)\Delta(r') \tag{2.9}
\]
\[
\Delta(r) = \alpha(\tau(\rho)) \epsilon(\sigma) r \sin(\theta) \tag{2.10}
\]

where $\alpha$ is the upper limit on anisotropy, $\epsilon$ is the eccentricity of the cell, which is 0 for an infinitely elongated cell and 1 for a circle, $r$ is the distance between the cell center and the point of contact and $\theta$ is the angle between the point of contact and the long axis of the cell. Using these extensions they showed how a random aggregate of elongated cells could be made to elongate, align, converge and extend based simply on anisotropic differential adhesion.

**Chemotaxis**

Real cells are often capable of detecting guiding signals in their environment that influence their direction of movement. When the signal takes the form of a diffusible chemical the phenomenon is called chemotaxis. Chemotactic cells sense a gradient of concentration of the chemical in their surrounding environment and adjust their movement according to the perceived direction of the source. If the cells move towards the source, then positive chemotaxis is said to occur and the chemical is a chemoattractant, or if the cells move away from the source then negative chemotaxis occurs and the chemical is a chemorepellent.

An intensely studied example of chemotaxis is the aggregation of *Dictyostelium discoideum*, soil dwelling myxamoebae. When nutrient levels in their environment are high, each cell lives independently and remains mostly vegetative, but when nutrient levels drop the scattered cells aggregate to form a crawling slug (Fig. 2.16). The slug moves much faster than the cells alone and is an efficient way of exploring the environment for food. If the cells are unsuccessful in their quest, the slug “metamorphoses” into a fruiting body composed of a long stalk that stands vertically with a body of spores at the top. Spores that are transported by chance to more favorable environments germinate and begin the cycle again.
Much of this process is guided by chemotaxis. When the cells begin to starve, a few autocyling cells start producing cyclic adenosine monophosphate (cAMP) and release it into the environment in regular pulses. Each cAMP pulse diffuses out from the cell and is sensed by neighboring cells which then produce their own cAMP pulse, but only if they have not already received a pulse recently. This refractory behavior guarantees that the cAMP signal is propagated outwards away from the original source (Fig. 2.16a). The cells are capable of sensing the direction of the wave and for each wave that passes through they move a little towards the source, eventually converging on the signaling center (Fig. 2.16b-c) and forming the slug. Within the slug,
the autocycling cells are located at the anterior and drive the slug forward by releasing waves of cAMP that travel posteriorly.

Savill and Hogeweg (39) used an extended CPM to model Dictyostelium aggregation and the resulting crawling slug by introducing multiple new elements to the CPM framework. First, the authors extended the formalism to three dimensions. This is straightforward for CPMs, since nothing presented so far is intrinsically limited to two dimensions. The authors then coupled the CPM with an implementation of the Panilov and Winfree model for excitable
systems and used it to simulate the cAMP dynamics. The model consists of a pair of partial differential equations, the spatial domain of which is a lattice of identical dimensions as the CPM. The equations are numerically integrated on the lattice points occupied by the cells, whereas in between the cells on the underlying substrate only diffusion occurs.

Chemotaxis is introduced into the CPM based on the cAMP concentration field defined by the PDE model. Instead of inserting a new term to the effective energy function, $\Delta \mathcal{H}$ is modified directly for each spin copy attempt:

$$\Delta \mathcal{H}' = \Delta \mathcal{H} - \mu (c(i,j) - c(i',j'))$$

where $c(i,j), c(i',j')$ are the concentrations at the target site and chosen neighbor site respectively and $\mu$ is chemotaxis strength. This scheme biases spin copies in the direction of the gradient, towards the autocycling cells and the aggregation center. Note that, although simulated separately, the CPM and PDE models interact with each other. The PDE system depends on the location of the cells and in turn the cells move according to the concentrations determined by the PDEs. Cellular automata models that are coupled to a system of PDEs are called a hybrid cellular automata.

Savill and Hogeweg (39) were able to successfully model Dictyostelium through the aggregation phase to the crawling slug phase using a single model with fixed parameters, showing how this relatively simple system of differential adhesion, chemotaxis and cAMP is sufficient to explain the major features of the process (Fig. 2.17(a)). Furthermore they demonstrated the superior crawling efficiency of an aggregate of amoeba versus single cells. Later Marée and Hogeweg (40) used a similar CPM to model the later stages of fruiting body formation with similarly impressive results (Fig. 2.17(b)).

### 2.8.5 Other Applications of the CPM

A few prominent early examples of CPM models have been shown here that introduced popular extensions to the original model. There are a host of other extensions, reviewed in Balter et al. (35). Since then, many other models have been published. Here we list a few which may be of general interest, in chronological order: Hester et al. (41) studied somitogenesis in the chick embryo using a hybrid CPM model; Larson et al. (42) studied patterning in the Drosophila pupal eye; Poplawski et al. (43) modeled the invasiveness of avascular tumors in 3D; Vasiev et al. (44) modeled the gastrulation of
2.8 – The Cellular Potts Model

(a) A 3D model of the aggregation and crawling slug phases. Adapted from Savill and Hogeweg (39).

nor the stalk cells are chemotactic, the tip elongates downwards at a greater speed than the chemotactic upward motion. The stalk tip reaches the base around 20 min after entering the prespore region, which is just as fast as observed in experiments (14). The mechanism can be elucidated as follows (see Fig. 2). Downward-moving cAMP waves periodically trigger an upward-directed chemotactic response. Because chemotactically moving cells push cells in front of them and pull cells that are behind them (because of cell adhesion), the equivalent of pressure differences are created. Therefore every cAMP wave is accompanied by a pressure wave. Moreover, because of cell adhesion, cells start moving before the cAMP wave arrives, because they are pulled toward the chemotactically moving amoebae. And when the cAMP wave has passed, cell motion continues for a while, because cells that are located just below keep on pushing the cells upwards. Hence the pressure waves are much broader than the cAMP waves, and upward motion is far more gradual, compared with the pulsatile cAMP signal. The pathfinder cells are pushed and pulled by these pressure waves, which results in a peristaltic motion of the stalk tip. The pathfinder cells, and along with them the stalk cells, are thereby squeezed downward. Newly recruited stalk cells are transported through the tube by the combination of pushing at the tube gate, attributable to the surface tension between PstA and stalk cells, and pulling at the stalk tip, attributable to the peristaltic motion. Although an elongated shape moving against a flow has a very strong tendency to bend sideways, the mechanism revealed by our model very efficiently restores any such deviation. Fig. 3 shows that even if initially the stalk is bent 90°, it extends downward again after only 15 min. When the stalk tip is not precisely pointing downward, the cAMP waves reach one side earlier. Hence, the moment this side is pushed, the other side is still pulled. This force efficiently transports the cells inward, instead of downward, and restores the original orientation. Peristalsis also explains the position of the pathfinder cells. When more than half of the pathfinder cells happen to

(b) The culmination phase. From Marée and Hogeweg (40).

Figure 2.17 – Modeling the life cycle of Dictyostelium using a CPM model extended with chemotaxis and coupled with a cAMP signaling model.
the chick embryo; Marmottant et al. (45) used a CPM to validate a model of the effective viscosity of cell aggregates. Shirinifard et al. (46) studied vascular tumor growth and Merks et al. (47) used a CPM for a unified model of vasculogenesis and angiogenesis.

2.9 Discussion

An important feature of CA models is that they are easy to implement as software programs. Since they are typically not computationally intensive, they can be written in virtually any computer language, from user-friendly-but-slow scripting languages to high performance compiled languages such as C. For the most part, they also do not require sophisticated data structures or algorithms. Also, the discretization of space into a regular grid is a big advantage from a computational standpoint since we can map the spatial domain directly to computer memory. Given any position in the lattice, we know \textit{a priori} exactly where to look in memory for any objects within a given radius of that position.

Given their qualitative nature, lattice-based methods free the modeler from choosing and implementing appropriate quantitative sub-models for collisions, cell motility, cell adhesion, etc., as well as an appropriate numerical scheme to integrate the resulting equations. Instead, the modeler can focus on programming “agents” with a discrete qualitative rule set that captures the basic behavior of a cell, with a view to demonstrate that a population of these agents have a non-obvious emergent property.

These technical advantages can be true enablers for researchers in theoretical biology who do not have a computer science background and who want to write small programs that “just work” rather than writing high-performance software for more detailed simulations. Lattice-based methods lower the entry barrier for developing new models, and allow for rapid prototyping of new ideas. This is perhaps one reason why lattice models are popular in the field of Artificial Life, which doesn’t seek to study life as it is, but “as it could be”. Lattice models allow researchers in Artificial Life to work with abstract principles without getting bogged down in detail. On the other hand, lattice models can also be useful for quickly testing the feasibility of a real-world conceptual model developed by a biologist based on experimental data. Simulations can support the biologist’s model, as was the case with the
BCR thinning model, or point out shortcomings if simulation results are not as expected.

The LGCA and CPM models presented here were adapted from quantitative physical models, where their parameters and independent variables had physical meaning. Unfortunately, when these frameworks are applied to biological systems their parameters are divorced from mechanistic underpinnings, since the models are applied based on analogy and are not derived from first principles. In particular, the dynamics of the CPM model are clearly not realistic, since a cell can spontaneously appear within another cell, albeit briefly since this behavior is "penalized" by the Hamiltonian. Furthermore, time has no meaning in a CPM simulation, since as of yet there is no physical basis for the CPM temperature and the bond energy and temperature parameters have arbitrary units (48). Therefore one must take care in interpreting a CPM timeseries, since Monte Carlo steps will not necessarily translate to some fixed unit of time.

Despite these issues, CPMs are still very useful because they are predictive and simple to work with from a technical standpoint. They are especially suited to systems in which differential adhesion plays a major role, however they are not suitable for modeling all systems:

- Each cell in a CPM model has identity, but its constituent lattice sites do not. Sites are randomly added and removed from a cell, and so it becomes very difficult to differentiate a cell internally, like introducing cell polarity, for example.

- On the same note, it is impossible, or at least extremely impractical, to represent individual mechanical elements, such as springs, or chain of such elements.

- Cells are coarsely represented and do not have the regular polygonal cell shapes that are common in cell aggregates. One potential solution would be to use a finer grid, however this does not scale well, especially in 3D.

- The dynamics of the CPM are inherently viscous, and it would be very impractical to try to skirt this and model more rigid bodies such as bacteria.
Chapter 2 – Lattice-based Models

If any of these issues is a significant concern, a lattice based model may not be the right solution. Instead lattice-free models, presented in the following chapter, may offer a better alternative. Although lattice-free models allow greater freedom for detail and can potentially be fully mechanistic, these models are more complex and require a greater time investment from the researcher. But in cases in which cell shapes, internal cell differentiation, the trajectories of individual motile cells within an aggregate and the mechanical response of a cell aggregate to deformation, etc., are important, then lattice-free models are likely to be more suited to the task.