Massachusetts Institute of Technology
Department of Electrical Engineering
and Computer Science

Proposal for Thesis Research in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

Title: Creating a Programmable Material

Submitted by: Radhika Nagpal
9 Hawthorne Place, 7H
Boston, MA 02114

____
(Signature of Author)

Date of Submission: December 13, 1999
Expected Date of Completion: May 2001
Laboratory: Artificial Intelligence Laboratory

Brief Statement of the Problem:

Rather than build precisely engineered mechanical structures, one could imagine programming complex static and dynamic structures from a programmable material, consisting of millions of distributed sensors and actuators. Example applications might be a flexible car surface that can change structure exactly at the point of impact or a programmable assembly line that moves objects by producing ripples in specific directions. The goal of this thesis is to demonstrate the feasibility of programmable materials by providing a model for a programmable material and a programming paradigm for engineering global behavior. This thesis proposes a model inspired by epithelial cell sheets - the programmable material is composed of identically programmed, connected, mechanical cells that organize into complex structures through the coordination of local shapes changes. The thesis will investigate algorithms for programming static and dynamic global shapes using only local communication, local sensing and local actuation and will provide a framework for reasoning about the dynamics of the system and the robustness of the algorithms. The ultimate goal is to develop a programming language (with primitives, means of combination and abstraction) for engineering global shape change on a programmable material.

Massachusetts Institute of Technology
Department of Electrical Engineering
and Computer Science
Cambridge, Massachusetts 02139

Doctoral Thesis Supervision Agreement

TO: Department Graduate Committee
FROM: Professor Gerald J. Sussman

The program outlined in the proposal:

TITLE: Creating a Programmable Material
AUTHOR: Radhika Nagpal
DATE: December 13, 1999

is adequate for a Doctoral thesis. I believe that appropriate readers for this thesis would be:

CO-SUPERVISOR: Professor Harold Abelson
READER 1: Professor Thomas F. Knight

Facilities and support for the research outlined in the proposal are available. I am willing to supervise the thesis and evaluate the thesis report.

SIGNED: ______________________________
PROFESSOR OF ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE

DATE: ______________________________

Comments:

____

____

____

Massachusetts Institute of Technology
Department of Electrical Engineering
and Computer Science
Cambridge, Massachusetts 02139

Doctoral Thesis Supervision Agreement

TO: Department Graduate Committee
FROM: Professor Harold Abelson

The program outlined in the proposal:

TITLE: Creating a Programmable Material
AUTHOR: Radhika Nagpal
DATE: December 13, 1999

is adequate for a Doctoral thesis. I believe that appropriate readers for this thesis would be:

CO-SUPERVISOR: Professor Gerald J. Sussman
READER 1: Professor Thomas F. Knight

Facilities and support for the research outlined in the proposal are available. I am willing to supervise the thesis and evaluate the thesis report.

SIGNED: ______________________________
PROFESSOR OF ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE

DATE: ______________________________

Comments:

Massachusetts Institute of Technology
Department of Electrical Engineering
and Computer Science
Cambridge, Massachusetts 02139

Doctoral Thesis Reader Agreement

TO: Department Graduate Committee
FROM: Professor Thomas F. Knight

The program outlined in the proposal:

TITLE: Creating a Programmable Material
AUTHOR: Radhika Nagpal
DATE: December 13, 1999
SUPERVISOR: Professor Gerald J. Sussman
CO-SUPERVISOR: Professor Harold Abelson

is adequate for a Doctoral thesis. I am willing to aid in guiding the research and in evaluating the thesis report as a reader.

SIGNED: ______________________________
SENIOR RESEARCH SCIENTIST
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

DATE: ______________________________

Comments:

____

____

____

____

____

____

1 Introduction

It is possible to imagine a flexible substrate, consisting of millions of tiny interwoven programmable fibers, that can be programmed to assume a large variety of global shapes. Not only could one design many complex static structures by programming a single substrate, but also create dynamic structures that react to, and affect, the environment. For example, a flexible car surface that can change structure exactly at the point of impact, rather than having specifically engineered crumple points; an airplane wing that can dynamically change shape to resist shear; a programmable assembly line or self cleaning surface that can move objects around by producing ripples in specific directions; or a manufacturing line that replaces precise mechanical engineering with programming. Programmable materials would make possible a host of novel applications that blur the boundary between computation and the environment.

Developments in microfabrication and microelectronic mechanical devices (MEMS) are already making it possible to bulk manufacture tiny computing elements integrated with microsensors and microactuators [2, 22]. It will soon become feasible to create programmable materials by embedding massive numbers of these computational elements into bulk materials or fabrics.

Fabrication, however, is only one part of the story. A question that still remains unanswered is - how does one achieve coherent and reliable global behavior from the local interactions of large numbers of identically programmed parts? Each individual element is likely to have limited resources, limited reliability and only local communication, sensor information and actuator impact. Current parallel programming assumptions, such as precisely engineered geometric interconnects and perfectly reliable parts, limit the ability to manufacture and embed large systems [13, 14]. Also, the role of the environment is rarely considered. New approaches are required that are insensitive to precise arrangement, make use of the corrective effects of feedback from the environment, and focus on coherent global behavior rather than precise individual answers.

Biology may hold the key to creating programmable materials. Biological systems regularly achieve coherent, reliable and complex behavior from the cooperation of large numbers of identically programmed (DNA) unreliable cells. In particular, pattern formation and morphogenesis (creation of form) in developmental biology can provide insights for creating programmable materials that can change shape [3, 20, 25, 19]. Even basic developmental processes, such as gastrulation, are based on deliberate cell migration and shape change and modeling such behaviors has been attracting increased attention from biologists [10, 12, 18]. Wolpert [24] argues that a "set of 'simple' universal principles may in fact be operative" in pattern formation and the same is likely to be true for morphogenetic processes. Understanding these underlying general organizational principles will have important implications, not only for programmable materials, but also for parallel and distributed computing.

Thesis Objective

The goal of this thesis is to demonstrate the feasibility of programmable materials by providing a mechanical model for a programmable material and a programming paradigm for engineering global behavior. The approach is inspired by epithelial cell sheets that organize into complex structures through the coordination of local shapes changes in identically programmed cells.

Proposed Research

- **Propose and evaluate a mechanical model for a programmable material.** The current model is a substrate composed of large numbers of identically programmed smart cells, each of which can sense local stress, apply local force and communicate with only a small neighborhood of adjacent cells. This proposal presents simulation results that suggest that a large range of global shapes can be achieved as simple compositions of local shape changes. Significant work will be required to extend these simulations and results to more complex structures composed of larger numbers of cells. The thesis will provide a complete dynamics model for simulating the programmable material and will evaluate the characteristics required for the local actuators, such as strength, stiffness, distribution, etc. Such information is important for ultimately building such a flexible substrate.

- **Investigate general techniques for engineering global shape change using only local communication, local sensing and local actuation.** This proposal presents some initial algorithms developed for cell differentiation and for determining global position and orientation. There are many instances in biology that will provide insights into mechanisms for coordinating cells to achieve global behavior. The focus however will be on engineering pre-specified global behaviors, rather than emergent behaviors or explaining particular biological phenomena. Part of the research is to determine what sets of behavior can be achieved by different techniques and how these behavior sets can be composed to achieve general programmable behavior.

- **Provide a computational framework for reasoning about the the behavior of the programmable material.** The goal is to provide a framework for analyzing the behavior of different algorithms and the interaction between the algorithms and the environment. The thesis will also investigate limitations to using local cell shape changes, as opposed to cell migration, replication or long-range forces. These limitations will provide insights into which biological structures can be formed solely from local shape change and which ones require migration or replication.

- **Provide a programming language for organizing global shapes on the programmable material.** The ultimate goal of this thesis is to demonstrate the feasibility of controlling a programmable material through a programming language that consists of primitives, with means of composing and abstracting these primitives. Preliminary research, as well as related work in biology, suggests such primitives do exist. The thesis will provide a complete simulation environment for programming and observing the dynamics of the system evolve over time.

The thesis will be conducted in the context of the Amorphous Computing Project [1], a project that aims at both identifying new parallel programming paradigms and inventing novel computational substrates to support such paradigms. This proposal presents preliminary research supporting this thesis topic. Section 2 presents an initial programmable material model and simulation results. Section 3 presents amorphous computing algorithms and programming languages for coordinating local interactions to achieve global behaviors. Section 4 concludes with the current research status and an approximate thesis timeline.

2 Designing a Programmable Material

Biology, in particular morphogenesis, gives us many examples of complex physical structures formed from the cooperation of large numbers of cells [3, 20, 25, 19]. However the majority of mathematical models have concentrated on pattern formation and regulation rather than the mechanical aspects of morphogenesis due to the difficulty of describing the dynamics of such systems. Some examples where this has been attempted are models of neural fold formation [11], cell traction models [12] and more recently modeling cells as tensegrity structures [10].

A seminal paper on mechanical models for morphogenesis is the paper on epithelial cell folding and invaginations by Odell, Oster, Alberch, and Burnside [18]. They present a mechanical model for how embryos form structures during development through the local deformations of individual epithelial cells. The cross-section of the embryo is modeled as a ring of cells, where each cell consists of fibers and contains an incompressible fluid. A cell can actively contract its apical (top) fiber and cause what is known as a *purse string contraction*. By varying the characteristics of the fibers, invaginations similar to those seen during gastrulation, neurulation and ventral furrow formation can be modeled.

The key insight is that the basis of the structure is a *simple mechanical cell* that can actively deform its own shape and sense deformations in its shape. Such a cell could form the basic element of a more general programmable material. This section presents a model for a programmable material composed of mechanical cells and presents simulations for cells connected in a ring and in a two dimensional sheet. The current simulations suggest that a wide variety of global shapes can be achieved as compositions of a few simple local cell shape changes.

2.1 The Mathematical Model for an Active Cell

The active cell forms the basis for the programmable material. The cell model is similar to the model proposed by Odell *et al* [18] for the epithelial cell. An active cell is modeled abstractly as a cube, whose membrane consists of active fibers. In two dimensions the cell is represented as a set of six fibers, one apical (top), one basal (bottom), two laterals and two diagonals. The fibers are constrained to be rigid. The purpose of the diagonals is to resist shearing forces, but they are weak to allow the cell to deform easily. A cell has polarity and can distinguish between the apical (top) side and the basal (bottom) side but not between lateral fibers. The cell may be filled with an incompressible fluid that tends to keep the cell volume constant.

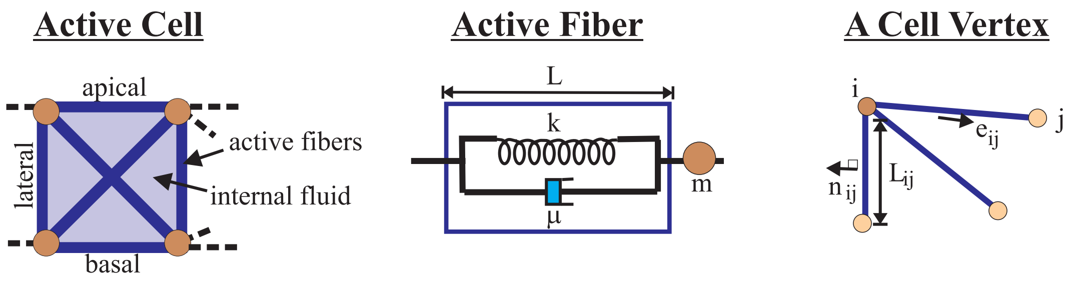


Figure 1: Model for an Active Cell

An active fiber is essentially a damped spring - a spring in parallel with a dashpot. The equation of motion for a fiber is:

$$m\frac{d^2L}{dt^2} = -k(L - L_0) - \mu\frac{dL}{dt} \tag{1}$$

where m is mass, L is the length of the fiber, L_0 is the rest length of the spring, k is the spring stiffness constant, and μ is the viscosity. In equilibrium the length of the fiber depends on the parameters k and L_0. The viscosity parameter μ affects the dynamic behavior - the higher μ is the slower the fiber moves.

The cell can be modeled by a set of differential equations, one for each vertex, describing the motion of each of the vertices of the cell. At any vertex of the cell, several fibers (from the same cell and possibly neighboring cells) apply force and the equation of motion for vertex i of a cell is:

$$m_i\frac{d^2\mathbf{x_i}}{dt^2} = -\sum_{j \in adj-nodes(i)} [k_{ij}(L_{ij} - L_{0ij}) + \mu_{ij}\frac{dL_{ij}}{dt}]\mathbf{e_{ij}} \tag{2}$$

where,

m_i	= mass at vertex i	$\mathbf{x_i}$	= vector position of vertex i
adj-nodes(i)	= vertices adjacent to vertex i	L_{ij}	= length of fiber (i, j)
$\mathbf{e_{ij}}$	= unit vector $i \to j$		

A cell can change its shape by contracting or expanding the fibers in its membrane. This is achieved by changing one or both of the following parameters for a fiber (i, j):

1. The rest length, L_{0ij}
2. The stiffness, k_{ij}

Either of these parameters change the amount of force applied by the fiber and affect the equilibrium shape of the cell and its neighbors. Each cell has a local program through which it can control the shape change. The cell may be programmed to trigger a particular change based on the local state or environmental cues. For example, a cell may contract its basal fiber if the length of its apical fiber exceeds some threshold or if it hears a message from its neighbors.

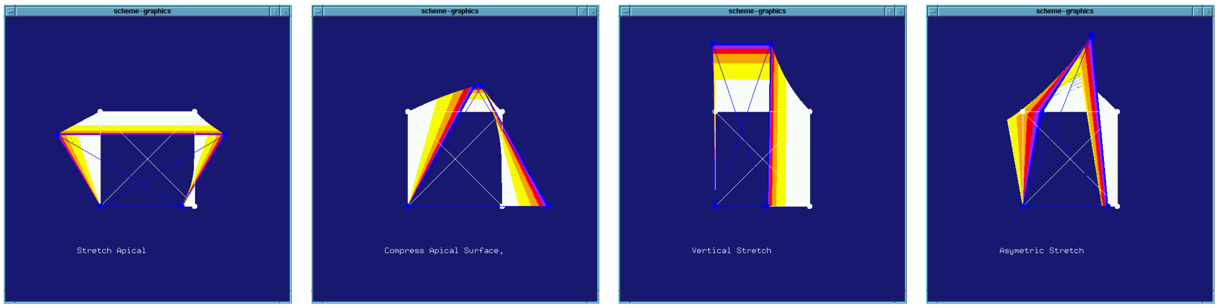
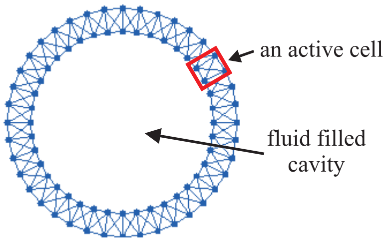


Figure 2: Shape changes in an Active Cell (from simulation)

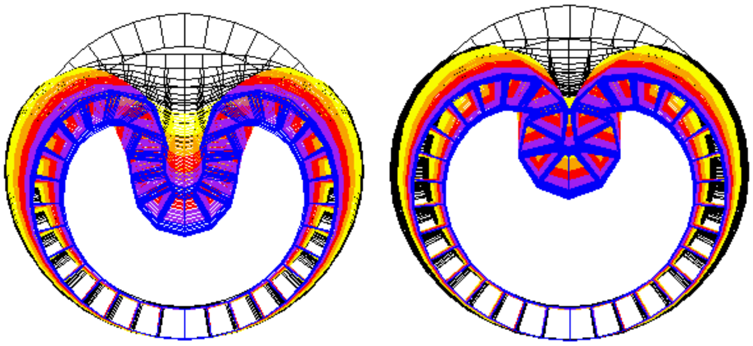
By programming the active cells, one can create many simple shapes - long columnar cells, short fat cells, triangular or conical cells. Figure 2.1 shows some of the shapes that can be achieved from a single cell. In each case, the cell actively changes the rest lengths of its fibers. The figures show how the shape changes over time. The different colors (white, yellow, orange, red, purple, blue) represent equal intervals of time allowing one to see the speed with which the shape changes and when equilibrium occurs. The simulation environment is described in more detail in section 2.4.

2.2 A Ring Composed of Active Cells

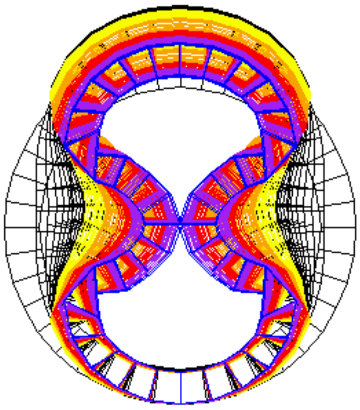
The model for the ring is similar to that used by Odell *et al* [18] and uses the active cells as described in the previous section. The ring contains a fluid that can leak when the ring is deformed. To some extent the ring represents the behavior of a closed sphere or cylinder in three dimensional space. These are some examples of shapes that can be formed from the ring.

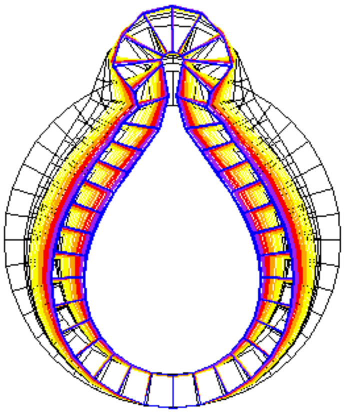


Gastrulation and Ventral Furrow: The left picture approximates gastrulation in organisms such as a sea urchin. The active cells contract their apical and lateral fibers, with more contraction happening at the tip of the invagination. Eventually the invagination reaches through the body to form the gut. The right mimics ventral furrow formation in the drosophila. Cells use apical contraction to form a tube on the ventral side of the cylindrical embryo (similar examples in [18]).

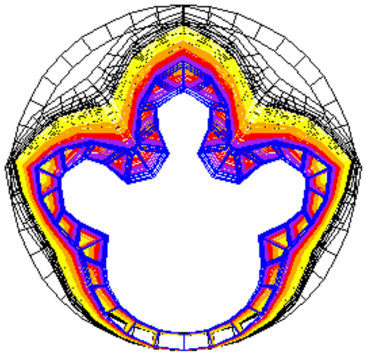


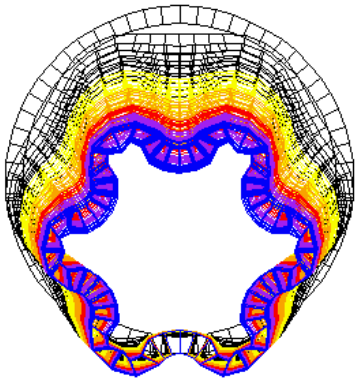
Cleaving and Bulb formation: The ring (or sphere) can form two smaller rings, using local contractions similar to those in the gastrulation example. This could represent the cleaving of a cell through the actions of fibers distributed in the cell membrane. The picture on the right shows bulb formation, which is simply the opposite mechanism to ventral furrow formation.





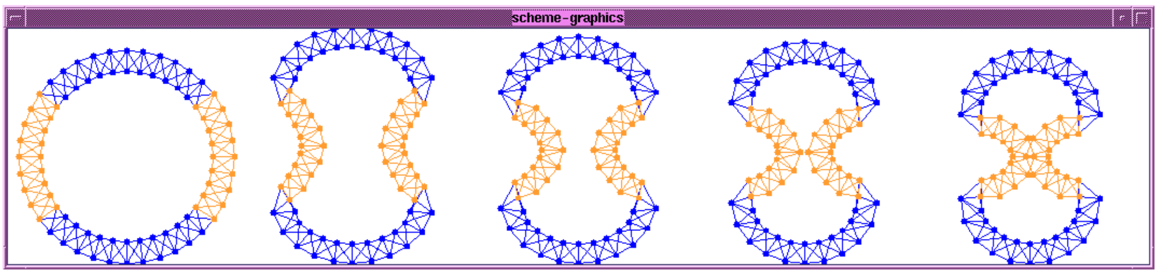
Fingers and formation of Eyefolds: Bulbs can originate at more than one site to produce fingers. Deepening invaginations between the fingers keep them separate. The second example shows the formation of folds in the ring through alternating cells with apical contractions and basal contractions. Folds are common during development (chick cornea), however usually due to differential growth.





The figures were generated by simulating a more generalized version of the model described in the previous section. The active cells were programmed to change rest lengths at the beginning of the simulation. The shapes are robust to changes in the parameters and small changes in the fiber characteristics or active cell rest lengths have little effect on the end result. Again, the different colors (black, yellow, orange, red, purple, blue) represent equal intervals of time giving an indication of how, and the speed with which, the shape changes over time.

Although the shapes above are biologically inspired, the goal is not to model biology. Practically any shape can be formed, limited only by the number of cells, their size and and the fact that they remain connected in a ring. More importantly, the same simple local shape changes in individual cells are reused and the global shape can be expressed as a composition of individual cell shape changes. For example if we look at the ring cleaving into two, we simply have two regions (orange) where cells are actively contracting their apical fibers to form triangular shapes.

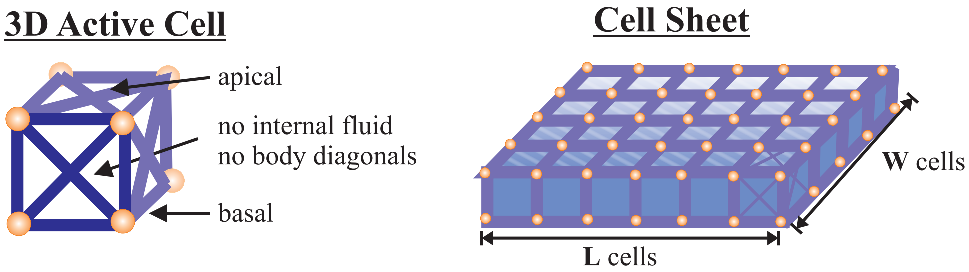


In section 3 we show that, using simple pattern formation techniques from amorphous computing, one can write a single program for a cell so that the cells in the ring collaborate to form the desired global shape.

2.3 A Sheet Composed of Active Cells

The shapes that can be formed with a ring are limited. A more interesting structure is a flexible sheet. One can imagine a wide variety of shapes that can achieved by folding the sheet in various ways. This section presents the model for a sheet of active cells and this work is currently in progress.

The two dimensional model of the cell is extended to three dimensions. The cell is a cube where each edge is a fiber, and there are two diagonal fibers on each face. The mass is concentrated at the vertices. The face diagonals resist shear and maintain the stability of the cell; without them the cell would collapse. Body diagonals are not needed. The cell still maintains a polarity and can tell the apical fibers from the basal fibers. The cell does not contain any internal fluid, since the internal fluid does not contribute significantly to the ability of the cell to form different shapes and unnecessarily introduces many complications into the three dimensional mathematical model.

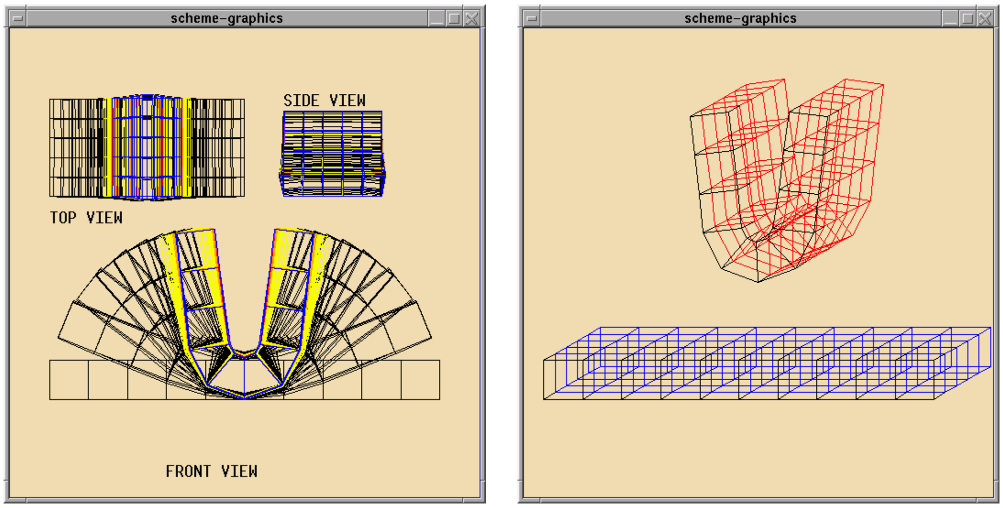


The sheet of cells consists of a rectangular grid of cells attached on their lateral sides and sharing lateral fibers. All the cells have identical stiffness k, viscosity μ and mass m. As before, the cell can change its shape by changing either the stiffness or the rest length of any of its fibers. Figure 3 presents some preliminary examples of simple shapes formed from the sheet. The shapes are simple extensions of the mechanisms used on the ring.

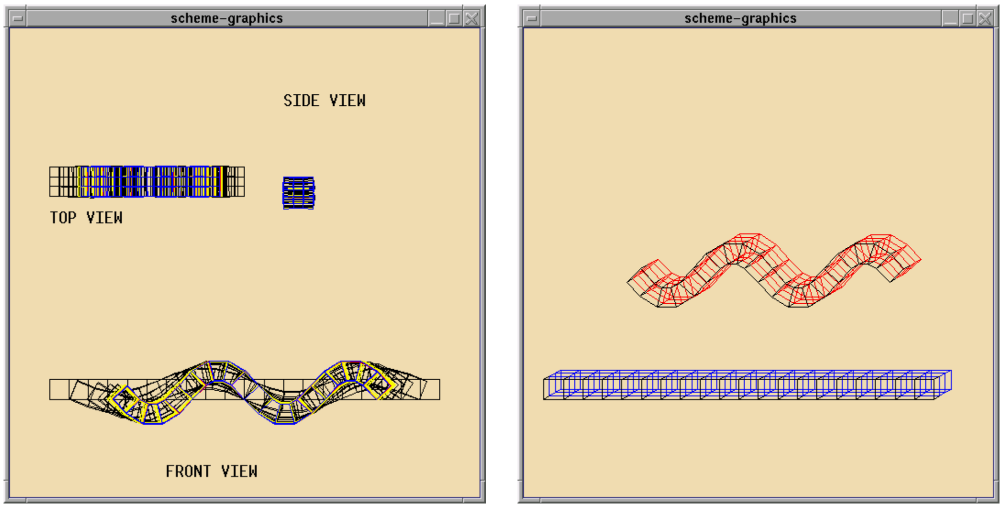
Already in generating these simple shapes many questions arise:

- Polarity: What kinds of cell polarity are needed? To what extent can this polarity be discovered, or must it be innate? For example, for the sheet to fold along a line, the cell contracts only two of its apical fibers that are opposite each other.
- Fiber Parameters: Is it necessary for fibers to change their stiffness as well as their rest lengths? How does that affect the shapes formed? How strong do the fibers have to be?
- Local vs Global Forces: Does a sheet of cells have the same kind of stability that a ring of cells has, given that the ring contains an internal fluid that transmits long range forces?

Fold Formation



Multiple Folds



Tube Formation

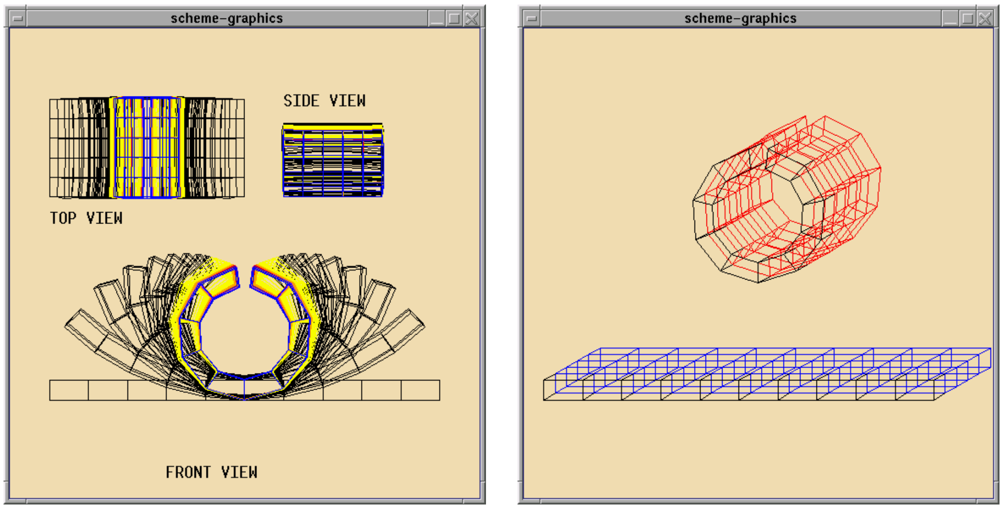


Figure 3: Shapes formed by a Sheet of Cells

- Geometry: Does the fact that the cells are connected in a rectangular grid limit the basic shapes that can be formed? How large a sheet is needed to produce interesting shapes?
- Limitations: Lastly, what shapes will require additional constraints such as gravity, friction or adhesion? What shapes require cells to migrate, replicate or dissociate?

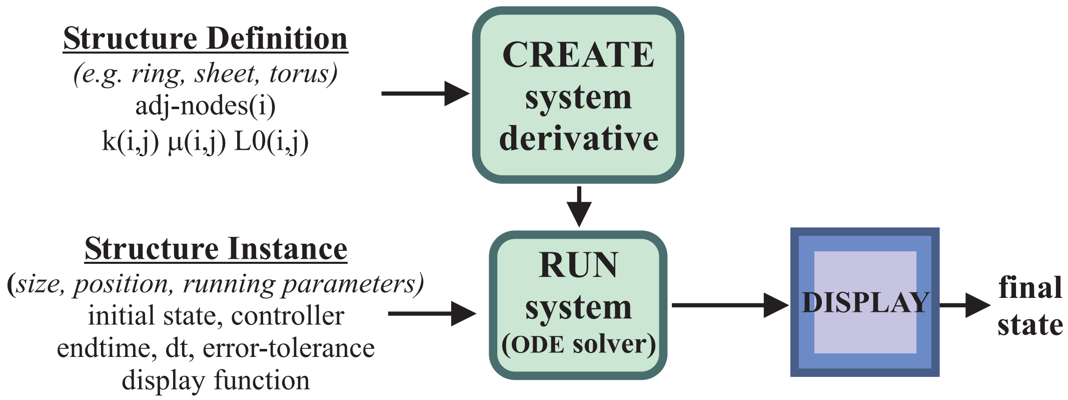
Some answers may be available in research on morphing and reconfigurable robots [26]. A recent project [27], under the Smart Matter Project at Xerox, is exploring producing global shapes from large numbers of identical rigid modules that can move past one another. This project will provide many interesting insights into the pros and cons of using deformable cells vs mobile cells.

Currently the focus is on experimenting with different basic shapes and studying geometric arguments for answering the questions above. The next step is to determine mechanisms for composing these basic shapes to produce more complex static and dynamic shapes. For example folds could be composed to generate more complex geometric structures - cubes, ridges, or, more ambitiously, origami. Examples of dynamic shapes are - a ring that locomotes by extending pseudopodia; or traveling waves on a cell sheet, that result either in movement of the sheet or movement of an object on the sheet.

The next section describes the simulation environment and discusses several issues that must be resolved in order to simulate larger structures and more complex shapes.

2.4 The Simulation Environment

A programmable material simulation environment has been designed that allows one to provide a high level definition of the structure layout (e.g. ring, sheet, torus) and then automatically generates a simulator for that type of structure. The simulator can then be run for structures with different sizes and different cell programs.



The simulator generalizes the cell model presented in section 2.1 to describe any structure composed of active cells. A structure can be described as a set of vertices and fibers, and equations of the form (2) can be written for each vertex. The result is a set of non-linear differential equations whose dynamics can be observed by numerical integration. There are many sources for information on describing and simulating dynamic systems [23, 4, 8].

In the simulation environment, a structure layout is described by an adjacency set function *adj-nodes* and lookup functions for k_{ij}, μ_{ij} and L_{0ij}. An appropriate system derivative is automatically generated. The system derivative is then fed to the simulator run loop along with an initial state, a controller, a display function, and numerical integration parameters such as total simulation time, step size and error tolerance. The initial state is an object consisting of a vector of vertex positions and initial values for k, μ and $L0$. The controller represents the programmatic control of the cells. At each step the controller can modify the values for k, μ and $L0$ for any fiber. More work is needed on to make the cell programming interface more intuitive.

The simulation environment is implemented in the language *Scheme* and uses the mechanics library which provides an ordinary differential equation solver, with adaptive step size and a variety of different methods, as well as appropriate mathematical constructs for solving linear equations. An important advantage of using scheme is the interpretive environment. The programmable material is a scheme object that can easily be manipulated and probed, or restarted with a different controller. In addition one has access to a real language with abstractions and a syntax that minimizes unnecessary debugging.

2.4.1 Simulating Large Structures

In modeling the programmable material, we make the assumption that the mass to damping $(m : \mu)$ ratio is so small that the inertial motion is negligible and can be ignored. This is similar to an assumption made in [18]. The reason this assumption makes sense is because the mass of an individual cell is extremely small compared to the size of the overall structure and the damping is significantly higher than the viscosity of water[1]. As a result, the left hand side of equation (2) can be set to zero. The system is then represented by a set of implicit differential equations $A(z)dz/dt = b(z)$, where z represents the vector coordinates of each of the vertices.

Given this model, if cells only change the fiber rest lengths L_0, then the parameters k and μ do not affect the equilibrium state of the cell sheet. They affect only the dynamics - higher k and higher μ increase the speed with which the structure reacts and reaches equilibrium. The structure moves slowly towards the equilibrium posiition and does not oscillate. Key to this behavior is the assumption that the masses are extremely small in comparison to k and μ.

The problem with using this assumption is that numerical integration involves solving $dz/dt = A(z)^{-1}b(z)$ at each step. For a large structure, the number of equations is very large. For example, for a L x W sheet of cells, the number of vertices is approximately $2LW$ and the number of scalar equations is $6LW$. Luckily, a vertex is only connected locally to a few fibers, which means the equations are very sparse, and the degree of connection remains constant even as the structure gets larger. Currently a sparse linear equation solver is being used, however it is memory intensive as such a large number of equations must be assembled and back substituted. It may be necessary to exploit the structure of $A(z)$ as a band matrix to achieve the necessary speed and memory

[1] A detailed argument is presented in [18]

requirements. It also may be possible to use an iterative method to solve directly for the equilibrium state for simple static shapes.

An alternative approach is to simulate the complete dynamics, which does not involve implicit equations at all, and choose very small masses for the vertices. This has different pitfalls as discussed next.

2.4.2 Simulating the Complete Dynamics

One can simulate the complete dynamics of the system by converting the second order equation (2) to two first order equations using a substitution of variables. For any vertex i, there is a vector position $\mathbf{x_i}$ and a vector velocity $\mathbf{v_i}$ $(= d\mathbf{x_i}/dt)$. Equation (2) becomes:

$$\frac{d\mathbf{x_i}}{dt} = \mathbf{v_i} \tag{3}$$

$$\begin{aligned} \frac{d\mathbf{v_i}}{dt} &= \frac{1}{m}\left(- \sum_{j \in adj-nodes(i)} [k_{ij}(L_{ij} - L_{0ij}) + \mu_{ij}\frac{dL_{ij}}{dt}]\mathbf{e_{ij}} \right) \\ &= \frac{1}{m}\left(- \sum_{j \in adj-nodes(i)} [k_{ij}(|\mathbf{x_i} - \mathbf{x_j}| - L_{0ij}) + \mu_{ij}(\mathbf{x_i} - \mathbf{x_j}).(\mathbf{v_i} - \mathbf{v_j})]\mathbf{e_{ij}} \right) \end{aligned} \tag{4}$$

The system derivative thus becomes trivial to compute and does not have the same memory problems as with the implicit equations. However in order to produce similar behavior the vertex masses must be extremely small and the damping very high. This results in extremely *stiff equations*, which means that reasonable error tolerance can only be achieved by taking extremely minute steps and the numerical simulation is unable to make any progress. The smaller the masses, the stiffer the equations get. Larger masses result in non-stiff equations and oscillating structures.

This behavior can be understood to some extent by analyzing the simple linear spring system (equation (1)) which has an exact solution $x(t) = Ae^{\lambda t}$ where λ represents the roots of the equation $m\lambda^2 + \mu\lambda + k = 0$. The mass m, damping μ and spring constant k must be chosen so that the roots are real ($\mu > 2\sqrt{mk}$) in order to have no oscillations. However this high damping also results in stiff equations. The case that is most amiable to numerical integration, and does not oscillate, is the case when the system is *critically damped*, i.e. $\mu = 2\sqrt{mk}$.

For the non-linear equation (2) it is difficult to analytically determine the critically damped parameters. Simulating the complete dynamics in reasonable time may require using implicit methods, such as Gear's method, to counteract the stiffness. In reality what matters is that the mass is sufficiently small compared to the damping so that the sheet approaches equilibrium slowly without oscillating. Simulating large structures and analyzing their behavior will require investigating these issues further.

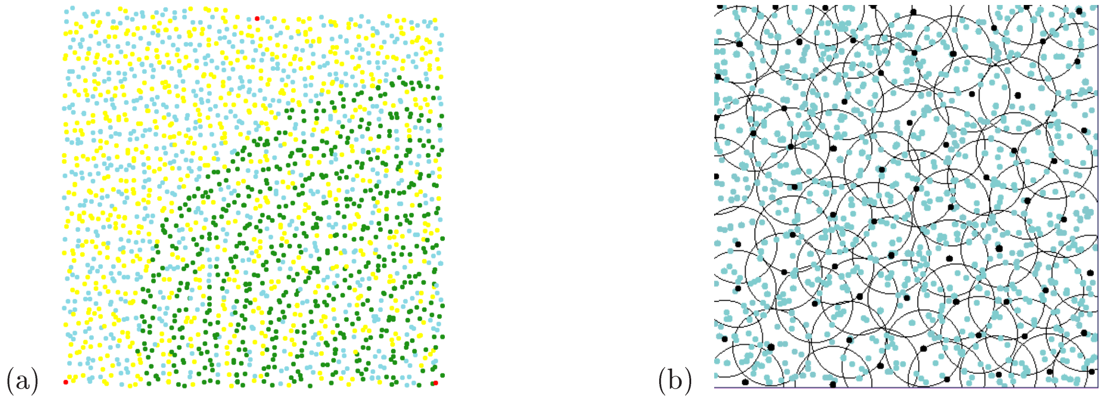


Figure 4: Amorphous computer simulations: (a) Gradient waves propagating from two anchors. (b) The result of the clubs algorithm. The dark processors are the leaders and the circles around them represent the area where they inhibited other leaders from forming.

3 Developing Programming Paradigms

A significant part of this thesis will involve addressing the important question of how one programs global behavior from the local interactions of large numbers of identically programmed elements - where each element has limited resources, limited reliability and only local communication, sensor information and actuator impact. This section presents some algorithms and languages that have been developed by my colleagues and I, that are likely to play a role in developing a language for the programmable material.

3.1 Amorphous Computing Algorithms

Our focus has been on algorithms for discovering global information, such as position or surface characteristics, and algorithms for generating global information, such as patterns. The algorithms have been developed in the context of an amorphous computer, which has a slightly different model from that of the programmable cell sheet. The most often used model for an amorphous computer consists of a large number of identical processors randomly distributed on a two dimensional plane surface. Individual processors do not have global ids, but instead have random number generators and processors have no apriori knowledge of topology, global position or even local orientation. A processor communicates with physically nearby processors within a fixed distance r, where r is much smaller than the dimensions of the plane.

Of the many amorphous computing paradigms that been researched, two key paradigms are:

1. **Gradients:** An important primitive that has been used in amorphous computing is the notion of 'gradients'. A gradient resembles a chemical diffusing away from a source, so that the concentration gets lower as one moves away from the source - giving an estimate of

distance and direction w.r.t. the source. On an amorphous computer, gradients are generated by a anchor processor sending a message to its neighbors, which is then forwarded on to their neighbors. The message maintains a hop count which the processors remember and increment before forwarding, ignoring any subsequent higher values and thus preventing the wave of messages from traveling backwards. The hop count provides an estimate of distance - if the count is h then the processor is roughly hr distance from the source - the accuracy of which can be significantly improved by smoothing techniques [15].

This turns out to be very powerful mechanism for generating global information, such as global position and pattern formation. In [15], we use gradients to generate a global coordinate system on top of randomly distributed elements that initially have no notion of orientation or position. This work is inspired by the drosophila embryo, where the segmentation into the head, thorax and abdomen is achieved by generating a pseudo coordinate system using chemical gradients [17]. The paper also presents theoretical and experimental results on the accuracy of gradients. Coore's thesis [6] presents a language for pattern formation where gradients direct the flow of the pattern. Topological diagrams are drawn by using the gradient to move towards, away from or orthogonal to the source. Our group is also exploring using gradients to determine surface characteristics, such as curvature, holes and surface geometry. Wolpert [24] presents numerous pattern regulation mechanisms that could be implemented using gradients on an amorphous computer.

In general, gross phenomena such as gradients are well suited to the amorphous computing environment because they are insensitive to the precise arrangement of processors or the occasional failure of processors, so long as the distribution is reasonably dense and the number of processors is reasonably large.

2. **Cell Differentiation:** Since the processors are identical, means for differentiation become crucial. In [16], we present an algorithm, called the *clubs algorithm*, for cells to elect local leaders and form local groups (clubs). The algorithm results in leaders that have a spacing of at least hr and at most $2hr$ between them with high probability, where r is the communication radius and h is a factor chosen by the user. The algorithm proceeds by each processor choosing a random number and then counting down until it reaches zero. At that point it declares itself a leader and sends a message to all neighboring processors within h communication hops. A processor that hears such a message discontinues counting down. Hence short range inhibition is used to prevent neighbors from being too close. Any processor that is not inhibited eventually attempts to become a leader, thus preventing leaders from being too far. The random number is chosen from a range that is proportional to h and the average neighborhood size. The algorithm completes in time proportional to the average neighborhood size.

   The advantage of this algorithm is its simplicity, which allows it to work correctly (and be analyzed) even with asynchronous or unreliable processors. This is achieved by focusing on providing the correct answer with high probability, rather than certainty. The algorithm has been used to self organize hierarchical groups [7]. Hierarchical organization is a recurring theme in society and biology and is valuable as a means of coordination as well as for increasing reliability. In general, the clubs algorithm provides a suitable mechanism for local leader election and competition on an amorphous computer.

One area that has not been explored is the use of *environmental paradigms*, algorithms that use environmental feedback and depend more critically on sensing and affecting the dynamics of the system. An interesting question is - what can one achieve, with and without sensing the physics? The current amorphous computing model, like other parallel computing models, is disembodied which makes it hard to explore such algorithms. The programmable material model will make it possible to explore such algorithms by allowing complex interactions between the programmable world and the physical world.

There are many possible sources of inspiration for such algorithms. One source is biology. Although modeling is rare, speculation abounds on the role various mechanical forces roles play in morphogenesis [3]. A programmable material model may be an interesting and practical way of exploring some of these theories. For example Odell *et al* [18] claim to present a purely sensing based algorithm for gastrulation, but in fact the algorithm depends critically on non-sensing prepatterning and our simulations show that the same final shape can be achieved without using sensing.

Research on locomotion (e.g. earthworm locomotion) and reconfigurable modular robots [26] provides important examples of sensor-based algorithms for dynamic shape change. Much of the research in multi-agent robot systems focusses on mobile entities which is less relevant. Control theory is also likely to have some insights. For example, Berlin [5] designed a bridge with columns that actively counter buckling by using distributed stress sensors and actuators to react to increased load on the column. There are many other examples, such as active control of vibrations [9] and adaptive aircraft wings [2]. However the focus tends to be more on global control and the mathematical properties (such as optimality) of the controllers, rather than the programming aspect of control.

3.2 A Programmable Material Language

Several languages have been developed for programming amorphous algorithms at the level of the individual processors. In this section, a language developed by Weiss [21, 1] is described, which may be extended to form a language for the active cells. The language was developed for programming biological cells. The main advantage of this language is it provides a very simple way for expressing the program for a single cell and incorporates many amorphous computing ideas.

The program for a cell consists of a set of independent rules, which is similar to the automata language for distributed systems [14]. A rule consists consists of a triggering event, a condition and an action. The state of a cell consists of boolean markers, and the markers can be set or unset, or set with a lifetime so that they act as timers. An event may be the reception of a message from a neighboring cell, or a marker whose timer expired. The condition is a boolean expression of markers. The possible actions are to affect the state or send a message.

Underlying this language are *gradients*. The system implements flat gradients, i.e. gradients that travel upto a certain distance but have no detectable slope. Flat gradients can be used as a way of marking regions. The only inter-cellular communication allowed is the initiation of a gradient. This language has mainly been used for pattern formation.

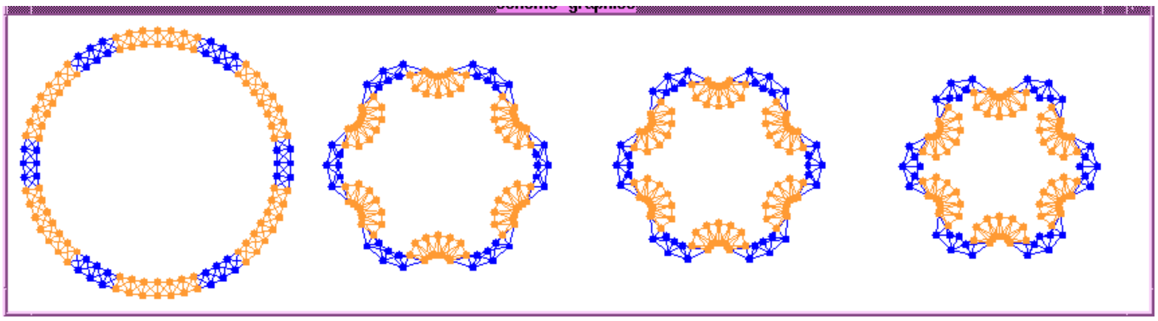
A Cell Program for Initiating Eyefolds: By extending the language to have sensor triggered events and actuator actions, we can write a pseudo program for initiating folds on a ring.

```
; THE CELL PROGRAM
; ================
((initial)                      ; EVENT: rule for when cell first runs
 #t                             ; CONDITION: none
 ((set TIMER (random))))        ; ACTION: set a timer to a random value

((TIMER)                        ; EVENT: timer expires
 (not inhibited)                ; CONDITION: have not been inhibited
 ((set fold-leader)             ; ACTION: start a fold and inhibit neighbors
  (send inhibitor-wave 3)       ;         within 3 hops from starting new folds.
  (contract-apical)))

((receive inhibitor-wave)       ; EVENT: inhibitor gradient reaches you
 #t                             ; CONDITION: none
 ((set inhibited)))             ; ACTION: inhibit the creation of a fold
```

The result of this program is that folds (or invaginations) appear at random places on the ring with a separation of between 3 to 6 cells. The program uses the *clubs algorithm*, which uses short range inhibition to ensure that folds appear with some preferred spacing. The program can be extended to cause cells close to the fold points to also contract their apical fibers. This program generates folds that are slightly different from this simulation example, where the folds are equidistant.



The language currently does not have all the features necessary to implement general shapes. For example, the gradients are flat and the program can not use the slope or direction of the gradient. In the gastrulation example in section 2.2, the actuation of the fibers decreases as one moves away from the original invagination point. This could be implemented as a reaction to the decreasing concentration of a chemical (gradient). Also boolean values may not be the most convenient way to express state in active cells and simple arithmetic may be useful.

3.2.1 Developing a High Level Language

One key piece of work that has emerged from the amorphous computing project is the Growing Point Language (GPL), which allows one to describe patterns at a high level using a botanical metaphor and then automatically translates that description into a program for the individual cell [6]. An interconnect pattern is described almost as one would describe a euclidean diagram, constructed by drawing lines with a ruler and arcs with a compass. Such a language not only provides a high level mechanism for describing patterns, but also provides a means for reasoning abstractly about what set of patterns can be generated. A goal of this thesis is to generate a similar high level language for describing global shapes. Origami, for example, is such a language and shapes are described as a series of global folds. There are likely to be other metaphors as well, in biology and in topology, that will provide inspiration for creating a language of programmable shapes.

The key is to find simple algorithms that can produce complex behavior and still be amiable to analysis. Ultimately, the most powerful techniques are likely to be the ones that tie computation intimately to the dynamics of the environment.

4 Conclusions

The aim of this thesis is to provide an example of how a programmable material can be designed, as well as shed light on the transformation from local interactions to global behavior. This document outlines the biologically inspired model for a programmable material and presents preliminary simulation results and algorithms.

The current focus of the research is on creating complex behavior on the programmable cell sheet, such as traveling waves and origami shapes. Experimenting with such complex behaviors is should reveal some general principles for shape formation and limitations of the model. Effort is also underway for building a simulation environment that combines a programming environment for the active cells with the dynamics simulation environment discussed in this proposal. The simulation environment will be key for exploring new algorithms and different programming constructs.

Future research will focus on high level languages for describing general shape construction. Such a language would ideally incorporate amorphous computing paradigms discussed here as well as new primitives that emerge. Future work will also include compiling the analysis techniques developed for the algorithms into a computational framework that can be used to reason about the system as a whole.

Thesis Timeline

Fall 1999	Dynamics of cell sheets, build simulation environment.
Spring 2000	Explore shapes, locomotion on cell sheets.
Summer 2000	Programming language design and implementation.
Fall 2000	Computational framework and analysis.
Spring 2001	Submission of completed dissertation.

References

[1] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight, R. Nagpal, E. Rauch, G. Sussman, and R. Weiss. Amorphous computing. *White paper*, 1999. http://www-swiss.ai.mit.edu/~switz/amorphous/.

[2] H. Abelson, A. Berlin, N. Cohen, L. Fogel, C. Ho, M. Horowitz, J. How, T. Knight, R. Newton, and K. Pister. Distributed information systems for mems. *ISAT (Information Science and Technology Study Group) Study*, 1995.

[3] J. Bard. *Morphogenesis*. Cambridge University Press, U.K., 1990.

[4] E. Beltram. *Mathematics for Dynamics Modeling*. Academic Press, Inc., San Diego, CA, 1987.

[5] A. Berlin. *Towards Intelligent Structures: Active Control of Buckling*. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, May 1994.

[6] D. Coore. *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*. PhD thesis, MIT, Department of Electrical Engineering and Computer Science, February 1999.

[7] D. Coore, R. Nagpal, and R. Weiss. Paradigms for structure in an amorphous computer. AI Memo 1614, MIT, 1997.

[8] G. Golub and C. Van Loan. *Matrix Computations, Second Edition*. John Hopkins University Press, Baltimore, 1989.

[9] S. Hall, E. Crawley, J. Howe, and B. Ward. A hierarchic control architecture for intelligent structures. *Journal of Guidance, Control and Dynamics*, 14(3):503–512, 1991.

[10] D. Ingber. The architecture of life. *Scientific American*, January 1998.

[11] A. Jacobson and R. Gordon. Changes in the shape of the developing vertebrate nervous system analyzed experimentally, mathematically and by computer simulation. *Journal of Experimental Zoology*, 197:191–246, 1976.

[12] A. Jacobson, G. Oster, G. Odell, and L. Cheng. Neurulation and the cortical tractor model for epithelial folding. *Journal of Embryology and Experimental Morphogenesis*, 96:19–49, 1986.

[13] F. Leighton. *Introduction to parallel algorithms and architectures*. Morgan Kaufmann, San Mateo, CA, 1992.

[14] N. Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers, Wonderland, 1996.

[15] R. Nagpal. Organizing a global coordinate system from local information on an amorphous computer. AI Memo 1666, MIT, 1999.

[16] R. Nagpal and D. Coore. An algorithm for group formation in an amorphous computer. In *Proceedings of the 10th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, October 1998.

[17] C. Nusslein-Volhard. Gradients that organize embryo development. *Scientific American*, August 1996.

[18] G. Odell, G. Oster, P. Alberch, and B. Burnside. The mechanical basis of morphogenesis: 1. epithelial folding and invagination. *Developmental Biology*, 85:446–462, 1981.

[19] J. Slack. *From Egg to Embryo, second edition.* Cambridge University Press, U.K., 1991.

[20] D Arcy Thompson. *On Growth and Form, abridged edition.* Cambridge University Press, U.K., 1961.

[21] R. Weiss. Programming colonies of biological cells. *PhD Thesis Proposal, MIT, Department of Electrical Engineering and Computer Science*, January 1999.

[22] R. Weiss, G. Homsy, and R. Nagpal. Programming biological cells. In *8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '98), Wild and Crazy Ideas Session*, October 1998.

[23] A. Witkin and D. Baraff. Physically based modeling: Principles and practice. In *SIGGRAPH '97 Course notes*, 1997.

[24] L. Wolpert. Positional information and the spatial pattern of cellular differentiation. *Journal of Theoretical Biology*, 25:1–47, 1969.

[25] L. Wolpert. *Principles of Development.* Oxford University Press, U.K., 1998.

[26] M. Yim. *Locomotion With A Unit-Modular Reconfigurable Robot.* PhD thesis, Stanford, Department of Computer Science, 1994.

[27] M. Yim. Modular robots. *part of the Smart Matter Project at Xerox PARC*, 1998. http://www.parc.xerox.com/spl/projects/modrobots/.