FINAL REPORT OF THE UCDHSC MATHEMATICS CLINIC
Modeling Tissue Dynamics

Taught by
Stephen C. Billups
University of Colorado at Denver
and Health Sciences Center
Mathematics Department
P.O. Box 173364
Denver, CO 80217-3364
Stephen.Billups@cudenver.edu
http://www-math.cudenver.edu/~billups/

Sponsored by
Randall Tagg
UCDHSC, Department of Physics

Participating students:
Deborah Batista, John Boren, Jon-Michael Brinkerhoff, Chris Comiskey, Asmaa Elmkhanter, Jesse Gilbert, Nathan Kurtz, Don McCuan, Patricia Moll, Sergei Mordovine, Megan Sawyer, Kim Story, Megan Van Dyke, Ena Vu, and Lea Wesbrook

Spring Semester 2007

Contents

Acknowledgements

The success of the Mathematics Clinic program is largely dependent upon identifying an important and mathematically interesting problem. I am therefore indebted to Randy Tagg for pointing us toward this interesting problem domain. Of course, the most important component of any Math Clinic is the students. I am grateful for the enthusiasm, creativity and hard work of all the students who participated in this project. They are

Deborah Batista,
John Boren,
Jon-Michael Brinkerhoff,
Chris Comiskey,
Asmaa Elmkhanter,
Jesse Gilbert,
Nathan Kurtz,
Don McCuan,
Patricia Moll,
Sergei Mordovine,
Megan Sawyer,
Kim Story,
Megan Van Dyke,
Ena Vu,
and Lea Wesbrook

Stephen C. Billups,
Math Clinic instructor

Chapter 1

Introduction

This report summarizes the results of the Spring, 2007 Mathematics Clinic at the University of Colorado at Denver and Health Sciences Center. The Clinic was sponsored by Randall Tagg (UCDHSC, Dept. of Physics), in collaboration with Arlen Meyers, (UCDHSC, Dept. of Otolaryngology) who is investigating a novel approach for early detection of oral cancer that incorporates information about structural changes of the tissue. A critical component of this research is to enhance current understanding of how tissue grows and differentiates.

The focus of the clinic was to develop computational models for tissue dynamics. The models handle populations of cells, modeling the life cycle of individual cells, (including mitosis (cell-division), differentiation, and apoptosis (cell-death)) as well as interactions between cells (such as cell signalling, adhesion, and migration) . By manipulating various controls in these models, we hope to identify hypotheses which will guide experiments in the Tagg laboratory.

The fifteen students in the clinic were divided into five teams. Two of the teams developed computational models of the cell cycle; two developed tissue models based upon populations of cells; and the final team evaluated various publicly available cell-modeling software packages. The results of each of these teams are described in the following five chapters.

Chapter 2

Summary of Cell Cycle and Dynamical Systems with Models of the Cellular Checkpoints

By Asmaa Elmkhanter and Megan Sawyer

2.1 Abstract

Understanding the basics of cellular checkpoints and dynamical systems is an important pre-requisite for modeling the reproductive behavior of cells. This paper attempts to present a suitable refresher in biological and dynamical systems. We also discuss a model of the metaphase/anaphase checkpoint and extrapolate on ways to create a more biologically accurate model.

2.2 Introduction

The cellular reproduction cycle is an important element in the growth and differentiation of tissue populations. It is important to model the effects of both intracellular and extracellular influences on the rate and course of the cell cycle. Often when observing cellular behavior, the use of dynamical equations is necessary to appropriately model the events. This paper

will present a refresher on the basic background for the cell cycle, the chemistry behind frequently observed reactions, and an overview of the dynamical systems found in the cell cycle. We also include a model of one of the reproduction checkpoints and hypothesize on the effects of modifying certain elements of this model.

2.3 Cell Cycle Biology

In modeling cancer, it is important to first understand the biology of a normal cell. It is not necessary at this stage of research to be able to comprehend the specifics of the numerous internal and external cellular chemical reactions, but rather to be able to understand the overall picture of cell reproduction and its effects on the surrounding cells. This understanding begins with the comprehension of the cell cycle.

2.3.1 A Broad Overview of the Cell Cycle:

The cell cycle is composed of 5 phases: three gap phases–G_0, where a cell remains in a quiescent state, and G_1 and G_2, in which protein synthesis occurs; S phase, in which DNA synthesis occurs; and M phase, where mitosis and cell division occur [32]. The gap phases and S phase are collectively referred to as interphase, which takes up a majority of the time spent in a cell cycle.

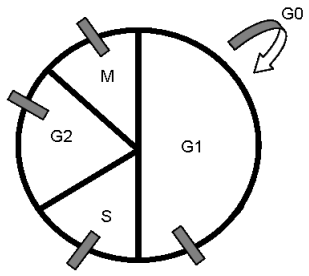


Figure 3.1: The Cell Cycle

Certain cells remain in G_0 throughout their life; others can be drawn out of this phase through external factors, such as the binding of growth factor

molecules to the cellular membrane. Once a cell enters G_1, it is generally committed to one of three paths: DNA synthesis and cytokinesis, differentiation, or apoptosis. G_1 is often referred to as the START phase, during which the cell division and differentiation machinery is switched on. Within this phase, the concentrations of certain chemicals as well as cellular events help to determine the path that the cell will take. Growth of the cell in G_1 is a signal that the cell is preparing for division; once a certain size is reached, the cell moves into the DNA synthesis S phase. In S, the cell employs a series of nuclear events to transcribe and replicate DNA. A checkpoint at the end of this phase ensures that all the DNA is synthesized appropriately and without errors. The G_2 phase heralds the final preparations for cytokinesis with the production of proteins to control the actual division of the cell. A final checkpoint before mitosis occurs at the end of the G_2 phase, confirming that the cell is ready for division. The M phase is where cell division actually occurs.

2.3.2 A Closer Look at the Cell Cycle

Transitions between stages within the cell cycle are controlled by the activation of a complex created by the binding of cyclins and cyclin-dependent kinases (Cdk). Hereafter, this cyclin/Cdk complex will be referred to as cyclin:Cdk. Activation of the complex is achieved through a series of steps, each requiring phosphorylation or dephosphorylation of certain amino acid receptor sites, namely threonine [56]. Although there are currently eight different cyclins and nine different Cdks identified, progression through the cell cycle stages typically relies on only one or two interactions between members of the two sets at each phase or checkpoint [83]. The total concentration of cyclins through the duration of the cell cycle oscillates; each type of cyclin is identified by the stage in which its peak concentration is reached and its association with specific Cdks. Accumulation of cyclin is a rate-limiting step, dictating the passage into subsequent stages. In contrast, the relative concentration of Cdks remains stable throughout the cell cycle [32].

The relationship between cyclins and Cdks acts as a positive regulator of the cell cycle. An example of this regulatory effect is the activation of the cyclin D/Cdk4 complex in the G_1 phase. The complex initiates progression through the stage by phosphorylating other substrates that lead to transcription of DNA synthesis and other reactions necessary for movement through the cell cycle [32]. Along with positive regulators, negative regulators are also

involved in the cell cycle. These cyclin-dependent kinase inhibitors (CDKIs) act by either binding to the cyclin:Cdk, blocking molecular sites for phosphorylation, or by causing an allosteric change in the structure of the Cdk molecule. This change in the tertiary structure of the protein alters the cyclin binding site as well as the ATP binding site, leading to a lower affinity for ATP molecules (see [32]). Other regulatory molecules include *wee1*, Cdk-activating kinase (CAK), and Cdc25, a phosphatase that activates the cyclin/Cdk complex by adding *wee1*. Cdk forms a positive feedback loop with *wee1* and Cdc25, enhancing the activities of each (see [81]).

Despite the difference in the type of cyclins and Cdks involved in the progression through each cell stage, the general method of creation, activation, and degradation of cyclin:Cdk is similar for each cyclin/Cdk pair. The general methodology can be described by a wiring diagram.

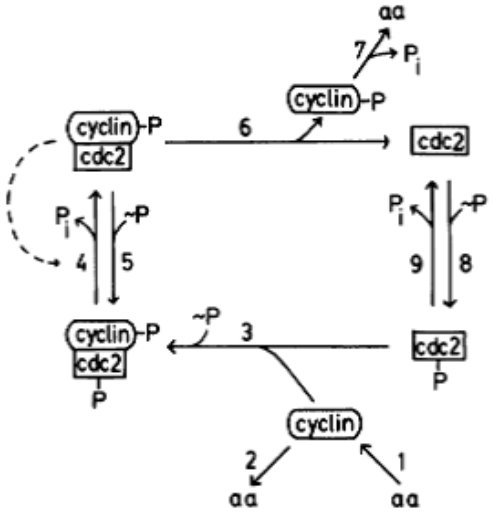


Figure 3.2: Wiring diagram for the cyclin:Cdk complex (Taken from [66])

In this diagram, we can see that there are nine basic steps. The first step is the formation of the cyclin protein from amino acids. The newly-created cyclin can be stable, in which case it becomes available for phosphorylation and subsequent binding to Cdk (step 3; note that in the wiring diagram, cdc2 is a particular type of Cdk), or unstable, in which case it degrades back into amino acids (step 2). After the formation of cyclin:Cdk, the heterodimer is then phosphorylated to activate the complex. Steps 4 and 5 are regulated

by the concentration of CDKIs, which suggest that there is a concentration threshold established by active CDKIs which the active cyclin:Cdk must overcome in order to progress through the checkpoint [61]. After a sufficient amount of cyclin:Cdk is activated, the cell cycle progresses and active cyclin:Cdk is rapidly destroyed (step 6). The cyclin is further degraded into amino acids and its phosphorus molecules taken up by ADPs (step 7). The Cdk subunit remains inactive within the cell until external triggers signal for phosphorylation of the kinase (step 8). This actives the Cdk, which is then ready for acceptance of the cyclin subunit, and the checkpoint cycle begins again.

Active cyclin:Cdks motivate progression through the checkpoint by enhancing the phosphorylation of a number of specific substrates, including proteins, lamins, and histones. Harper [32] suggests that substrate specificity is determined by the type of cyclin bound to the Cdk (in the case where more than one cyclin type is accepted). This allows for contrasting effects on a common substrate, which can act as a safety mechanism to prevent premature progression through the cell cycle.

2.3.3 Cell Cycle Checkpoints

Within the cell cycle, numerous checkpoints are built in to ensure normal progression between phases. The four main checkpoints occur between stages G_1/S and G_2/M, as well as within the S and M stages. The G_1/S checkpoint monitors cell size and checks for damage within the DNA of a cell. If these conditions are not adequately met, the cell cycle arrests until the proper modifications have been made. It is also possible to trigger transition into the G_0 phase if environmental conditions, such as restrictions on available space, prevent checkpoint requirements from being met. The G_1/S checkpoint is also where the decision for apoptosis and differentiation is made. If the G_1/S conditions are met, the cell proceeds to the S phase and begins synthesis of DNA. The checkpoint within the S stage evaluates the accuracy of DNA replication. The next checkpoint, G_2/M, monitors the physiological conditions of the cell. A further check of DNA replication is performed; if there is incompleteness or damage of the DNA, the cell cycle is arrested until corrections are made. The fourth checkpoint occurring within mitosis between metaphase and anaphase, controls for incomplete spindle formation. This checkpoint also serves as the last place to stop damaged or defective cells from reproducing.

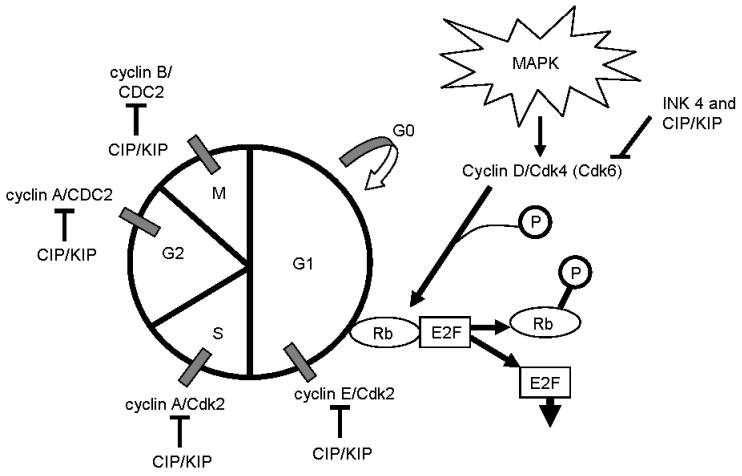


Figure 3.3: Specific cyclin:Cdk pairings and inhibitors for each checkpoint. (Taken from [32])

Each checkpoint in the cell cycle is associated with unique couplings of cyclins and Cdks (Figure 3.3) [32]. The behavior of these complexes controls successful movement through the cell cycle. Over-expression of the cyclin E/Cdk2 complex, as an example, promotes passage through the G_1/S checkpoint. Once the checkpoint is passed, the complex rapidly degrades. Both the disassociated cyclin and Cdk are present throughout the next stage, although the former is continually being broken down into its amino acid components.

2.3.4 External Motivators of the Cell Cycle

Progression through the cell cycle can be motivated by internal factors, such as the activation and degradation of cyclin/Cdk complexes, as well as through external factors. Signal cascades, which help to trigger the internal cellular reproductive pathway, can be initiated by a variety of molecules and ions. The primary stages affected by external factors are G_0 and G_1. Mitogens, such as growth factors, and the mitogen-activated protein kinase (MAPK) cascade are the main contributors for signaling the cell to progress from G_0 into G_1 (see [32]). Within the G_1 phase, it is suggested by Crespo [17] that the Ras protein, whose creation is signaled by the MAPK cascade, helps to drive progression into the S phase and to the mitotic machinery within the cell.

Ions are also important factors in the progression of cellular activities. Baran [5] has identified calcium as a potential effector for transition through G_1 and G_2. The Ca^{2+} ion can either be extracted from extracellular sources bound to the cell membrane, or absorbed internally from mitochondrial releases.

2.4 The Brusselator

The brusselator is an autocatalytic reaction studied in Brussels by Prigogine and a group of other scientists [29]. Autocatalytic reactions are an example of nonlinearities in chemical reactions. The brusselator involves the following series of reaction steps:

$$\begin{aligned} A &\rightarrow X \\ 2X + Y &\rightarrow 3X \\ B + X &\rightarrow Y + D \\ X &\rightarrow E. \end{aligned}$$

The concentration of X and Y may change in time. However, this reaction can be controlled from reaching equilibrium by continually adding the substances A and B and subtracting the substances D and E. We do not want this reaction to come to a halt because it is used to represent real-world systems. Prigogine argues that almost any real-world system is open, characterized by nonlinearities, and maintained out of equilibrium with their surroundings. An example is an individual's consumption of material and energy inputs from their surroundings and their excretion of waste products and waste heat [29].

In this experiment, the kinetic constants have been set to one for simplicity and D has been eliminated because it does not enter any of the reaction and is continuously removed from the system. After this elimination, the following nonlinear equations result:

$$\begin{aligned} \frac{dX}{dt} &= A + X^2Y - BX - X \\ \frac{dY}{dt} &= BX - X^2Y. \end{aligned}$$

The two conditions necessary for a limit cycle to occur is that the system must be open and interactions among system components must be nonlinear.

This system then will ultimately converge to a steady limit cycle. It is helpful to note that the brusselator model has already been programmed into Matlab under "brussode". The Matlab code can be found in Appendix B.

2.4.1 Chemistry of the Brusselator

An autocatalytic reaction is a chemical reaction in which a product (or a reaction intermediate) also functions as a catalyst. In such a reaction the observed rate of reaction is often found to increase with time from its initial value. To further understand what this means, let's look at the second reaction step in the brusselator:

$$2X + Y \rightarrow 3X.$$

This reaction says that substances X and Y are reacting to form a larger amount of the substance X. We say the reactants X and Y reacted to form the product, in this case, X. The concentration of chemicals is measured in moles/grams. The molar ratio can be found by taking note of the 2 in front of the X and the 1 in front of the Y in the reactant side, and the 3 on the product side. So we say that the molar ratio between the reacting X and Y is 2 to 1. The molar ratio between the reactant Y and the product X is 1 to 3.

Note that X and Y reacted to form more of the X substance. This is an example of autocatalytic reaction where X stimulates its own production from Y.

Let's observe and understand how the differential equations in our system were obtained. Again, the complete chemical reaction and the equations are:

$$\begin{aligned} A &\rightarrow X \\ 2X + Y &\rightarrow 3X \\ B + X &\rightarrow Y + D \\ X &\rightarrow E. \end{aligned}$$

Let's look at the first equation $\frac{dX}{dt} = A + X^2Y - BX - X$. The equation represents the rate of change of X with respect to time. In this equation, the term A comes from $A \rightarrow X$ and X^2Y comes from $2X + Y \rightarrow 3X$; since both equations form the product X, they have positive values in the equation. Note that the equation only represents the concentrates that formed X. We

can also represent the change in concentration X with respect to time using the products formed. More details on how to solve for the rates can be found in the Rate Law section.

The same concept applies to the second equation $\frac{dY}{dt} = BX - X^2Y$. BX is positive since it forms the Y substance ($B + X \rightarrow Y + D$); while X^2Y is negative indicating the loss of the substance Y and the making of X ($2X + Y \rightarrow 3X$).

Rate Law

The rate of a reaction is often proportional to the concentration of the reactants. It is the amount of substance reacted or produced per unit time. The rate law is an experimentally determined equation of a reaction. It is an expression indicating how the rate depends on the concentrations of the reactants and catalysts for the reaction [12]. The power of the concentration in the rate law expression is called the order with respect to the reactant or catalyst. Consider the following reaction

$$2A + B \rightarrow C.$$

Let $r =$ the rate and $k =$ the kinetic constant. This constant is dependent on the temperature at which the reaction occurs, but independent of the concentration of each variable.

$$r = k[A]^2 * [B].$$

*Note that the notation $[X]$ indicates the concentration of X.

The rate of consumption of the equation is:

$$r = -\frac{1}{2}\frac{d[A]}{dt} = -\frac{d[B]}{dt}.$$

The rate of formation is:

$$r = \frac{d[C]}{dt}.$$

The rate of consumption is negative because the concentration of A and B substances is decreasing, while the concentration of substance C is being formed and therefore is increasing. Since all substances are being formed and consumed at the same rate, the rate of the reactions is:

$$r = -\frac{1}{2}\frac{d[A]}{dt} = -\frac{d[B]}{dt} = \frac{d[C]}{dt}.$$

In summary, the rate r can be represented in all of the following forms:

$$r = -\frac{1}{2}\frac{d[A]}{dt} = -\frac{d[B]}{dt} = \frac{d[C]}{dt} = k[A]^2 * [B].$$

2.5 Dynamical systems

The study of dynamical systems requires knowledge of bifurcations. In dynamical systems, a bifurcation happens when a small change applied to the parameter values of a system eventually leads to a long-term change in the system's dynamical behavior. The dynamical system concept is a mathematical formalization for any fixed "rule" which describes the time dependence of a point's position in its ambient space. The ambient space is defined as the space surrounding our system of equations. The mathematical models used to describe the swinging of a clock pendulum, the flow of water in a pipe, or the number of fish each spring in a lake are examples of dynamical systems [76]. The following section will go into details about linear and nonlinear dynamical systems and how different bifurcations can occur in this system.

In order to understand what is meant by the limit cycle in the brusselator reaction, we must visit the dynamics behind it and explain some basics. Understanding linear systems will help the reader understand how to classify the fixed points (nodes) in a nonlinear system. This is important since limit cycles occur in nonlinear systems. A linear systems basically states that equal causes have equal affects. Consider the example if you ran out of gas and had to step out and push the car. If you start doubled the pushing effort and in return the car moved twice as fast, then that would be an example of a linear system. In a nonlinear system, if doubled the pushing effort the car would not move twice as fast. Sometimes it is safe to assume that the car is moving twice as fast in the non-linear system even though it is only approximately (but not exactly) moving twice as fast. This process is called linearizing a nonlinear system. Mathematicians do this because solving a linear system is much easier than solving a non-linear system which is generally difficult or impossible to solve. In this case, the approximation is safe enough to make this assumption.

Let's consider the following 2 dimensional linear system of the form

$$\dot{x} = ax + by$$
$$\dot{y} = cx + dy,$$

where a, b, c, and d are parameters and $\dot{x}$ and $\dot{y}$ represent the derivatives of the variables x and y. Note that this system is of the matrix form $\dot{x} = Ax$ where

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \; x = \begin{bmatrix} x \\ y \end{bmatrix}.$$

This system is linear; therefore if x_1 and x_2 are solutions then so is any linear combination $c_1x_1 + c_2x_2$.

The solutions of $\dot{x} = Ax$ can be visualized in a phase portrait. A phase portrait is a geometric representation of the trajectories moving on the (x, y) phase plane [64].

As an example of a linearized system, consider the following problem:

$$\dot{x} = x^2 - 1$$

We will need to graph the equation (see Figure 5.4), classify its fixed point and their stabilities.

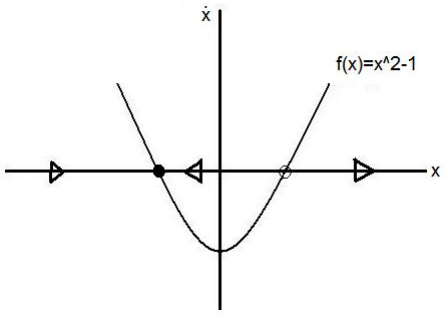


Figure 5.4: Phase portrait

Since we are interested in seeing how $\dot{x}$ is behaving. We will do this by treating $\dot{x}$ as a function and then graphing $f(x) = x^2 - 1$ to see how its trajectories are moving. Trajectories are normally represented by little arrows in a system of differential equations' graph. Trajectories can be thought of as little particles in space moving toward where the attraction is. The points that these trajectories point to are referred to as attractors or sinks and are represented by a bold circle on the graph. The points on the graph that the trajectories point away from are called repellers and are represented by an empty circle on the graph.

By solving the equation $f(0) = x^2 - 1$, we get$x = \pm 1$. The value of $x^2 - 1$ is positive if we pick any point less than -1 so the flow is to the right toward this fixed point. The value of $x^2 - 1$ is negative if we pick any point where $-1 < x < 1$. The flow is again to the right where $x > 1$. As the graph can easily demonstrate, the flow is toward the stable fixed point $x = -1$ and away from the unstable fixed point $x = 1$.

2.5.1 Limit Cycles

Linear systems can have closed orbits, but these orbits are not isolated as they are in the limit cycles in nonlinear systems. Orbit can occur in linear systems as well. You can actually have one orbit inside the other; an analogy is that the limit cycles are represented by the inside and outside rings of a donut. In this case the trajectories will point toward the stable orbit and away from the unstable one. There are cases where both orbits would be stable; in this case, they share their trajectories. Limit cycles cannot form within each other. They are defined as isolated closed trajectories. They are said to be isolated because their neighboring trajectories are not closed; they spiral either toward or away from the limit cycle. The limit cycle is said to be closed because the trajectories inside it can never leave it to go anywhere else. The limit cycle is said to be stable if the trajectories spiral toward it and is denoted by a solid black lining, and unstable if they spiral away from it and is denoted by a dotted black lining.

Let's consider a simple example of a limit cycle:

$$\dot{r} = r(1 - r^2), \theta = 1,$$

where $r \geq 0$ (Figure 5.5). The radial and angular dynamics are uncoupled and so can be analyzed separately. Treating $\dot{r} = r(1 - r^2)$ as a vector field on the line, we see that the solution $r^* = 0$ is an unstable fixed point and $r^* = 1$ is stable.

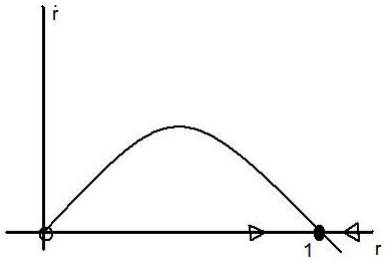


Figure 5.5: Radial and angular dynamics graph for $\dot{r} = r(1 - r^2), \theta = 1$

Hence, back in the phase plane (Figure 5.6), all trajectories (except $r^* = 0$) approach the unit circle $r^* = 1$ monotonically. Since the motion in the θ-direction is simply rotation at constant angular velocity, we see that all trajectories spiral asymptotically toward a limit cycle at $r = 1$ [64].

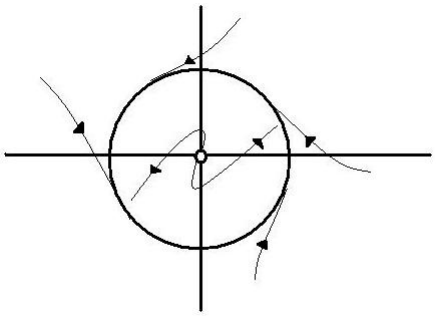


Figure 5.6: Phase plane image of a limit cycle

2.5.2 Bifurcation

Hopf and saddle-node bifurcation have come up in several papers that we have attempted to model. This section will explain some of the different kinds of bifurcations and give an example of a saddle-node bifurcation.

Bifurcation is the qualitative or topological change of the fixed point of a dynamic system as the result of a varying parameter [34]. Bifurcation occurs in both continuous systems and discrete systems. There are two classes of bifurcations: local and global. A local bifurcation occurs when a parameter change causes the stability of an equilibrium (or fixed point) to change [64]. There are four different kinds of local bifurcations: transcritical, pitchfork, hopf, and saddle-node bifurcation.

In general, the nullclines (where $\dot{x} = 0$ and $\dot{y} = 0$) would have to be defined to know where the stationary points are located. The direction of the field is to the right where $\dot{x} > 0$ and to the left where $\dot{x} < 0$. The attraction is upward when $\dot{y} > 0$ and downward when $\dot{y} < 0$. Once the nullcline is sketched, the stationary points are classified and it becomes clear which point is stable and which one is unstable depending on the direction of the flow.

Transcritical Bifurcation

A transcritical bifurcation is a particular kind of local bifurcation, meaning that it is characterized by an equilibrium having an eigenvalue whose real part passes through zero [76]. There always exists a stable point and an unstable point in transcritical bifurcation. As the parameter changes, the fixed points

collide and exchange stability. An example of transcritical bifurcation is the logistic equation in which the parameter and the x value are both positive [64]. A typical real-life example could be the consumer-producer problem where the consumption is proportional to the (quantity of) resource [76].

Pitchfork Bifurcation

Pitchfork bifurcations occur generally in systems with symmetry [64]. There are two different types of bifurcation: supercritical and subcritical. As the parameter of our differential equation is varied, the fixed points tend to appear and disappear in this bifurcation. To understand this better, think of the case when we only have one equation and the fixed points are found along the x axis where $\dot{x} = 0$. As a simple pitchfork bifurcation example, consider a system that has 1 stable limit cycle when the parameter is less than zero. Suppose that we get 3 limit cycles when we increase it to be greater than zero, then we say that the other 2 fixed points appear when our parameter is greater than 0 and disappears when the parameter is less than zero.

Hopf Bifurcation

To understand hopf bifurcation in the simplest qualitative way, we consider a physical system that settles down to equilibrium through exponentially damped oscillations (an oscillation decreasing in amplitude). Small disturbances decay after time in this system. Now suppose that the decay rate depends on a control parameter β. If the decay becomes slower and slower and finally changes to growth at a critical value β_c, the equilibrium state will lose stability . In many cases the resulting motion is a small-amplitude, sinusoidal limit cycle oscillation about the former steady state. Here we say the system has undergone a hopf bifurcation.

Saddle-node Bifurcation

Saddle-node bifurcation is the basic mechanism by which fixed points are created and destroyed. As a parameter is varied, two fixed points move toward each other, collide, and mutually annihilate [64]. If the phase space is one-dimensional, one of the fixed points is unstable (the saddle), while the other is stable (the node) [76].

Consider the following problem:

$\dot{x} = \varepsilon + x^2$, where ε is a parameter and $\dot{y} = -y$.

Note that when we solve the differential equation $\dot{y}$, $y(t)$ goes to zero and disappears as t goes to ∞.

$$\frac{dy}{dt} = -dty = ye^{-t}.$$

The parameter of the parabola in x changes as ε varies. The function $\dot{x} = 0$ is shown with different values of ε.

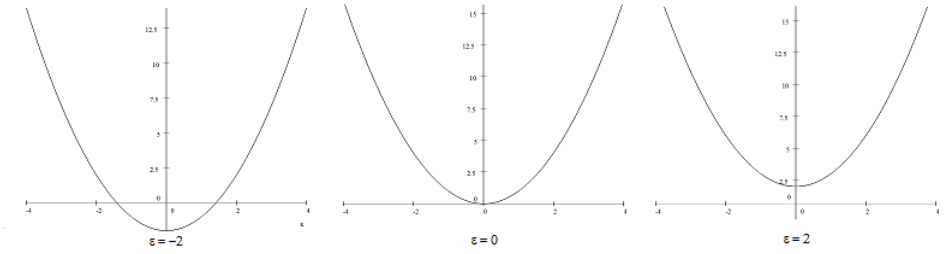


Figure 5.7: Varying levels of ε

The first fixed point is stable at $(-\varepsilon^{1/2}, 0)$ and unstable at $(\varepsilon^{1/2}, 0)$. As the parameter increases from $\varepsilon = -2$ to $\varepsilon = 2$, the two fixed points approach each other and form a saddle at the origin. If you imagine a moving graph starting from the negative parameter to the point 0 where our object are on our figure at all time, then you can see that it seems that these 2 fixed points were moving toward each other until they finally met at point 0. We say that they formed a saddle point at $x = 0$. The saddle point is very delicate; as soon as $\varepsilon > 0$, the point disappears and we're left with no fixed points. This is an example of saddle-node bifurcation where the fixed points are destroyed when the parameter is greater than 0 and created when the parameter is 0 and as the parameter is decreased, the 2 fixed points are apparent.

2.6 Modeling

In a effort to better understand the way cell cycles are controlled by the interactions between cyclins and Cdks, we have attempted to replicate models found in a variety of different papers. We have successfully reproduced the model presented in [66] using Matlab; the coding for this model can be found in Appendix B.

2.6.1 Modeling of the Metaphase/Anaphase Checkpoint

The transition between the metaphase and anaphase stages in the M phase serves as the last major checkpoint in the cell cycle before cytokinesis finalizes. During this checkpoint, the positions of chromosomes with respect to the cellular axis, as well as proper spindle formation, are evaluated to ensure proper separation during anaphase. If errors are detected, the division mechanism will stall until either the error is corrected or the cell degenerates. Cellular determination of spindle alignment is dependent on the activity of an enzyme complex, maturation promoting factor (MPF), composed of cyclin B and Cdk1 [66]. This heterodimer is influential in driving the cell into and through the M phase, peaking abruptly at metaphase. At the transition between metaphase and anaphase, barring any complications, MPF disassociates and the cyclin B is rapidly degraded. This action leads to the triggering of a signal cascade that allows the cell to finish cytokinesis. Tyson [66] explores the dynamic nature of the synthesis and breakdown of MPF and its effects on cell cycle progression.

2.6.2 Description of Model

Our model, based off a system of ordinary differential equations presented in Tyson [66], focuses on two parameters governing the formation and degradation of MPF: k_4, describing the rate of auto-catalytic activation of MPF, and k_6, describing the rate of breakdown of active MPF. We also explored the effects of changing parameters from constant to dynamical in order to observe the effects of a time-dependent chemical reaction; time-dependent reactions are potentially more biologically accurate. To do this, we changed the rate parameter k_1, which governs the synthesis of Cdk subunits from amino acid building blocks. We chose to use a sinusoidal function for this parameter to model the fluctuations that many organisms experience. Results for these changes are discussed later in the paper.

For the purposes of our model, we assumed that k_4 remained unchanged throughout our testing of different parameter values for k_6. Through experimentation, we found that this assumption does not significantly alter our results. Biologically, when the value of k_6 falls between critical value bounds (roughly determined as between .12 and .34), active MPF induces a massive dephosphorylation of inactive stores of MPF. This increase in MPF activity is what presumably drives the cell into mitosis [66]. This burst of activity is

followed by disassociation of the MPF enzyme and degradation of the cyclin component. See (Figure 3.2) for a wiring diagram of this sequence.

The chemical reaction, as shown in Figure 3.2, is a modified version of the Brusselator. Our model uses a pair of ordinary differential equations to describe the concentrations of the subunits of the checkpoint reactions. We are most interested in the concentration of active MPF versus the concentration of phosphorylated cyclins and MPF. This relationship can be described as follows:

$$\begin{aligned} \frac{du}{dt} &= k_4(v-u)(\alpha+u^2) - k_6 u \\ \frac{dv}{dt} &= (k_1) - k_6 u, \end{aligned}$$

where u is the ratio of the concentrations of MPF and total cyclins; v is the ratio of the concentrations of phosphorylated cyclins and MPF versus total cyclin; α is the rate of change of k_4 over the rate k_4; and k_1 is a constant that assumes Cdk1 is continuously synthesized from amino acids to maintain a constant concentration of Cdk1 subunits.

2.6.3 Analysis of Model

Utilizing the ode45 function in Matlab, we were able to successfully replicate the graphs found in [66]. This model makes a number of assumptions regarding the behavior of cyclin and Cdks within the cell: this portion of our model does not take into account a variable rate of synthesis of Cdk subunits (given by the parameter $k_1 * [aa] / [CT]$), newly synthesized cyclin is stable, and inhibition of MPF by re-phosphorylating the subunit (step 5 in Figure 3.2) is disregarded. However, our model still accurately portrays the behavior between cyclin B and Cdk1 as observed in laboratory experiments [66]. Newer papers attempt to model the variable rate of Cdk synthesis by adding a variable division rate into the ODEs [56]. We allow for our first assumption due to the general overall concentration of amino acids within the cell. As mentioned earlier, we modified our model to account for a changing rate in the availability of amino acids necessary for the creation of Cdk subunits. The reasoning behind the second two assumptions is that the rate of activation of the complex should be much higher than the rate of deactivation of the complex and its subunits in order for the signal cascade to have

a probability of being activated. By leaving out this much smaller rate of deactivation, the model is not significantly impacted.

We are interested in the effects of changing the rate of activation and degradation of the MPF complex. With lower values of k_4 and k_6 than given as parameters in [66], the model yields more interesting results. We have found that the value of k_4 serves as a stretching factor, and does not affect the appearance of the limit cycle; for the graphs shown, we allow $k_4 = 5$. When the rate of breakdown of MPF is low ($k_6 = .075$), the system goes to a stable point (see Figure 6.8). This can also be seen in the graph of the dampening oscillations of the concentrations of active MPF and total cyclin versus time.

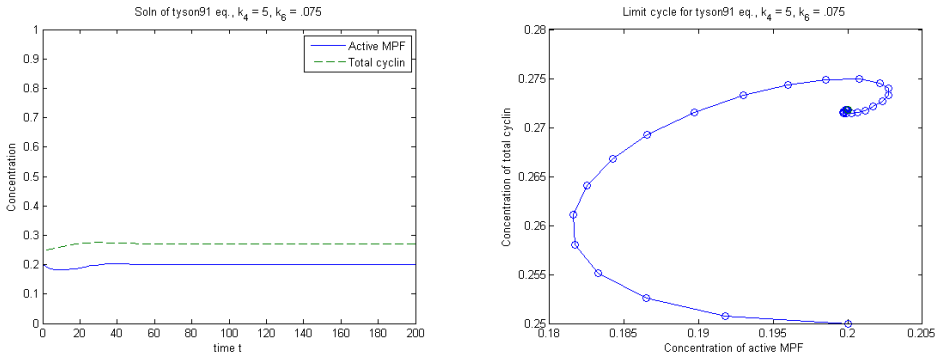


Figure 6.8: Concentrations of active MPF and total cyclin: $k_4 = 6, k_6 = .075$

For slightly higher values of k_6 ($k_6 = .2$), we can see that the concentrations of MPF and cyclin in the system reach a limit cycle (Figure 6.9).

This steady oscillating behavior is reminiscent of laboratory observations of rapid cell cycling in early embryos [66]. The limit cycle is stable, and does not show any chaotic behavior in the simulations we have run. The period of the function, roughly 75 minutes long, corresponds to the transition through G_2 up to metaphase. Notice that as the concentration of total cyclin begins to drop, MPF becomes rapidly activated. This triggers a breach of threshold and promotes movement through the cell cycle. This model shows a continual recycle of cyclin B and Cdk; it does not assume movement into the next subphase, but rather restarting at the beginning of M.

If the rate of breakdown of MPF is increased slightly higher ($k_6 = .4$), the limit cycles disappear (Figure 6.10) and a stable point is again reached.

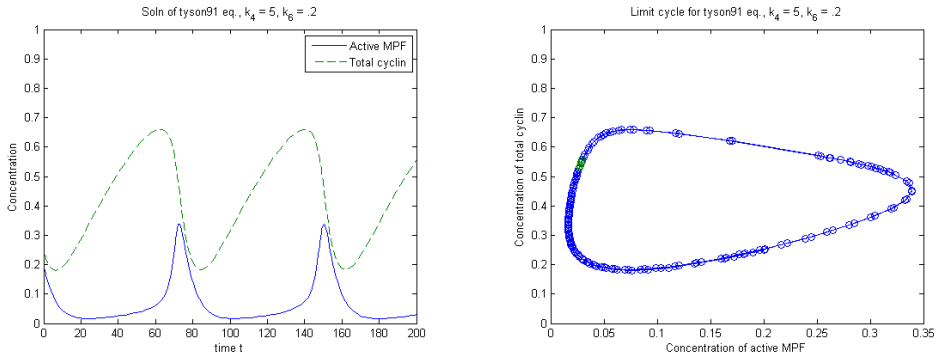


Figure 6.9: Concentrations of active MPF and total cyclin: $k_4 = 6, k_6 = .2$

We can see that the amount of phosphorylated cyclins and MPF approaches the total amount of cyclins in the system. The consequence of this high concentration is the decreased availability of cyclins for binding to Cdk1, which results in fewer MPF complexes created.

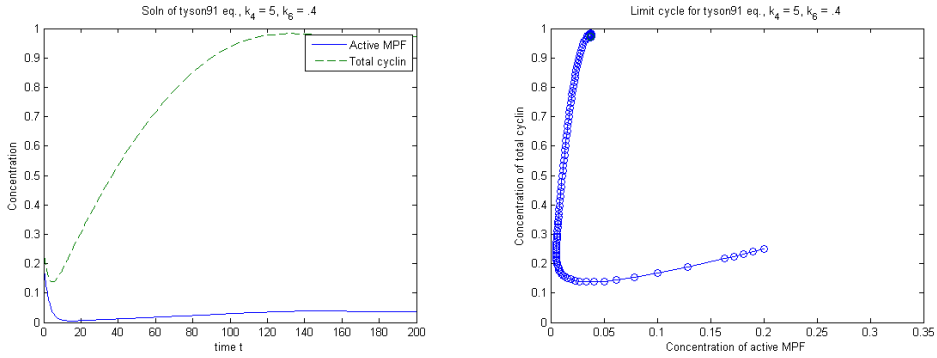


Figure 6.10: Concentrations of active MPF and total cyclin: $k_4 = 6, k_6 = .4$

2.6.4 Future applications of checkpoint modeling

Each checkpoint is controlled by a different cyclin/Cdk complex. It may be possible to couple each checkpoint model in the future to produce an overall picture of cellular reproduction; however, our group was unsuccessful in achieving this model.

An understanding of reproduction and growth at the cellular level is beneficial for understanding actions of cells at the tissue level. We have noted that extracellular influences can affect cell cycle progression from G_0 to G_1 and that the cell shows different behaviors depending on rates of auto-catalytic and degradation activity.

2.6.5 Chaos

As we performed preliminary research, we noted that there were numerous papers investigating chaos. This led us to investigate chaos within our own model. Chaos can be defined as having aperiodic long-term behavior in deterministic systems that exhibit sensitive dependence on initial conditions [64].

Aperiodic long-term behavior means there are trajectories that do not settle down to a fixed points, or a limit cycle. As can be seen on the graph, none of the orbits are attracted to just one point or cycle, they just spiral around in an irregular way.

By deterministic we mean that the system has no random parameters. The irregular behavior arises from the system's nonlinearity. The effects of the nonlinearities tells us that the effects are no longer proportional to causes.

The nearby trajectories separate exponentially fast in a system that exhibits sensitive dependence on initial conditions. This means by minutely altering the parameters in our model, a dramatic change of behavior is observed.

By changing parameters in our model, we can see that chaotic behavior has affected the attractor of our model, leading to orbits that converged to a chaotic region. An attractor is a closed set to which all neighboring trajectories converge. In our model, this attractor was the stable limit cycle [64]. The conversions occurred when we forced the variable representing the rate of synthesis into a time dependent sinusoidal function. After manipulating the constants, a chaotic system was observed. The following graph represents chaos in our system.

As can be observed in Figure (6.11), there seems to be some type of controlled parameter since the graph shows cycles that are going all over the place but within a certain area. By changing k_6 from 0.2 to 0.6, our initial limit cycle converged into a chaotic system. The system's behavior that resulted after the small perturbation in the value of the parameter indicated that the system is sensitive to initial conditions. It is also easy to see that the

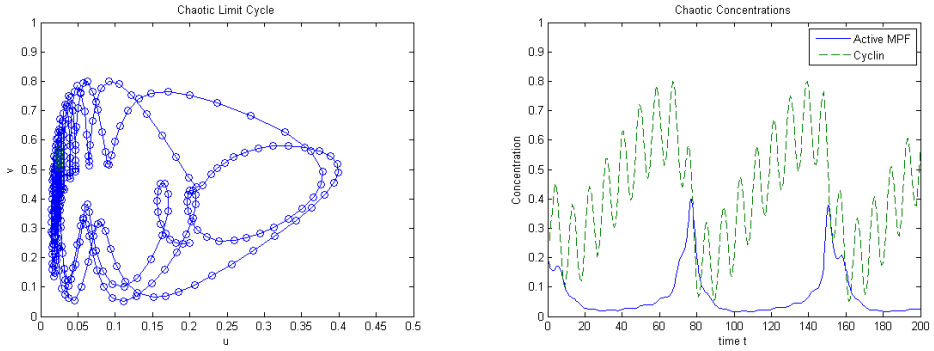


Figure 6.11: Chaos in our model

graph shows an aperiodic long term behavior as it never settles to a stable point or a stable limit cycle. The right-hand side of Figure (6.11) describes the aperiodic evolution of MPF and the total concentration of cyclins in the system with respect to time.

In order to validate our observations, we performed a further literature search to determine if there were other papers investigating chaos in biology. Some of the other research papers we found modeled chaos by forcing the system to be dependent on time (similar to our model). These types of models are said to be non-autonomous. The paper that we chose to summarize was the first of its kind [60]. This is due to the fact that it modeled chaos without coupling the rate of synthesis and time. The paper modeled the periodic oscillations of cdk1 and cdk2 in the M and S phase in the cell cycle. The paper focused on the coupling between two enzymatic cascades controlling the periodic activation of cdk1 and cdk2 by studying the change in the inhibition rates of the two. They measured the strength of mutual inhibition of the two oscillators. The increase of the synthesis of cdk1 and the inhibition of cdk2 starts the M phase. While the inhibition of cdk1 and the increased synthesis in cdk2 starts the S phase. The paper goes on to show how 2 limit cycles were formed. When the parameters changed to represent a system where the inhibition of cdk1 and cdk2 was mutually increased or decreased, chaos was observed.

The Cause of Chaos in Biology

By manipulating the constant in the models chaos was observed. The constants were not determined experimentally but were rather arbitrary. They were chosen specifically to form chaos. It becomes questionable if it physiologically possible that chaos can occur within the cells cycle? Is it possible for the inhibition rate to ever be high enough to cause chaos? Research shows that extremely high rates or extremely low levels of cdk within the cell can occur due to errors in the cell signaling process. genetic errors can cause a cellular adapter to go missing and can cause disruption in signal transduction [73]. Signal are transported into the cell by proteins located on the surface of the cell. These signals control cell functioning. The process in which these signals are transferred is referred to as signal transduction. One can think of signal transduction as a domino effect: one signal is sent out and whole bunch of "things" are affected along the way as the signal continues its travel within the cell. If the signal transduction is disrupted, the signal will not carry the correct information. The cell has cellular adapters all over just to make sure that the signals are good. To simplify this idea, we can think of the signal transductions as codes, and the cellular adapters as programmers that are located all over the cell to guard it from errors. As the codes travel along the cell they have to go through the programmers before they are allowed to make any changes in the system. If there is an error in the code, the programmer will then recode it before allowing it to pass on to the next stage.

The existence of the programmers indicates the possibility of bad signal disruptions. According to research done in Hanover Medical School [73], there is a gene that was found to be responsible for a cellular adapter protein. The cellular adapter may go missing if this gene is mutated for example. the combination of a missing cellular adapter and a disrupted signal transduction can lead to different disruptions within the cell. depending on the type of the signal, this disruption could be the cause of the inhibition of the cdks.

The body of course does not only depend on the cellular adapters to fix errors. It in fact takes many steps to stay healthy and many others to stop the cell cycle in cases where the error(s) cannot be fixed. We can think of these steps as cell "guards" that make sure everything is working properly and take care of business when something looks out of the ordinary. One of these guards that we found of significance is $p53$. $p53$ is a protein that is responsible for stimulating another protein called $p21$. $p21$ runs an error

check on the DNA string and when defected, it reduces the synthesis of cdk and stops the cell signal. In the absence of well-functioning $p53$ (as in the case where it is only passed on by one parent) $p21$ is not stimulated and the cell cycle is not stopped. The bad DNA is then passed on to the daughter cells as the cell cycles of the bad cells continue, uncontrollable division is observed within the cell population.

According to biologist Zinat Ismael [39], cdk was only discovered within the last few years. It is currently a hot research topic as its reduction or excess has been observed in cancer cells. this matches our result in both of our models. in both models the variation of the inhibition synthesis constant of cdk formed chaos within certain parameters. It is true that we only stumbled through chaos in our model, but it is amazing to find that not only was the manipulation worthy of investigating in mathematics, but biologists are strongly considering the same scenario.

As we can see, chaos in the cell cycle does not directly affect the causation of the growth of bad cells. When there is something wrong with the cell signaling, the inhibition or the synthesis of cdk can get a signal to either increase or decrease causing chaos within the cell cycle. The body then can go through multiple steps to fix the chaos; when it cannot, it has several other steps it can take to stop the cell cycle and send the cell to apoptosis. When all fails, the mother cell would divide, passing bad DNA to daughter cells. the bad DNA can cause the cells to differentiate uncontrollably. As the defective cells divide, the population of the surrounding cells will become infused with defective cells. this is a key step in tumor formation.

It is still not clear if the chaos modeled by forcing the system into a periodic oscillator is of physiological importance; or if it's the one where chaos occurred autonomously. Perhaps it is a combination of both. One thing for sure, chaos can occur in the cell cycle and it seems that researchers are focusing on the relationship of the change in the rate of inhibition of cdks and the formation of chaos. Many researchers are still working on proving or disproving the fact that chaos also occurs in the health cell cycle [60].

2.7 Future explorations

The micro-model of the cellular reproduction cycle is only as good as its applications to the macro-model of the tissue environment in which it lives. We know that different types of cyclins and Cdks are involved at each check-

point, limiting the amount of overlap in the differential equations we can use to model the concentrations of each. We also know through manipulations of the model set forth by [56],[66] that the rate constants of the activation and degradation of cyclin:Cdk is a major factor in the rate of passage through the checkpoint. Our model currently holds these as "constants" rather than accounting for their variable nature. However, there is limited information on the rate of change of activation and degradation of kinases. It is our goal in future models to incorporate the rate of activation and degradation as functions of an arbitrary molecule that fluctuates with time and external signaling; our hope is that by doing so, we will be able to generate a more biologically accurate picture of the cell's reproduction cycle.

Based on our understanding of the cyclin B/Cdk1 complex, we feel that good molecular candidates for fluctuation are the amino acid threonine and tyrosine. We have previously mentioned that threonine receptor sites are essential to the activation of cyclin:Cdk [56]; tyrosine receptors are also important for activation of Cdk1 (see [20]). If a cell is starved of these amino acids, threonine in particular, the level of available Cdk units will be lowered.

Chaos played a big impact in our research. It helped us understand a few possibilities of how a healthy cell can get damaged and how it takes many steps to stop the damage. It helped us find a path that demonstrated how a cell damage can lead to tumors. It is concluded that disruptions in cell cycle and cell differentiation (which can occur due to chaos or they can cause chaos) play a decisive role in cancer formation. It would be important then, that the following groups do further research on chaos because we believe it will help them with the process of trying to model cancer cells.

2.8 Summary

In this paper, we have attempted to provide a refresher on cellular reproduction and dynamical systems. An understanding of these subjects is important in the comprehension of current research on cell cycle modeling. Our model contains many simplifications; with further research, we feel that we will be able transform some of the assumptions into equations that adequately model observed situations.

2.9 Appendix

2.9.1 Definitions

Anaphase: The stage of mitosis and meiosis in which the chromosomes move from the equatorial plate toward opposite ends of the nuclear spindle.

Apoptosis: A natural process of self-destruction in certain cells that is determined by the genes and can be initiated by a stimulus or by removal of a repressor agent. This is also referred to as programmed cell death in some texts.

Autonomous: Not controlled by others or by outside forces; A reaction in which a compound stimulates its own making.

Cyclins: A class of proteins that fluctuate in concentration at specific points during the cell cycle and that regulate the cycle by binding to a kinase.

DNA: Deoxyribonucleic acid. A nucleic acid that consists of two long chains of nucleotides twisted together into a double helix and joined by hydrogen bonds between complementary bases adenine and thymine or cytosine and guanine. DNA carries the cell's genetic information and hereditary characteristics via the nucleotide sequence and is capable of initiating self-replication and RNA synthesis.

Gap 0 (G_0): A phase in which a cell has exited the active portion of the cell cycle and has ceased to divide. This may either be a temporary resting period or more permanent. Cells that have reached an end stage of development (e.g. neuron) no longer divide and remain in G_0.

Gap 1 (G_1): A cell cycle phase at the beginning of the cell cycle during which cellular growth and preparation for division occurs.

Gap 2 (G_2): A cell cycle phase between S phase and M phase in which protein synthesis and further preparation for cell division occurs.

Enzyme: A highly specific protein that serves as a biochemical catalyst to cellular reactions.

Interphase: A term that refers to the sequence of events occurring in G_0, G_1, S, and G_2. Interphase generally lasts at least 12 to 24 hours in mammalian tissue. During this period, the cell is synthesizing DNA and RNA, producing protein and growing in size.

Kinase: An enzyme that catalyzes the conversion of a pro-enzyme to an active enzyme. This can occur through the transfer of a phosphate group from a donor, such as ADP or ATP, to an acceptor site on the pro-enzyme.

Metaphase: The stage within mitosis and meiosis, following prophase and preceding anaphase, during which the chromosomes are aligned along the metaphase plate.

Mitosis (M): Cell growth and protein production stop at this stage in the cell cycle and cytokinesis (cell division) occurs. Mitosis is much shorter than interphase, lasting perhaps only one to two hours.

Non-Autonomous: Depends on it own making. In our chemical reactions x goes to y is a nonautonomous chemical reaction since y depends on x for its making.

Phosphorylation: The method by which a phosphate group is added to an organic molecule.

Protein: A complex composed of chains of amino acids. The generic term protein can be broken up into many classes, including but not restricted to: enzymes, hormones, kinases, and antibodies. Protein-protein interactions as well as protein-chemical reactions serve to power the cell.

Synthesis (S): A cell cycle phase where DNA synthesis occurs.

2.9.2 Matlab Coding

The Brusselator `function brussode(N)`

```
% BRUSSODE  Stiff problem modeling a chemical reaction

if nargin < 1
  N = 20;
```

```
end
tspan = [0; 10];
y0 = [1+sin((2*pi/(N+1))*(1:N));
repmat(3,1,N)];
options = odeset('Vectorized','on',...
                 'JPattern',jpattern(N));
[t,y] = ode15s(@f,tspan,y0,options);
u = y(:,1:2:end);
x = (1:N)/(N+1);
surf(x,t,u);
view(-40,30);
xlabel('space');
ylabel('time');
zlabel('solution u');
title(['The Brusselator for N = ' num2str(N)]);
% ------------------------------------------------

function dydt = f(t,y)

c = 0.02 * (N+1)^2;
dydt = zeros(2*N,size(y,2));  % preallocate
dy/dt

% Evaluate the two components of the function at
% one edge of the grid (with edge conditions).

i = 1;
dydt(i,:) = 1 + y(i+1,:).*y(i,:).^2 - 4*y(i,:) + ...
            c*(1-2*y(i,:)+y(i+2,:));

dydt(i+1,:) = 3*y(i,:) - y(i+1,:).*y(i,:).^2 + ...
              c*(3-2*y(i+1,:)+y(i+3,:));

% Evaluate the two components of the function
% at all interior grid points.

i = 3:2:2*N-3;
dydt(i,:) = 1 + y(i+1,:).*y(i,:).^2 - 4*y(i,:) + ...
```

```
            c*(y(i-2,:)-2*y(i,:)+y(i+2,:));

    dydt(i+1,:) = 3*y(i,:) - y(i+1,:).*y(i,:).^2 + ...
                  c*(y(i-1,:)-2*y(i+1,:)+y(i+3,:));

    % Evaluate the two components of the function at
    % the other edge of the grid (with edge conditions).

    i = 2*N-1;
    dydt(i,:) = 1 + y(i+1,:).*y(i,:).^2 - 4*y(i,:) + ...
                c*(y(i-2,:)-2*y(i,:)+1);

    dydt(i+1,:) = 3*y(i,:) - y(i+1,:).*y(i,:).^2 + ...
                  c*(y(i-1,:)-2*y(i+1,:)+3);

    end   % End nested function f
    end   % End function brussode

    % ------------------------------------------------

    function S = jpattern(N)

    B = ones(2*N,5);
    B(2:2:2*N,2) = zeros(N,1);
    B(1:2:2*N-1,4) = zeros(N,1);
    S = spdiags(B,-2:2,2*N,2*N);
    end
```

Tyson's 1991 Model `function tyson91`

```
    %This code models the metaphase/anaphase checkpoint
    %using code from Tyson's 1991 paper.

    tspan = [0 200];
    y0=[.2;.25];
    k6 = .075;
    k4 = 5;
```

```
k1 = 0.015;
alpha = 0.018/10;
axes = [0 .35 0 1];

%parameters taken from Tyson1991
%alpha = (k4)'/k4
%k1 = rate of synthesis of Cdk1 from amino acids
%k4 = rate of autocatalytic activation of MPF
%k6 = rate of breakdown of active cyclin B/Cdk1 complex

subplot(1,2,1);
options=odeset('OutputFcn','odephas2');
[t,y]=ode45(@f,tspan,y0,options);
title('Limit cycle for tyson91 eq., k_4 = 5, k_6 = .075');
xlabel('Concentration of active MPF');
ylabel('Concentration of total cyclin');
axis([0 .35 0 1]);

subplot(1,2,2);
plot(t,y(:,1),'-',t,y(:,2),'--');
title('Soln of tyson91 eq., k_4 = 5, k_6 = .075');
xlabel('time t');
ylabel('Concentration');
legend('Active MPF','Total cyclin');
axis([0 200 0 1]);

%y(1) = [M]/[CT] = (conc of active MPF)/(total cdc2)
%y(2) = ([Y]+[pM]+[M])/[CT] =
%             (cyclin + preMPF + active MPF)/(total cdc2)
%-------------------------------------------------------

function dydt = f(t,y)

dydt= [(k4)*(y(2)-y(1))*((alpha) + y(1)^2)-(k6)*y(1) ...
        (k1)-(k6)*y(1)   ];
  end
end
```

Chapter 3

Cells and Their Dynamical Systems

By Deborah Batista, Patricia Moll, Sergei Mordovine, and Ena Vu

3.1 Introduction

In order to investigate a novel approach for early detection of oral cancer that incorporates information about structural changes of cell tissue, our group has focused on modeling cell growth and cell division. Modeling of cells is important because cells are the fundamental building blocks of the body. Therefore a good understanding of cell tissue is required in order to further medical technology. We have studied and implemented recent detailed cell models and hope that our findings are able to bring a better understanding for the development of future tissue models.

In our project, we reviewed several journal articles that describe cell models and selected one particular model that we felt looked very promising. We chose the model in Qu et al's paper, "Dynamics of the Cell Cycle: Checkpoints, Sizers, and Timers", as an application to study and imitate. We like the model that is in this article for many reasons; first of all, because it can be used to produce both G1/S and G2/M checkpoints, it can also be used to depict different species by changing various parameters, and finally, because it is a generic mathematical model of a cell cycle signaling network. Before describing the model further, in section 2 we will give a description of the biology of a cell that is essential to discuss our model. Following the biolog-

ical background description, in section 3 we will give a detailed presentation of the mathematical model.

3.2 Cell Cycle Background

From the biologist's perspective, the cell is life's most basic unit. Understanding how cells grow and divide is directly relevant to the ability of medical experts to detect cancer and other medical problems. Some detailed descriptions of the current understandings of cell cycle and cell differentiation can be found in [1], [28], [41], [52].

Rapidly dividing human cells have a cell cycle that lasts about 24 hours. Two fundamental parts of the cell cycle are the following: Interphase, which the cell is in the majority of the time, and Mitosis, which last about 30 minutes. Interphase consists of three phases: G1-phase, S-phase (short for synthesis phase), and G2-phase. During G1-phase, the cell senses growth signals, increases in size, produces RNA, and synthesizes proteins. Following the G1-phase is the S-phase, whose main feature is to replicate DNA. The last phase of Interphase is the G2-phase, where the cell continues to grow, produces necessary proteins required for cell division, and is checked to ensure that the DNA has been replicated successfully. After Interphase is Mitosis, which is where the cell divides into two daughter cells. Mitosis is the only place where changes in the cell are visible to the human eye. In a typical mammal cell, G1-phase lasts about 12 hours, S-phase lasts about 6 hours, G2-phase! lasts about 6 hours, and mitosis lasts about 30 minutes.

During a cell cycle a cell grows from its birth size to a critical size. This is referred to as a sizer. After division if a cell is already at its critical size or larger, then it does not have a sizer phase. After a cell reaches its critical size, it is ready to divide. The time it takes to divide into two daughter cells is called a timer, which is a constant time. The cell is regulated by checkpoints to maintain the conditions of the cycle before moving on to the next stage. These checkpoints are within the G1-phase (G1/S checkpoint), G2-phase (G2/M checkpoint), and Mitosis. The mathematical model that we examined relates the molecular procedure of the G1/S and G2/M checkpoints with the properties of the cell cycle. Knowledge of these networks is fundamental to making predictions about cellular conductivity while the ability to model these phases and processes mathematically saves biologists experimental lab time and costs.

Within the cell cycle, checkpoints, cell size, and the sizer and timer phases are regulated by a signaling network of kinases and phosphates. Kinases are enzymes that catalyze the transfer of a phosphate group from a donor to an acceptor. The cell cycle of higher eukaryotes involve the following cyclins and cyclin dependent kinases: Cyclin A (CycA), Cyclin B (CycB), Cyclin E (CycE), CDK1, and CDK2. Cyclins are defined as a class of proteins that fluctuate in concentration at specific points during the cell cycle. Cyclins also regulate the cycle by binding to a kinase. In the G1/S transition there is an increase in activity of the binding of the regulating protein Cyclin E and enzyme CDK2. In the G2/M transition there is an increase in activity of the binding of the regulating protein Cyclin B and CDK1. Although these chemical complexes have different functions in the cell cycle, the signaling networks regulating their activities are similar, as illustrated in Figure 2.1. For these reasons, Qu et al. utilized a generic model to describe the G1/S or G2/M transition [57].

3.3 Mathematical Model

Recently biologists have recognized mathematical modeling as an important tool in understanding cell cycle signaling networks. Molecular biologists and biochemists, such as Tyson and Goldbeter, have utilized systems of nonlinear differential equations to describe the rate of change of chemical reactions within a cell cycle , [25], [49], [58], [68]. The above body of work illustrates that numerical solutions of ordinary differential equations can provide valuable insights into the cell's internal dynamics and enable scientists to form new hypothesis to guide future laboratory experiments.

The model we chose to study incorporates checkpoints, sizers, and timers. In past research papers, G1/S checkpoints were modeled using a saddle-node bifurcation and the G2/M checkpoint were modeled using a saddle-node-loop bifurcation [71, 68], (See Appendix A for description of the nonlinear dynamics term). The model in the paper we used utilizes a Hopf bifurcation to model both the G1/S and G2/M checkpoint. A Hopf bifurcation occurs when the steady state changes from a stable focus, to an unstable focus. This happens when a control parameter μ, for the decay rate of the system, becomes slower and slower and finally changes to a growth rate at a critical value μ_c. Then the equilibrium state loses stability and goes from a stable spiral to a stable limit cycle with an unstable spiral at the origin.

The sizer and timer phases, which have been overlooked in a number of mathematical models, are important features. The underlying assumption that utilizes sizers and timers is the fact that cells, in general, divide asymmetrically. Therefore, cell birth sizes vary and the time in which it takes a cell to reach a critical size is different in each case. This model, by using a Hopf bifurcation, shows the features of sizers and timers as they arise naturally in this signaling network.

3.3.1 Understanding The Model

The cell cycle engine consists of many proteins where Cyclin Dependent Kinase (CDK) is the controller of the cell cycle, as shown in Figure 2.1. CDK is the controller because it initiates crucial events of the cell cycle by adding phosphate to specific proteins. CDK becomes active when it binds to a Cyclin protein, and in turn we get a Cyclin:CDK complex. Cyclin is also activated by binding back to CDK, forming a protein complex. Arrow 3 in Figure 2.1 points at one of the Cyclin:CDK complex. The number of Cyclin:CDK complexes depend on how many Cyclins are present [18].

The three equations (1a), are part of the whole system that models Cyclin and CDK regulation. In the G1/S and G2/M transition, Cyclin is being synthesized at a constant rate of k_1 and is broken down at a constant rate of k_2. The complex is inactive because phosphate groups are attached to the Cyclin:CDK complex (see Figure 2.1 Arrow 3). Recall that CDK belongs to a group of proteins, and proteins are made up of amino acids. Phosphorylation occurs as the phosphate groups bind with different amino acids, depending on which checkpoint the complex is in [57].

In order for the Cyclin:CDK complex to become active, Cell Division Cycle 25 (CDC25) must remove phosphates from the complex. In order for CDC25 to become active, it needs to be phosphorylated by an active Cyclin:CDK complex. This forms a *positive feedback loop* shown in Figure 3.2. In contrast, wee1 is a protein kinase that inhibits Cyclin:CDK activity by phosphorylating the Cyclin:CDK complex at the amino acids, and is deactivated by phosphorylation. The active Cyclin:CDK complex, with one phosphate group, can deactivate wee1. Thus a *double-negative feedback loop* is formed (Figure 3.3)

We assume that the protein synthesis rate and the total CDK are constant [57]. The breaking down of cell cycle proteins usually occurs through ubiquitanation. Ubiquitination is a process in which Ubiquitin protein attaches to

cell cycle protein, such as Cyclin. This inactivates the cell cycle protein and acts as a signal for elimination of the Cyclin. In G1/S transition, S-phase kinase associated protein (SKP2) tags Cyclin E for ubiquitination. To get ubiquitinaiton in G2/M transition, anaphase promoting complex (APC) tags Cyclin B.

The active Cyclin:CDK may also be inhibited by binding to a CDK inhibitor (CKI). The bound Cyclin:CDK:CKI complex is only degradable when it is phosphorylated by an active Cyclin:CDK complex. The degradation process breaks the Cyclin:CDK and CKI bond, thus, the Cyclin:CDK is free to recycle in the system. This forms another *positive feedback loop*.

TABLE 1 Differential equations, variable definitions, and default parameters.

Differential equations for cyclin and CDK regulation

$$\begin{aligned}
\dot{y_1} &= k_1 + k_4 y_2 - k_3 y_1 c - (k_2 + k_{2u} y_9) y_1 \\
\dot{y_2} &= k_3 y_1 c + (k_6 + y_7) y_3 - k_4 y_2 - (k_5 + y_6) y_2 \\
\dot{y_3} &= (k_5 + y_6) y_2 - (k_6 + y_7) y_3 - (k_7 + k_{7u} y_9) y_3 - k_{14} y_3 y_{10} \\
&\quad + k_{15} y_{11} + (k_{16} + k_{16u} y_9) y_{12}, \qquad (1a)
\end{aligned}$$

where $c = (c_0 - y_3 - y_2 - y_{11} - y_{12})/c_0$.

Differential equations for CDC25 regulation

$$\begin{aligned}
\dot{y_4} &= k_8 + k_z^- y_5 - k_z^+ y_4 - k_9 y_4 \\
\dot{y_5} &= k_z^+ y_4 + k_z^- y_6 - k_z^- y_5 - k_z^+ y_5 - k_9 y_5 \\
\dot{y_6} &= k_z^+ y_5 - k_z^- y_6 - k_9 y_6, \qquad (1b)
\end{aligned}$$

where $k_z^+ = b_z + c_z y_3$ is the rate constant for CDC25 phosphorylation and $k_z^- = a_z$ is for dephosphorylation, b_z is the rate constant for CDC25 phosphorylation not catalyzed by active Cyclin:CDK, and $c_z y_3$ is for phosphorylation catalyzed by active Cyclin:CDK.

Differential equations for wee1 regulation

$$\begin{aligned}
\dot{y_7} &= k_{10} + k_w^- y_8 - k_w^+ y_7 - k_{11} y_7 \\
\dot{y_8} &= k_w^+ y_7 - k_w^- y_8 - k_{11} y_8, \qquad (1c)
\end{aligned}$$

where $k_w^+ = b_w + c_w y_3$ is the rate constant for wee1 phosphorylation and

$k_w^- = a_w$ is for dephosphorylation, b_w is the rate constant for wee1 phosphorylation not catalyzed by active Cyclin:CDK, and $c_w y_3$ is for phosphorylation catalyzed by active Cyclin:CDK.
Differential equation for SKP2 or APC regualtion

$$\dot{y_9} = \left(\frac{y_3^2}{a^2 + y_3^2} - y_9 \right) / \tau. \tag{1d}$$

Differential equations for CKI regulation

$$\begin{aligned} \dot{y_{10}} &= k_{12} - k_{13} y_{10} - k_{14} y_3 y_{10} + k_{15} y_{11} \\ \dot{y_{11}} &= k_{14} y_3 y_{10} - k_{15} y_{11} + k_i^- y_{12} - k_i^+ y_{11} \\ \dot{y_{12}} &= k_i^+ y_{11} - k_i^- y_{12} - (k_{16} + k_{16u} y_9) y_{12}, \end{aligned} \tag{1e}$$

where $k_i^+ = b_i + c_i y_3$ is the rate constant for CKI phosphorylation and $k_i^- = a_i$ is for dephosphorylation, b_i is the rate constant for CKI phosphorylation not catalyzed by active Cyclin:CDK, and $c_i y_3$ is for phosphorylation catalyzed by active Cyclin:CDK.
Variable definitions

y_1	Free cyclin
y_2	Inactive Cyclin:CDK complex
y_3	Active Cyclin:CDK complex
c_0	Total CDK
c	FreeCDK (normalized with c_0)
y_4	Unphosphorylated CDC25
y_5	One-site Phosphorylated CDC25
y_6	Two-site Phosphorylated CDC25
y_7	Unphosphorylated wee1
y_8	Phosphrylated wee1
y_9	Active SKP2 or APC
y_{10}	Free CKI
y_{11}	Cyclin:CDK:CKI complex with CKI unphosphorylated
y_{12}	Cyclin:CDK:CKI complex with CKi phosphorylated

Default parameters

$k_1 = 300, k_2 = 5, k_3 = k_4 = 30, k_5 = 0.1, k_6 = 1, k_7 = 10, k_8 = 100, k_9 = 1, k_{10} = 10, k_{11} = 1, k_{12} = 0, k_{13} = 1, k_{14} = 1, k_{15} = 1, k_{16} = 2, k_{2u} = 50, k_{7u} = 0, k_{16u} = 25, c_0 = 200, a = 4, \tau = 25, a_z = a_w = q_i = 10, b_z = b_w = b_i = 0.1, c_z = c_w = c_i = 1.$

Another assumption of the model is that CDC25 controls the entry to the cell cycle, it can have as few as 2 phosphorylation sites where the phosphate binds, and it is synthesized at a constant rate k_8. We also assume that all forms of CDC25 have degradation rates proportional to their concentration with constant coefficient k_9, where its regulation is represented by (1b) of Table 1.

When modeling wee1 regulation, which is represented by (1c), we assume that wee1 is synthesized at a constant rate k_{10}, and both phosphorylated and unphosphorylated forms of wee1 have degradation rates proportional to their concentrations. Since only unphosphorylated wee1 is active, the third equation of (1a) in Table 1 illustrates that active Cyclin:CDK is impacted by inactive wee1 y_7.

SKP2 and APC have similar roles, they are both proteins that mark the cyclin proteins for elimination. Recall that SKP2 is an F box type protein whose role is to catalyze phosphorylation dependent ubiquitination of G1 cyclins and other cell cycle proteins. Unfortunately, no information is available on the regulation of SKP2 in the cell cycle [57]. But we do know that the levels of SKP2 decrease as cells exit the cell cycle and increase as cells reenter the cell cycle. SKP2 is expressed and phosphorylated during the G1/S transition and S phase of the cell cycle, as shown in Figure 3.4. In addition, we assume that SKP2 is degraded at a rate proportional to its concentration.

The role of APC in the G2/M transition phase is to target Cyclin B for elimination. In order for APC to be active, it must be phosphorylated and bind to CDC20. Active CDK1 can catalyze the phosphorylation of APC and CDC20 directly or indirectly. APC is inactivated at a rate proportional to its concentration. This is represented in equation (1d) of Table 1.

In order for Cyclin:CDK:CKI complex to be degraded, it needs to be phosphorylated by active Cyclin:CDK. Our model assumes that CKI is synthesized at a constant rate k_{12}, and is degraded at a rate proportional to its concentration. The degradation of phosphorylated Cyclin:CDK:CKI complex is aided by SKP2 [57]. Equations (1e), in Table 1, model CKI regulation; bottom part of Figure 2.1 gives us a visual representation.

3.4 Simulation and Results

We have 12 differential equations and 31 parameters (see Table 1). We used Matlab command **ode45** to numerically solve our system of differential equations by using a fourth order Runge-Kutta method with variable step size. The command chooses the step size at each step to acquire the desired accuracy. We also plotted the different variables versus time to try to understand what is happening in our selected model [8].

In solving these equations with fixed constants, we are able to see that cell cycle checkpoints are an example of a Hopf bifurcation. The dynamics of the cell cycle are actually controlled by some parameter p (such as cyclin synthesis rate k_1, cyclin degradation rate k_2, cell size, CDK phosphorylation rate, or a combination of parameters). As the control parameter increases, the steady state of the system changes from low stable steady state region (Figure 4.5) to limit cycle region (Figure 4.6), and finally to a high stable steady state (Figure 4.7). If we choose k_1 as our control parameter, and keep all other Table 1 parameters constant, a Hopf bifurcation occurs when k_1 is between 181 and 399. While for the control parameter k_2, holding other parameters constant, a Hopf bifurcation occurs when k_2 is between -4.3 and 8.7.

In the top part of Figure 4.6, the active cyclin:CDK (solid blue line), free cyclin (the dashed green line), and total cyclin (solid red line), are plotted versus time for fixed parameter $k_1 = 300$ and fixed constants, which are presented in Table 1. It can be seen from Figure 4.6, when we have a limit cycle, the amount of free cyclin dramatically declines as Cyclin:CDK is activated. Over an interval of approximately 40 time units, the active Cyclin:CDK gradually decays and the free cyclin level gradually increases.

We simulated the incorporation of sizer and timer phases of the cell cycle model as described in Qu et. al. [57]. In order for a cell to begin DNA replication, a cell must reach a proper critical size. Cell size may impact the synthesis rate of the cyclins k_1, so we now assume that the synthesis of cyclins is proportional to the cell size and that the cell grows exponentially, instead of k_1 being a constant. The following equations were combined with the equations presented in Table 1 to simulate the G1/S model:

$$s(t) = s_0 \exp(\mu t), \qquad k_1 = \kappa s(t)/[s_c + s(t)] \tag{2}$$

where s_0 is the cell size at the beginning of the cycle, and μ, κ, and s_c are

constants. In the simulation, the values of μ, κ, and s_c are 0.003, 1000, and 40 respectively. Qu et al. neglected how they defined the first cell division occurrence T. Yet, through correspondence with the author, we discovered that the timing of the first cell division is defined to occur after the first active cyclin:CDK peak when the active cyclin:CDK equals 0.3. At this particular time point, T, the initial cell size, is set to s(T)/2. Then we simulated the values of the cell size, $s(t)$, and synthesis rate, k_1, according to equations (3.2).

The active cyclin:CDK activity peaks represent passing through a checkpoint, therefore the sizer phase is represented by the time from the cells birth to the peak. Cell division should occur after passing through this checkpoint, and the time it takes is the timer phase. Note that in the second panel, as the cyclin:CDK activity (solid red line) increases, the inactive cyclin:CDK (solid green line) and free cyclin (solid blue line) activity decreases; this portion of the plot is in reason with our assumptions. Recall that CDC25 must be phosphorylated by active cyclin:CDK to become active. Thus as the active cyclin:CDK complex increases, so do phosphorylated CDC25. In general the level of unphosphorylated CDC25 is constant when active cyclin:CDK is about zero but the unphosphorylated CDC25 levels drop as the cyclin:activity is increased. This is exactly what is illustrated in the center panel. In addition, recall that as active cyclin:CDK compl! ex phosphorylates wee1. Thus, with increased active cyclin:CDK activity, phosphorylated wee1 is increased. Furthermore, the unphosphorylated wee1 also phosphorylates active cyclin:CDK complex making the cyclin:CDK complex inactive. The unphosphorylated wee1 remains at constant levels when there is no active cyclin:CDK complex but drops as the level of active cyclin:CDK complex increases. In the bottom panel, we see that SKP2 increases as the cell reenters the cell cycle and as the level of SKP2 decreases, the cell exits the cell cycle.

3.5 Conclusion

We believe that this generic mathematical model of G1/S and G2/M transitions serves as a great stepping stone for the future development of a cell cycle model. For future Math Clinics, students may consider coupling G1/S and G2/M equations and study the coupled systems. In addition, we suggest that future Math Clinics pursue mathematical models developed by these authors. We see the next step being: to understand Qu et al's more recent mathemat-

ical model of the signaling pathways involved in cell growth, which predicts that cell growth rate is proportional to cell surface area at birth [58]. These mathematical models appear to successfully reproduce sizer, timer, and restriction point features of the cell cycle, in addition to experimental findings.

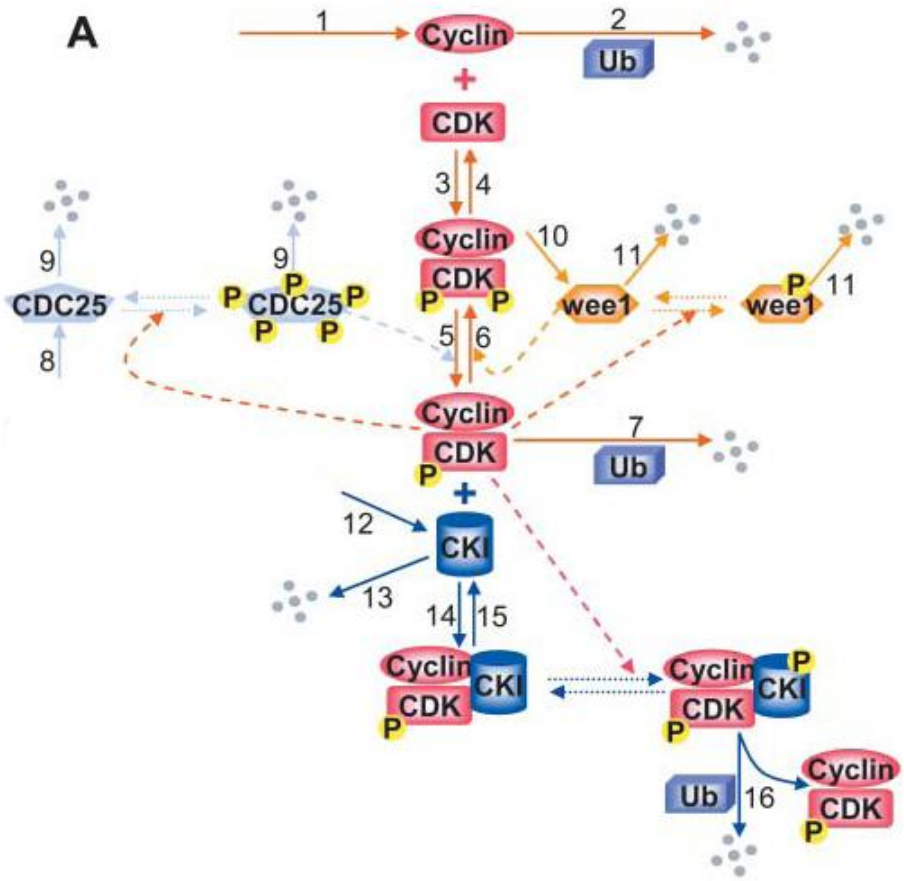


Figure 2.1: Wiring Diagram of Cell Cycle

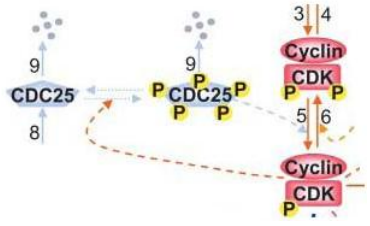


Figure 3.2: Positive Feedback Loop

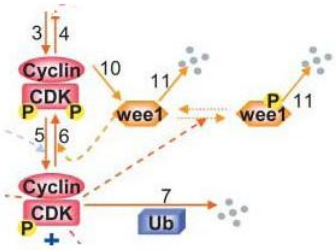


Figure 3.3: Double-Negative Feedback Loop

Figure 3.4: SKP2

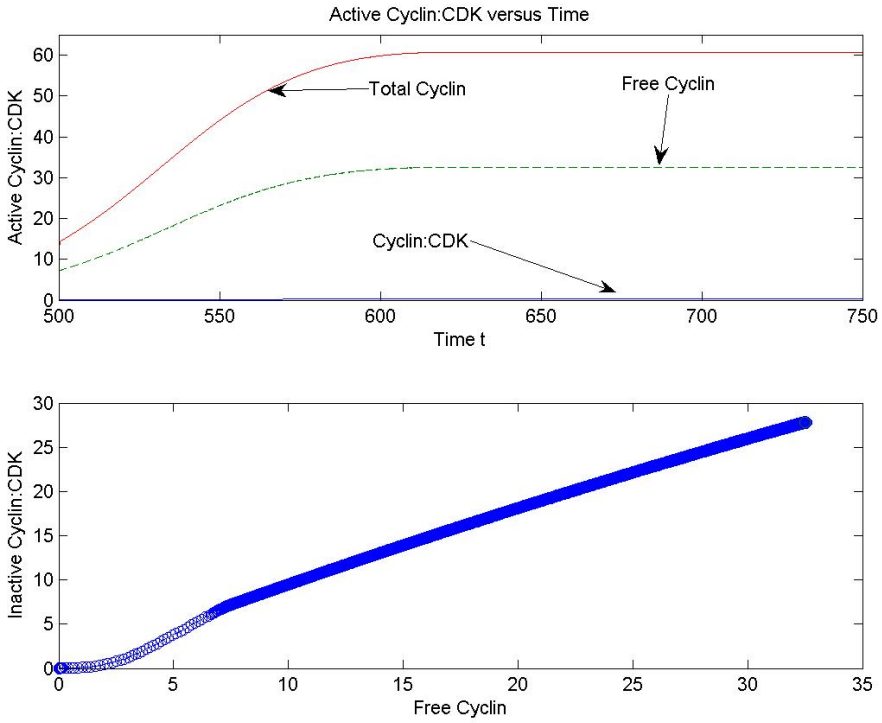


Figure 4.5: Steady state of the system with $k_1 = 181$

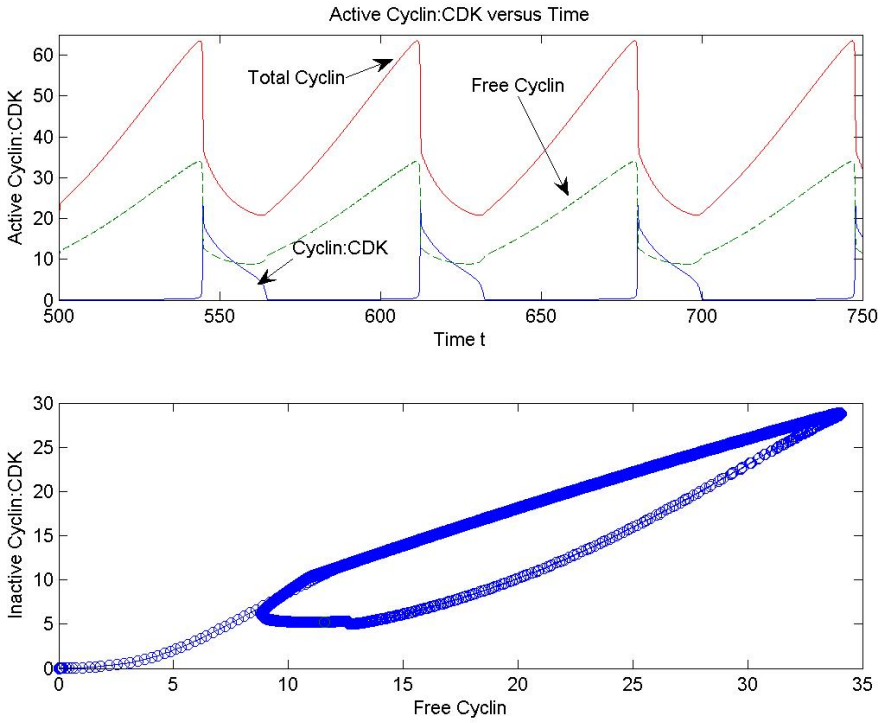


Figure 4.6: The active cyclin:CDK, free cyclin and total cyclin versus time in the system for $k_1 = 300$ illustrated here is in the limit cycle region.

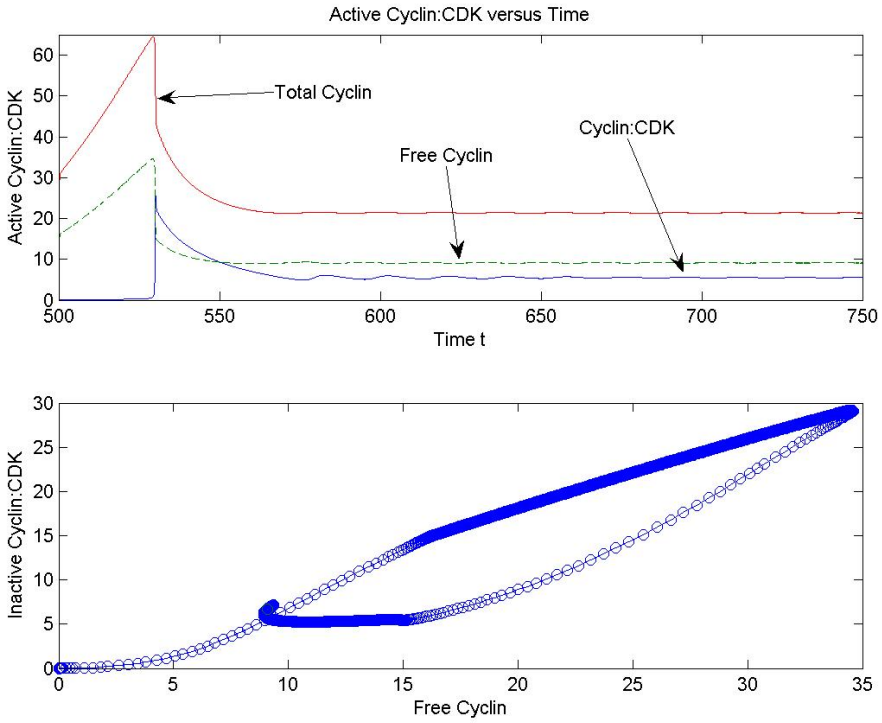


Figure 4.7: Steady state of the system with $k_1 = 400$

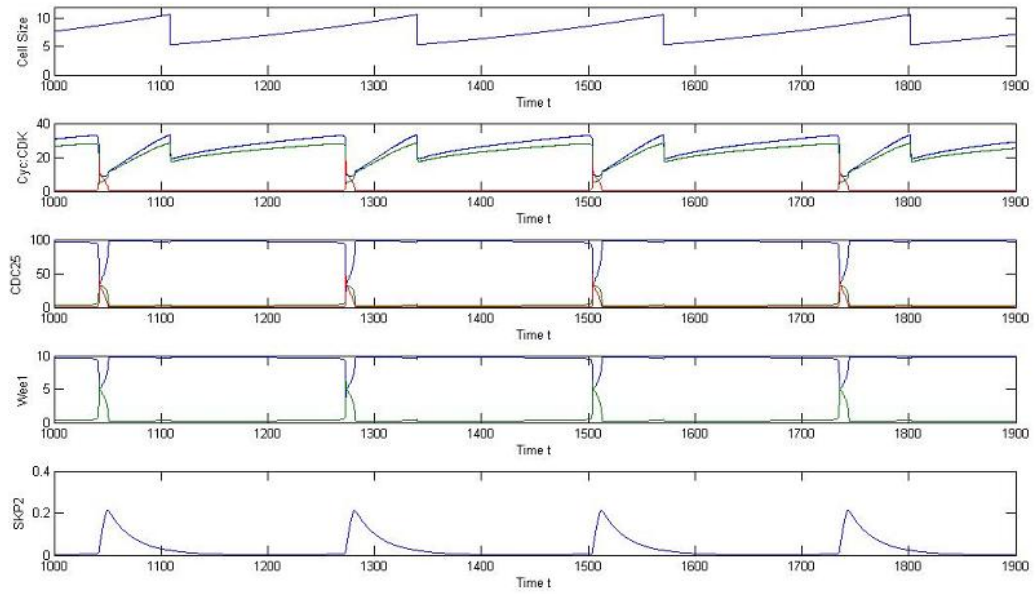


Figure 4.8: The top panel shows cell size (solid blue) versus time for the control group. The second panel illustrates the cyclin:CDK activity of the cell, where the solid red line is the active cyclin:CDK, the solid green line is the inactive cyclin:CDK complex and the solid blue line is the free cyclin. The middle panel represents CDK25 regulation. Here the blue solid line represents unphosphorylated CDC25, green line is one site phosphorylated CDC25, and the red line is the two site phosphorylated CDC25. The second panel from the bottom represents wee1 regulation. The blue solid line is unphosphorylated wee1 while the green solid line represents phosphorylated wee1. The bottom panel is a plot of SKP2. All of the plots illustrated here are plotted against time.

Appendix A: Nonlinear Dynamics

The cell cycle can be mathematically modeled by a set of nonlinear ordinary differential equations:

$$\frac{dy_i}{dt} = f_i(y_1, y_2, .., y_n; p_1, ...p_m) \qquad \text{for} \quad i = 1, \ldots, n$$

where y_i = concentration (or activity) of the i-th protein in the reaction network, and p_j = value of the j-th parameter rate constant [70]. Each function f_i describes chemical rate equation in terms of synthesis, degradation, activation, and inactivation of proteins of the cell cycle. In particular, the form of the chemical rate equation is as follows:

$$f_i = \text{synthesis - degradation + activation - inactivation,}$$

where synthesis, degradation, activation, and inactivation are nonlinear functions of the variable concentrations and constant parameters in the model [70, 57]. The exact forms of the functions f_i is governed by a set of chemical rate laws and the assumptions made for each reaction. A model is composed of three parts: a set of rate equations $f_1, \ldots, f_n$, a set of parameters $p_1, \ldots, p_m$, and a set of initial conditions $y_1(0), \ldots, y_n(0)$. With the specification of the functions f_j, parameters p_k, and initial conditions $y_j(0)$, the differential equations can be solved numerically to give the time-dependence of each component protein (See Appendix B on how to solve numerically differential equations) [70].

Nonlinear Dynamics Terminology

When solving differential equations, we are interested in understanding the long term behavior of the system. Our interest is to find the point at which the rate of change of all variables is zero. This point is known as the **steady state** (equilibrium solution) [57]. To find the steady state of a system of differential equations, we first set the derivatives of all the variables to zero: $f_i(y_1, \ldots, y_n, p_1, \ldots, pm) = 0$ for all i. Then simultaneously solve the resulting set of nonlinear algebra equations algebraically (or numerically). In terms of mathematical models of the cell cycle, the steady state corresponds to the point at which protein concentrations are unchanging in time.

We are also interested in **"oscillatory" solutions** of dynamical system.

For an oscillatory solution, protein concentrations change in time, repeating themselves after a characteristic period.

Steady states or oscillatory solutions can be classified as **stable** or **unstable**. A steady state is **stable** if a solution starting at a small change away from the steady state remains near the steady state. On the other hand, if a solution deviates from the point, it is not stable and is called **unstable**.

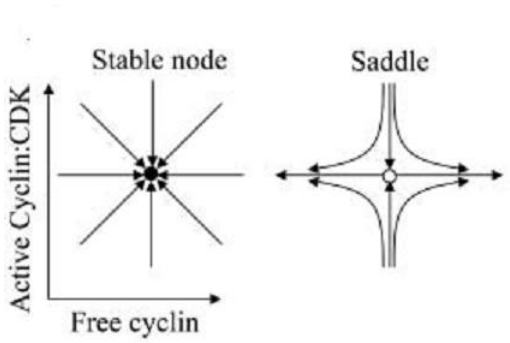


Figure 5.9: A schematic plot of a stable node (*solid circle*) and a saddle point node (*unfilled circle*).

The location and stability of the steady states of a dynamical systems depends on the particular values of parameters in the model. For example, the schematic plots in figures 5.9 and 5.10 of active Cyclin:CDK(x) and free cyclin show four common types of steady states in a two-variable space using different parameter values of the model. Figure 5.9 illustrates a **stable node**, which is a point in the vector field where all of the trajectories are attracted. In Figure 5.9, there is also an illustration of **saddle point**. With a saddle point, there are some trajectories coming towards the point but other trajectories are moving away from the point. Thus, the saddle point is classified as unstable.

Figure 5.10 illustrates a **stable focus**, which is also known as an attracting spiral since the trajectories are spiraling towards the node, and Figure 5.10 illustrates an **unstable focus** where trajectories are spiraling away from the node. These are just some representatives of possible types of steady states (equilibrium solutions), and information on the various types are available in many textbooks [21, 10, 65].

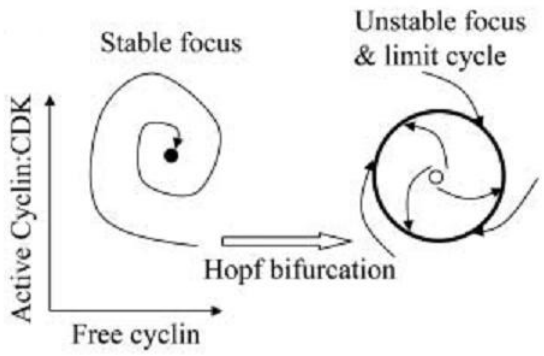


Figure 5.10: A schematic plot of a stable focus (*solid circle*),an unstable focus (*unfilled circle*) and a limit cycle (*thick large circle*).

When a parameter value is changed, then the properties of the solutions may change. For example, a stable steady state may lose its stability or even no longer exist, and an isolated trajectory enclosing the steady state, known as a **limit cycle**, may appear. Qualitative changes in the behaviors of the dynamical systems are called **bifurcations**. The qualitative changes occur at specific values of the parameters, called **bifurcation points**. Figure 5.9 illustrates the impact of changing parameter value k1, the synthesis rate of cyclins. By changing parameter the synthesis rate of cyclins, the steady state changes from a stable node to a saddle. This critical value in which the change occurs is called a **saddle-point bifurcation**. Saddle point bifurcations usually lead to **bistability**, a property of a nonlinear system in which two different stable steady states coexist in the dynamical system with an unstable steady state! in between. In a bistable system, when a parameter is increased, a sudden jump may occur at a particular value of the parameter. As the parameter is decreased, the jump back to original behavior occurs at a much lower parameter value. This type of behavior is called **hysteresis**. In a nonlinear system, another type is of bifurcation is the **Hopf bifurcation**. In Hopf bifurcation, the steady state may change from a stable focus to an unstable focus of a limit cycle oscillation as the parameter increases.

Appendix B: Solving Differential Equations with Matlab

Solving a first order differential equation in Matlab is pretty trivial for the easy equations. We use the command **dsolve**. As an example:

```
>> dsolve('Dy - t^2 + y', 't')
ans =
t^2-2*t+2+exp(-t)*C1
```

We can also add an initial value:

```
>> sol = dsolve('Dy = t^2 + y', 'y(0) = 3', 't')
sol =
-t^2-2*t-2+5*exp(t)
```

A lot of times it is a really good idea to plot the solutions to the differential equation so it can be analyzed further. Here we show Figure 5.11, which is followed by the code to plot it:

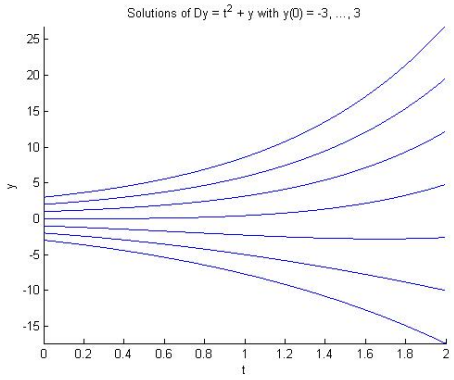


Figure 5.11: Solutions of $dy/dt = t^2 + y$ with various initial values

```
>> sol = dsolve('Dy = t^2 + y', 'y(0) = c', 't');
>> figure; hold on
>> syms t
>> for cval = -3:3
       ezplot(subs(sol, 'c', cval), [0 2])
   end
```

```
>> axis tight
>> title 'Solutions of Dy = t^2 + y with y(0) = -3, ..., 3'
>> xlabel t, ylabel y
>> hold off
```

The first line solves the equation and stores it in the variable *sol*. Notice that we specified the initial condition as *c*; this gives us the means to plot the solutions with different initial values. **figure** command opens a new graphics screen and **hold on** command keeps graphing new solutions without erasing the old ones. Then we make a loop to graph the solutions with different initial values starting with -3 to 3 with step sizes 1. So, our initial values are $[-3, -2, -1, 0, 1, 2, 3]$; that means that we should get seven solution lines and that's exactly what we got. The command **axis tight** eliminates white space at the edges. We added a title and labeled the axes. Notice how with different initial values our solutions differ. Graphing is a good tool to see how the solutions differ instead of looking at the numerical solutions.

Plotting Direction Fields

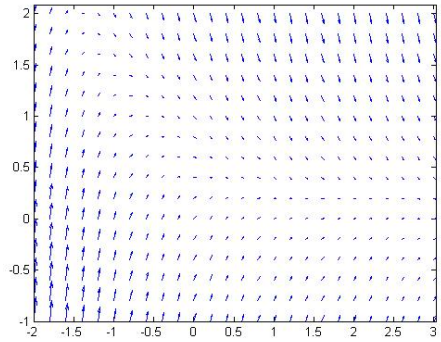


Figure 5.12: Direction Field for Equation 3.1

To plot Figure 5.12, we need to use **quiver** in conjunction with **meshgrid**. Here's an example:

```
>> [t, y] = meshgrid(-2:0.2:3, -1:0.2:2);
>> s = exp(-t) - 2*y;
>> quiver(t, y, ones(size(s)), s), axis tight
```

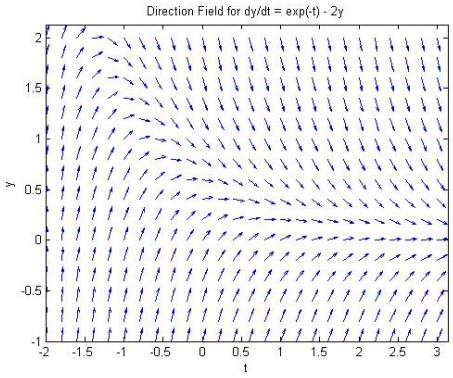


Figure 5.13: Improved Direction Field for Equation 3.1

To explain the code, **meshgrid** creates points where the function will be evaluated and in turn, the arrows are going to be put at those coordinates. In our example we have the t-values start at -2 and go up to 3 every 0.2 spaces. So the t vector looks like $[-2, -1.8, -1.6, \ldots, 2.8, 3]$. The y-values go from -1 to 2 in the same 0.2 increments.
Next line has our function defined as:

$$s = e^{-t} - 2 * y \tag{3.1}$$

We use **quiver** to plot the direction field. It takes the values of t and y, then we create an array of ones the size of s, and then we put in our function.

We can see from Figure 5.12 that some of these vectors are small and hard to read. To get the arrows to the same size, so it'll be easier to read, we input these commands:

```
>> [t, y] = meshgrid(-2:0.2:3, -1:0.2:2);
>> s = exp(-t) - 2*y;
>> l = sqrt(1 + s.^2);
>> quiver(t, y, 1./l, s./l, 0.5), axis tight
>> xlabel 't', ylabel 'y'
>> title 'Direction Field for dy/dt = exp(-t) - 2y'
```

In Figure 5.13, it is a lot easier to see the arrows and their directions. We just rescaled the arrows to the same magnitude by dividing each vector by

its length, $\sqrt{1+s^2}$. We also cut the size of the arrows in half by putting 0.5 in the quiver command. Finally, we added a title and a label for the axes.

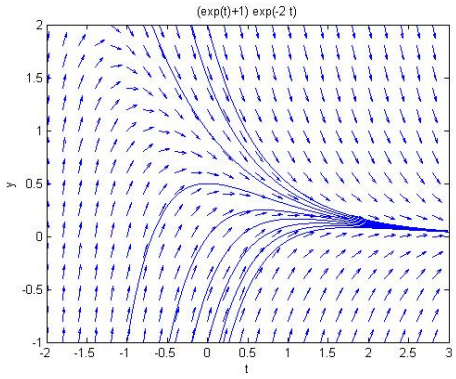


Figure 5.14: Direction Field and Solution Curves for Eqation 3.1

In addition to the most recent code we can draw the solution curves to the equation and see how the direction fields compare to the solution curves:

```
>> sol = dsolve('Dy = exp(-t) - 2*y', 'y(0) = c', 't');
>> hold on
>> syms t
>> for cval = -2:0.5:2
       ezplot(subs(sol, 'c', cval), [-2 3])
   end
>> axis([-2 3 -1 2])
```

Figure 5.14 shows that the solutions and direction fields are in sync with each other.

Next, we look at a non-linear equation. Consider the differential equation:

$$dy/dt = y^2 + t \tag{3.2}$$

The following code plots the direction field for Equation 3.2:

```
>> [t, y] = meshgrid(-2:0.2:2, -2:0.2:2);
>> s = y.^2 + t; l = sqrt(1 + s.^2);
```

```
>> quiver(t, y, 1./l, s./l, 0.5), axis tight
>> xlabel 't', ylabel 'y'
>> title 'Direction Field for dy/dt = y^2 + t'
```

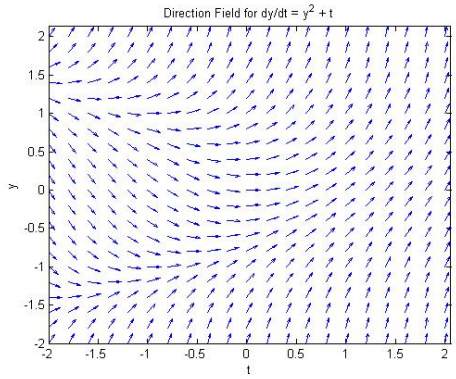


Figure 5.15: Direction Field for Equation 3.2

The code produces Figure 5.15. Matlab plots direction fields for both linear and non-linear equations using the same commands.

A lot of differential equations cannot be solved analytically, so we have to use different methods to approximate the solutions. Matlab has several functions that do just that, and the most used for approximating differential equations is **ode45**. This takes in the function, the domain on which to approximate, and the initial value. There are two ways to define our equation as a function. These are:

```
>> f = @(t, y) t./y
```

and to make an M-file that could be called, for instance, ourf.m. This M-file would contain:

```
function z = ourf(t, y)
z = t./y;
```

There are three ways that **ode45** can take in a function. The first way just takes in our f that we created:

```
>> ode45(f, [0 2], 1)
```

The next uses our M-file:

```
>> ode45(@ourf, [0 2], 1)
```

And yet the third way is to write our function in the **ode45** command itself:

```
>> ode45(@(t, y) t./y, [0 2], 1)
```

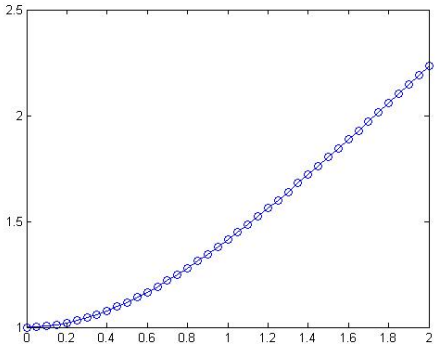


Figure 5.16: Numerical solution of $dy/dt = t/y, y(1) = 1$

All these commands automatically draw the solution curve as shown in Figure 5.16.

To use **ode45** without plotting the curve, one just has to assign it to a couple of vector variables:

```
>> [t y] = ode45(@(t, y) t./y, [0 2], 1);
```

Now we move to systems of first order differential equations. We show how to plot a single trajectory. Consider the system:

$$\begin{cases} x' = -3x + 2y, & x(0) = 1 \\ y' = -x, & y(0) = 0 \end{cases}$$

First we let **dsolve** solve the system and then plot it:

```
>> ivp = 'Dx = -3*x + 2*y, Dy = -x, x(0) = 1, y(0) = 0';
>> [x, y] = dsolve(ivp, 't');
>> xf = @(t) eval(vectorize(x));
>> yf = @(t) eval(vectorize(y));
```

We've used **vectorize** to vectorize our symbolic expressions for x and y, and then use **eval** to evaluate those expression with respect to t. We also made two function of t which we can plot with respect to time. The commands are as follows:

```
>> t = -0.3:0.1:5;
>> plot(xf(t), yf(t))
>> xlabel 'x'
>> ylabel 'y'
```

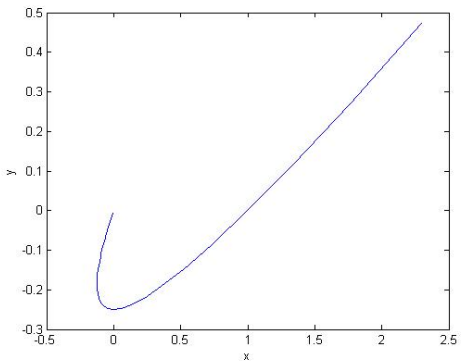


Figure 5.17: A Single Trajectory

We used the command **plot** because we need to plot x vs y. Figure 5.17 gives us a single trajectory because we only used one initial value for both $y(0)$ and $x(0)$.

We can also plot multiple trajectories by specifying a set of initial values for either $y(0)$ or $x(0)$ or for both $y(0)$ and $x(0)$. This example shows how to use a set of initial values for $x(0)$ and the result is Figure 5.18:

```
>> ivp = 'Dx = x + 2*y, Dy = -x, x(0) = a, y(0) = 0';
>> [x, y] = dsolve(ivp, 't');
>> xf = @(t, a) eval(vectorize(x));
>> yf = @(t, a) eval(vectorize(y));
>> figure; hold on
>> t = -10:0.1:10;
```

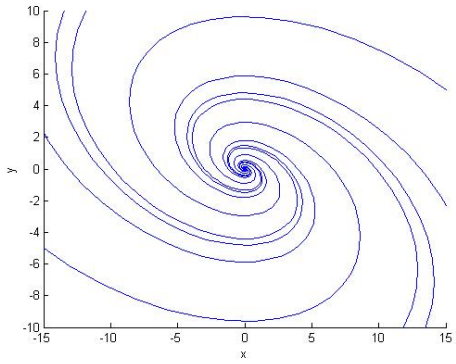


Figure 5.18: A Family of Curves

```
>> for a = -4:4
       plot(xf(t, a), yf(t, a))
   end
>> hold off
>> axis([-15 15 -10 10])
>> xlabel 'x'
>> ylabel 'y'
```

We can also use **ode45** for the equations that **dsolve** can't solve:

```
>> figure; hold on
>> f = @(t, x) [x(1) + 2*x(2); -x(1)];
>> for a = -4:4
       [t, xa] = ode45(f, [0 10], [a 0]);
       plot(xa(:,1), xa(:,2))
       [t, xa] = ode45(f, [0 -10], [a 0]);
       plot(xa(:,1), xa(:,2))
   end
>> axis([-15 15 -10 10])
```

Figure 5.19 looks exactly like Figure 5.18. From this example, we can conclude that **ode45** does a pretty good job at approximating the solution.

To plot vector field for systems of differential equations is not that much different from the previous examples:

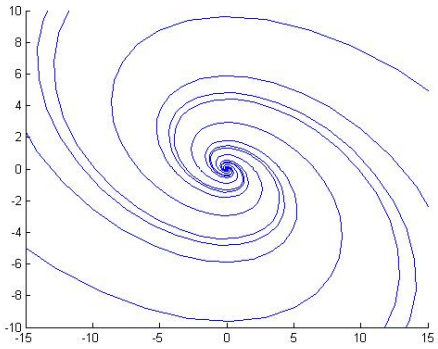


Figure 5.19: A Family of Curves using **ode45**

```
>> [x, y] = meshgrid(0:0.2:6, 0:0.2:4);
>> u = x.*(5 - x - y);
>> v = y.*(-2 + x);
>> l = sqrt(u.^2 + v.^2);
>> quiver(x, y, u./l, v./l, 0.4)
>> axis equal tight
>> xlabel 'x'
>> ylabel 'y'
```

As one can see, all we did is add another equation u or v instead of having a vector of ones. Figure 5.20 shows the popular predator-prey ploblem.

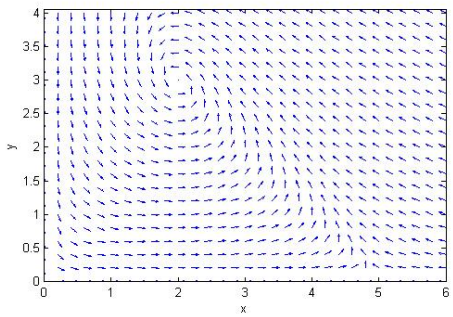


Figure 5.20: Vector Field for Predator-Prey Problem

Appendix C: Matlab Code

```
function QuOdeTimer
clear
%close
% Initiate time vector
tspan = [500 750];
% Initial values - select 13 initial values
% Look into some good initial values the initial values
y0 = [0; 0; 0; 0; 0; 0; 0; 0; .4; 0; 0; 0];

% Generic Model of G1/S or G2/M
% This code models 13 differential equations in Qu et. al 2003


% Default parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k1 = 400;
k2 = 5;
k3 = 30;
k4 = 30;
k5 = 0.1;
k6 = 1;
k7 = 10;
k8 = 100;
k9 = 1;
k10 = 10;
k11 = 1;
k12 = 0;
k13 = 1;
k14 = 1;
k15 = 1;
k16 = 2;
k2u = 50;
k7u = 0;
k16u = 25;
c0 = 200;
a = 4;
```

```
tau = 25;
az = 10;
aw = 10;
ai = 10;
bz = 0.1;
bw = 0.1;
bi = 0.1;
cz = 1;
cw = 1;
ci = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Variable definitions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% y1   Free cyclin
% y2   Inactive Cyclin:CDK complex
% y3   Active Cyclin:CDK complex
% c0   Total CDK
% c    Free CDK (normalized with y4)
% y4   Unphosphorylated CDC25
% y5   One-site phosphorylated CDC25
% y6   Two-site phosphorylated CDC25
% y7   Unphosphorylated wee1
% y8   Phosphorylated wee1
% y9   Active SKP2 or APC
% y10  Free CKI
% y11  Cyclin:CDK:CKI complex with CKI unphosphorylated
% y12  Cycle:CDK:CKI complex with CKI phosphylated
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function dydt = f(t,y)
dydt = [ k1 + k4*y(2) - ( k3*y(1)*( c0 - y(3) - y(2) - y(11) - y(12) ) ...
       /c0 ) - ( k2 + k2u*y(9) )*y(1)
     ( k3*y(1)*( c0 - y(3) - y(2) - y(11) - y(12) )/c0 ) + ( k6 + ...
      y(7) )*y(3) - k4*y(2) - ( k5 + y(6) )*y(2)
     ( k5 + y(6) )*y(2) - ( k6 + y(7) )*y(3) - ( k7 + k7u*y(9) )* ...
       y(3) - k14*y(3)*y(10) + k15*y(11) + ( k16 + k16u*y(9) )*y(12)
      k8 + az*y(5) - ( bz + cz*y(3) )*y(4) - k9*y(4)
```

```
     ( bz + cz*y(3) )*y(4) + az*y(6) - az*y(5) - ( bz + cz*y(3) )* ...
       y(5) - k9*y(5)
     ( bz + cz*y(3) )*y(5) - az*y(6) - k9*y(6)
      k10 + aw*y(8) - ( bw + cw*y(3) )*y(7) - k11*y(7)
     ( bw + cw*y(3) )*y(7) - aw*y(8) - k11*y(8)
     ( (y(3)^2/( a^2 + y(3)^2) ) - y(9) )/tau
     k12 - k13*y(10) - k14*y(3)*y(10) + k15*y(11)
     k14*y(3)*y(10) - k15*y(11) + ai*y(12) - ( bi + ci*y(3) )*y(11)
     ( bi + ci*y(3) )*y(11) - ai*y(12) - ( k16 + k16u*y(9) )*y(12) ];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  plot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
subplot(2,1,1);
[T,Y] = ode45(@f, tspan, y0);
plot(T, Y(:,3), T, Y(:,1), '--', T, ( Y(:,1) + Y(:,2) + Y(:,3) ) )
title('Active Cyclin:CDK versus Time');
xlabel('Time t');
ylabel('Active Cyclin:CDK');
axis( [500 750 0 65] )
subplot(2,1,2);
options = odeset('OutputFcn', 'odephas2');
[t,y] = ode45(@f, tspan, y0, options);
xlabel('Free Cyclin');
ylabel('Inactive Cyclin:CDK');

end
```

Chapter 4

Cell Population Model I: Modeling of Healthy Tissue in the Oral Cavity

By Jesse Gilbert, Nathan Kurtz, Megan Van Dyke, and Lea Wesbrook

4.1 Introduction

Randall Tagg is the current sponsor of the Math Clinic. He is trying to grow in vitro tissue samples and model healthy tissue. In particular, he would like to model what goes wrong in a healthy tissue sample when cancer develops. In order to understand exactly what the limitations are of in vitro tissue growth, we built a model of tissue growth that is completely abstract, but captures most of the physical and chemical forces behind cell proliferation, movement, and adhesion. To this end, our task was to research and write a MATLAB program that modeled, in two dimensions, a three-cell deep layer of epithelial tissue with a basal layer, a mucosal layer, and a squamous layer. Our program is supposed to incorporate realistic models of cell growth, death, division, adhesion and differentation. We found it necessary to define and track certain attributes of each cell in order to make the model realistic as well as dynamic. Our attributes included cell volume, integrins, cadherins, and diffusible promoters. After creating a two dimensional model of a static tissue, we were able to build several functions in MATLAB and apply those functions iteratively in order to see how our tissue evolved over time. Our

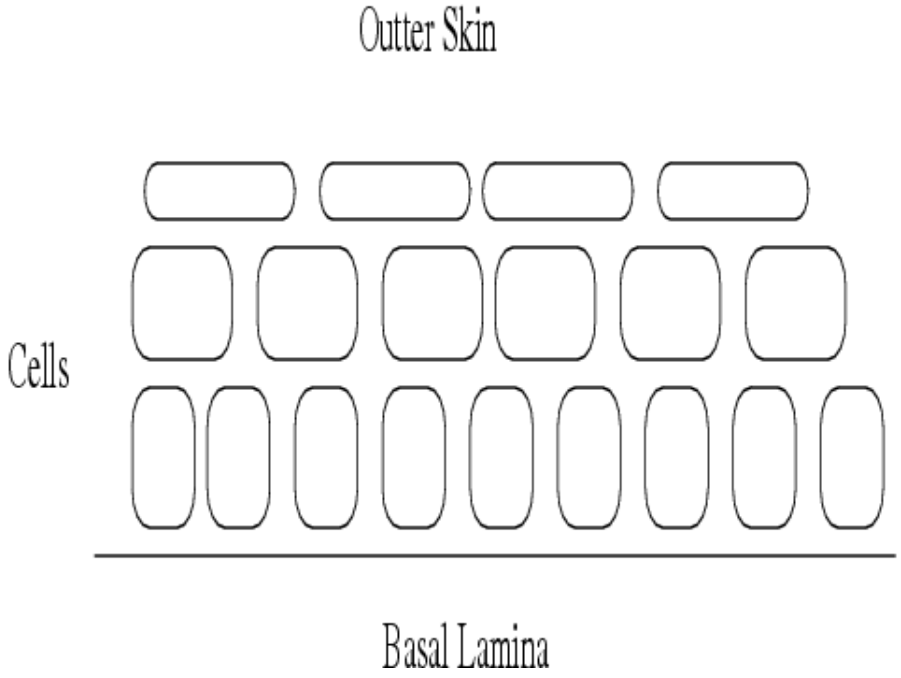
model has several functions, which set up a cell population and modeled cell death, cell movement, cell division, and cell signaling. Our model records certain characteristics, which can be displayed cell by cell after each iteration of the program. There is also a pictorial display of the tissue as a whole which can be used to track a cell's movement and growth over time. Our main goal was to create a model that remained in equilibirum over time, by which we meant the appearance of the tissue in our pictorial display remained similar throughout many iterations of the program.

Through research and our MATLAB model we were able to describe and illustrate how a system of cells changes over time. We first present the necessary mathematical and biological background for our program, then we outline the program.

We describe the following biological aspects of the cell cycle: division, growth, death and movement. We briefly describe how we tried to model these aspects using MATLAB and various cell attributes which change over time. As we outline the MATLAB program, where we implemented our model, we give a general overview, then describe the functions of the program one by one. There are several functions and we break our description of these functions into four parts: we describe the purpose of the function, analyze the relationship between our model and the function, describe how the function works and conclude how the function could be improved.

4.2 Biological Background and Overview

We modeled a 3 layer cell population. In order to model this multi-layer cell population or tissue, we had to model various attributes and functions of individual cells. We did this by organizing the cells into layer and building the functions and attributes that controlled their ability to signal, move, grow and die accordingly. The first layer was called the basal layer, the second layer was called the mucosal layer, and the outer, surface layer was called the squamous layer. Our goal was to have a set of rules that governed cell death, movement, growth, and division so that the number of cells and the appearance of the cell layers remained in an equilibrium called homeostasis. Meanwhile, the user could track an individual cell's path of growth and differentiation. We incorporated various cell signaling mechanisms which modeled ways in which a cell communicated with the population and the population communicated with individual cells. These included integrins, cadherins, and

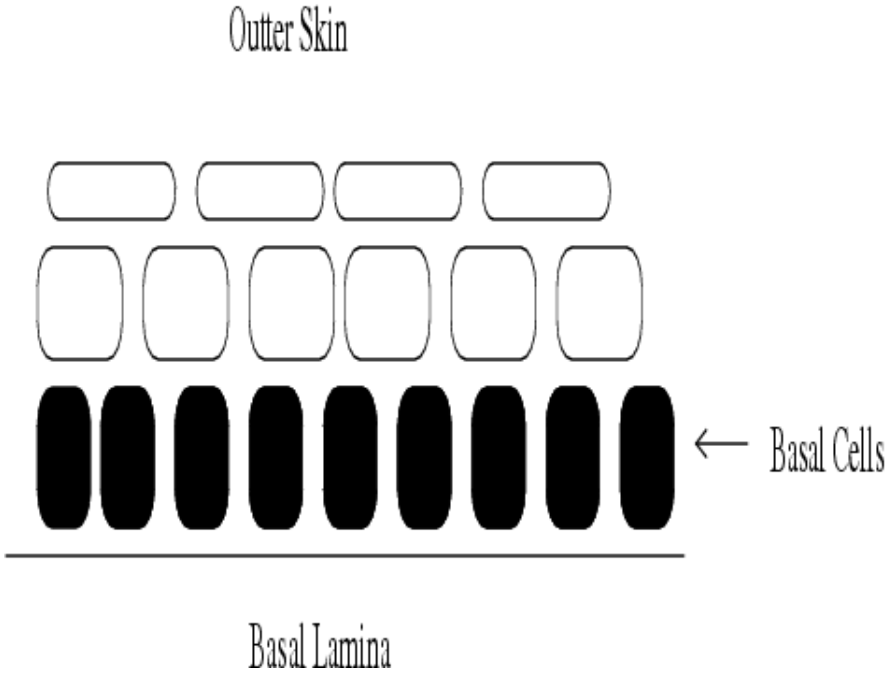


diffusible promoters which encouraged movement or chemotaxes.

The cell population was modeled by a linked list which was essentially a queue of the cells which stored information about each cell. The attributes we stored were an x-location, a y-location, a left and right neighbor, top and bottom neighbors, an integrin and a cadherin level, a left boundary, a right boundary, a cell space and an actual space.

We also wanted to model the extracellular matrix. To do this, we allowed the cell space and the volume to be separate parameters. We let volume be a percentage of the cell space and actual space be the product of cell space and cell volume. We assumed the empty percentage of the cell space did not affect the individual cells except to promote cell growth, division and movement in a healthy way.

The integrin level was based on contact area with the lamina(|left boundary - right boundary|). We assumed that integrin level was a linear function of this contact area. The cadherin level was based on the number of neighbors. We assumed that cadherin level was a linear function of the number of neighbors of a cell. The integrin level governed adhesion, while the cadherin level governed movement. We assumed that cell death, division and



movement were governed by cell volume, integrin levels, cadherin level, and the life cycle of the nieghboring cells. Once a cell reached critical mass, it divided, died or moved. The location to which a cell moved was affected by whether neighboring cells had died. We called one attribute the diffusible promoter or just promoter. It caused cells to move towards the top layer where cells sloughed off.

The main function was cell step. It iteratively increased cell.volume and cell.promoter at different levels depending on which layer contained the cell. Cell death was the first function called in cell step. A cell died if

$$\frac{1}{\text{cell(n).cadhedrin}} + \text{cell(n).promoter} + \text{cell(n).volume} \geq 1.95.$$

When a cell died its promoters were dispersed evenly to its neighbors. A cell moved from layer 1 to layer 2 if

$$\frac{1}{\text{cell(n).cadherin}} + \text{cell(n).promoter} + \text{cell(n).volume} \geq .8.$$

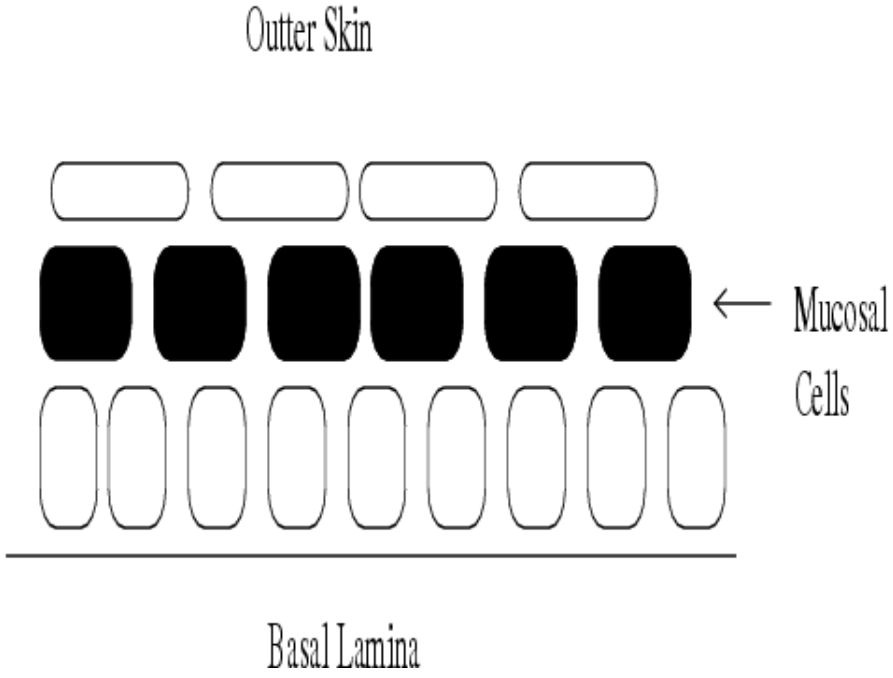


Figure 1.1: Our population of cells.

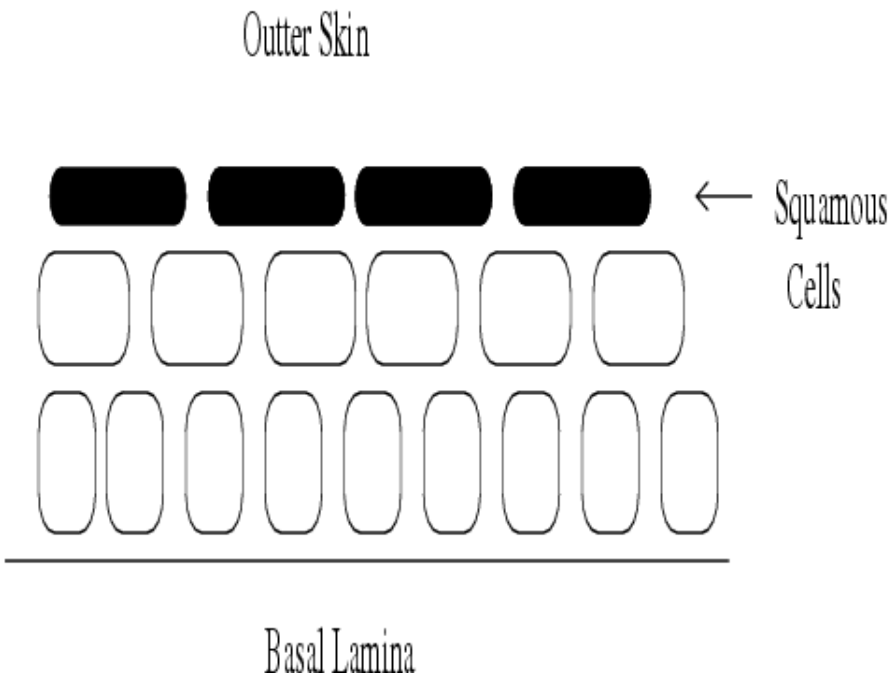
A cell moved from layer 2 to layer 3 if

$$\frac{1}{\text{cell(n).cadherin}} + \text{cell(n).promoter} + \text{cell(n).volume} \geq 1.25.$$

A cell divided if

$$.4\text{cell(n).cellspace} + .6\text{cell(n).integrin} - \text{cell(n).promoter} \geq .4.$$

In the first case we assumed that the cadherin level would inhibit cell death and the volume and promoter level would encourage cell death. In the second case we made the same assumptions about movement. In the last case we assumed that the promoter level would inhibit cell division and an increase in cell space or integrin level would promote cell division.



4.2.1 Cell Signaling

An important aspect of our model was the ability of cells to communicate among and between one another and the tissue. Our main tool in modeling cell signaling was the collection of attributes accorded to each cell. We assigned to each cell a cadherin and integrin level and these controlled, along with promoter level, cell volume, cell space, and actual space, the time at which cells died, moved, and divided. The functions and attributes that acted on a cell and were linked to the cell are described throughout the rest of the section and the report.

4.2.2 Cell Death

When looking at the shape of the human body, one might never suspect that cellular death is responsible for a healthy life. Apoptosis, or programmed cell death ("PCD"), is a unique reaction many cells undergo to prevent disease and damage to the human body. Its signaling, cycle, and completion are carried out in an ordered fashion, and it is a command many cells are

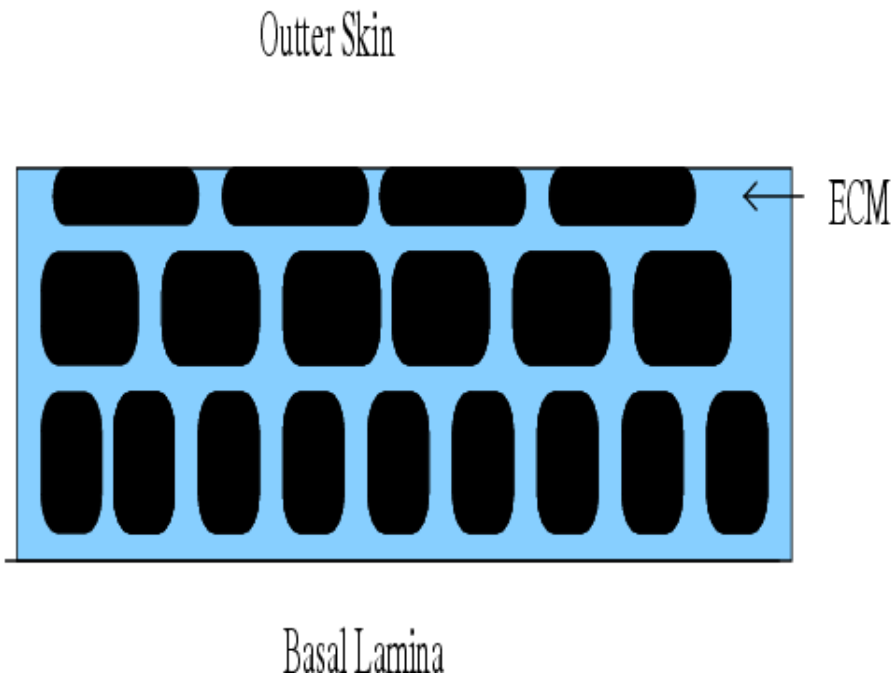


Figure 2.2: Our population of cells.

required to do at some point during their life. In order to comprehend this programmed cellular suicide, it is best to understand its basic functions and importance, and also the reasons why it may fail. [79]

A cell cycle contains of 4 stages, each of which help the cell to prepare for mitosis. After mitosis has occurred, a cell may enter what is known as phase G_0. This phase occurs when a cell no longer readies itself for mitosis, but instead continues carry out its function until it either re-enters the cycle, or dies. Should the cell die, it can undergo two types of death: a death caused by damage that in turn damages the surrounding tissue, or apoptosis.

When a cell is subjected to some kind of damage (DNA or trauma) that results in death, the human body suffers the consequences. The plasma membrane, which is a sort of "skin" for a cell, is no longer able to control the passage of essential fluids, and thus causes a swelling of the cell. Cellular contents begin to leak and cause inflammation in the surrounding tissue. This is why we see swelling of an injury site. Chronic inflammation, which can occur due to damaged DNA causing cellular death, can lead to tissue

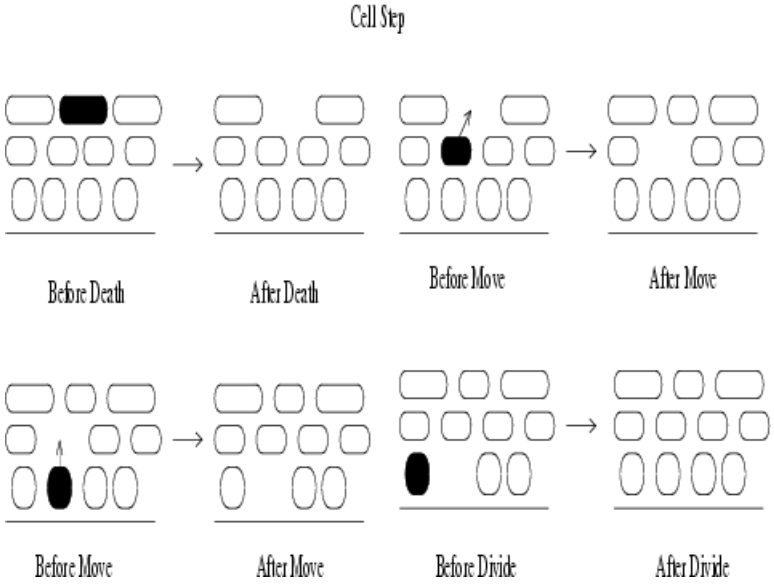


Figure 2.3: A an iteration of cell step.

destruction. Because this can be so damaging, it is best that we rely upon PCD to rid the body of potentially harmful cells.

Apoptosis can be generated by signals within the cell itself, or by external cellular triggers. Internal signals begin the process by withdrawing all signals that the cell relies upon to continue its cycle or function. Next, the cell receives negative signals, which can be caused by molecular binding, protein accumulation, or high levels of oxidants, among many reasons. These negative signals tell the cell to first begin shrinking. The DNA and protein, called chromatin, become tightly packed within the cell. This condensation, called pyknosis, is trademark of apoptosis. Next, the mitochondria, which are responsible for converting food into energy, begin to degrade along with the chromatin during a process known as karyorrhexis. The cell membrane will then form bubble-like structures known as blebs, and eventually the cell begins to break apart. In order to rid the body of these floating cell bits, specialized cells called phagocytes begin a process called phagocytosis, or "cell-eating". This process literally consumes the bits of dead cell, keeping the body free of necrosis.

Should cells not undergo apoptosis, we may begin polluting our bodies

with potentially cancerous cells. At its core, cancer is uncontrolled cellular division that can form into tumors and invade other tissues. Apoptosis helps to regulate cell division, and is associated with a tumor-suppressor protein called p53. A precancerous cell undergoes DNA mutations. p53, located on chromosome 17, begins to accumulate as the mutated DNA is replicated during mitosis of the altered cell. This protein will then stop the cell cycle at phase 1 (the growth phase) and allow the cell to repair itself. If it does not repair, then p53 induces apoptosis, thus killing off the harmful cellular body. However, there are certain types of cancers, such as melanomas, leukemias, lung, and colon cancer amongst others that have developed ways to inhibit or even block the signal altogether. By expressing high levels of certain proteins, these cancer cells can become immune to PCD.

Apoptosis is an extremely important function within the body. An average adult may lose 50 billion to 70 billion cells each day due to apoptosis. We rely on cellular suicide to provide a homeostatic environment, and without it would have a proliferation of potentially dangerous cells. The link to apoptosis malfunction and cancer has been proven in many types of malignancies. By understanding its full potential and the unique structure PCD follows, it may demonstrate a better understanding of homeostasis and the eradication of cancerous cells altogether.

The attributes which governed cell death were similar to the attributes which governed cell movement. Our model only allowed for cell death on the third and outer layer to model cells sloughing off the outer surface of the tissue. We will discuss cell death more in the next section in the overview and the description of the function cell death.

4.2.3 Cell Division

Cyclin and cyclin-dependent kinases (CDKs) are the two main classes of molecules that regulate cell cycle progression. The two molecules act together to cause the cell to move through its cycle. The CDKs are activated by a bound cyclin and once activated perform phosphorylation of that activates target proteins and usher the cell into the next phase of its cycle. The main function of cyclin-CDK complexes where the two molecules are stored and are activated is to insure that each piece of the genome is replicated once and only once. At each stage some types of complexes are activated and torn down and other complexes are built up. This cycle of regeneration and activation mimics and promotes the cell cycle and is tied heavily to phosphorylation

of certain proteins due to the action of stable DNA. As DNA transcriptase moves along the DNA it blocks or allows this phosphorylation which leads either build-up or to the activation of the cyclin-CDK complexes respectively.

One of the most important part of the cell cycle for our model is cell division. In order for new cells to be created in a skin tissue, older cell s must divide. Cells are not just created but are instead replicated. The division of a cell is incorporated in the cell cycle. The cell cycle includes interphase and mitosis. Interphase is the process in which a cell replicates its DNA; mitosis is the process by which a cell actually splits in two.

Our group had to incorporate cell division into our model in order to model a dynamic healthy tissue. Cell division in our model takes a cell and creates two replicas that are half the size of the original. A cell will divide if it has high integrin levels and enough room. The newer cell won't be as likely to divide as an older cell, thus we tracked a diffusible promoter attribute which we increased at each iteration of the cell step function.

In our model the attributes which affected cell division were cell space, integrin levels, and promoter levels. The equation governing cell division is .4cell(n).cellspace + .6cell(n).integrin − cell(n).promoter >= .4. If this equation is satisfied then the program creates two identical cells (in every attribute except location) which replace the mother cell. The only cells that can divide in our model are the basal layer, or bottom layer of cells. We take the old location of the mother cell and place the new cells at semi-randomized positions centered around the old location. The cell boundaries of the two cells are computed in the usual way, using the midpoint bwteeen cell locations. One of the new cells recieves $\frac{3}{4}$ of the mother cell's promoter levels while the other cell receives $\frac{1}{4}$ of the mother cell's promoter levels. This was meant to model the idea that one of the two cells will be less likely to remain in the basal layer. A cell will divide if it has high integrin levels, enough room, and low promoter levels.

4.2.4 Cell Growth

Cell growth is a function of cell metabolism. Cells metabolize glucose to build up chemical energy which is stored for processes like cell division and higher order cell functions in differentiated cells. Cells increase their volume by allowing oxygen, CO_2, water, glucose and other compounds to pass through their cell membrane. These processes are explored in Section 3. There is evidence to show that while yeast cells grow exponentially in time, mammalian

cells grow more linearly in time [14].

Our model did not track cell growth continuously, but rather discretely using a linearly growth function. The cells grew linearly at each iteration of cell step.

4.2.5 Cell Movement

Cell receptors called ligands bond with cAMPs and the interaction causes actin and myosin filaments in the cytoskeleton to contract. Because forces are opposed equally and oppositely, this causes the cell to move forward in some direction. Because the cells are oriented towards the squamous cells, and because the chemicals involved in necrotaxis are attractive rather than repulsive, cell death causes migration of a replacement cell towards the cell surface. There is also, in addition to the chemical gradient a pressure gradient. The density of cells causes cell migration and growth towards the surface [3].

We decided to represent cell movement through a specific taxi, called chemotaxis. There are many different taxis that also affect movement including haptotaxis, and necrotaxis. Haptotaxis are directional responses to a gradient of substratum adhesiveness; while necrotaxis is a type of chemotaxis employed specifically to attract white blood cells and some response to cell death from the nervous system [11] [33].

Chemotaxis is not the only cell function which governs cell movement in our model. Integrins and cadherins are also very important cell attributes in the model we designed. Integrins are integral membrane proteins that allow a cell to communicate with the outside world. Integrins bond basal cells to cartilage and bone. Integrins allow information to flow in both directions between the extra-cellular matrix and a cell, making them bidirectional signalers. Cadherins are similar extra-cellular structures, except that they are responsible for cell-cell connections.

Bonds between cells allow for communication between and among cells. There are two bonds that our group dealt with: integrins and cadherins. They are both very similar, but for our model integrins partially control division and cadherins partially control movement between layers. Integrins and cadherins can be considered to be like Velcro on the outside of the cell that allows the cell to attach to the basal membrane. The integrin and cadherin levels can increase and decrease based on whether the cell moves or remains static.

In our model integrins are based on the contact area between basal cells and the surface. The more contact area a cell has, the less likely it is to move. The only cells with integrins in our model are the basal cells; these basal cells are also the only cells that divide in our model.

We applied our research on diffusion and motility of cells to our model in a very fundamental way. We assumed that cell movement was based on chemical and pressure gradients created when a cell dies. The mechanisms by which a cell becomes motile is based on the cadherin and integrin levels. Precisely, our basic movement algorithm is based on the inequality

$$\frac{1}{\text{cell(n).cadherin}} + \text{cell(n).promoter} + \text{cell(n).volume} \geq 1.95.$$

We assumed that the fewer neighbors a cell has the more likely it would be to move. Again, the cadherin level is a linear function of the number of neighbors a cell has. The next factor we consider was the promoter level. We assumed that the higher the diffusible promoter level was the more likely it would be to moce. The final factor we considered was cell volume. We assumed the more volume a cell had, the older it was and thus more likely to move or undergo apoptosis. We describe the relationship between movement, death, and division in more detail in the overview of the next section.

4.3 The Program

Our task for this semester was to program a model of a healthy cell population using MATLAB. We were given a few guidelines as to what we should incorporate in the program: model at least 3 layers of cells, try to incorporate integrins, cadherins, diffusible promoter, and diffusible inhibitors, model cell movement, cell division, cell death, and cell signaling. We will give a brief overview of what our program does and then describe in more detail how each function works.

4.3.1 The Functions

Now that we have given a brief overview of what the program does we will explain in detail what each function does, what assumptions are made, what rules we used, why we did what we did, and any possible errors or unrealistic characteristics of the function.

Cell Structure

Cell Structure is the first function used in our program. The idea of cell structure is to create a population of cells, each containing unique characteristics. Cell structure gives us the population we will use for the rest of the functions.

Cell Structure sets up an array of 30 cells. It creates 3 layers of cells, each layer containing 10 cells. To each cell it assigns:

- x-location, y-location
- left boundary, right boundary
- left neighbor, right neighbor, top neighbors, bottom neighbors
- integrin level, cadherin level, diffusible promoter level, volume
- cell space, actual space

After all these characteristics are stored a plot of the cells is displayed.

Here is an example of using cell structure, type in cell structure into the Matlab window and you will get a picture as follows. Once the cell array is created we can access any individual cell by typing cell(n), where n is the number of the cell you want to access. Below shows us what each cell has stored in it.

We assigned semi-random x-values so our model would be a little more random and realistic. We created boundaries at the midpoint between cells to give us an equal boundary for both cells to grow or move within. Integrins are base level ties from layer 1 to the basal layer. We decided the more actual contact area the cell has with the basal layer the more integrins it should have. The formula for integrin level calculates the exact contact area a cell has with the basal layer. Cadherins are ties between cells and its neighbors, the more neighbors a cell has the more cadherins it is likely to have. The diffusible promoter is a random chemical that makes a cell want to move or die. A cell in layer 1 is not as likely to move or die as a cell in layer 3, so we adjusted levels accordingly. We determined 60% is the minimum amount of space the cell will actually use of actual space. Cell space is the distance between the boundaries, or the area a cell can move or grow within. The actual space formula is the amount of area the cell actually uses.

Each of the values are stored as and determined by the following:

- x-location: a number between .5 and 10.5 with initial values assigned randomly for cell(n) within the interval [n-.5, n+.5] (mod 10)
- y-location: a number, 1,2, or 3 depending on the layer the cell is in, there are 10 in each layer
- left boundary and right boundary: half the distance between the x-location of its left and right neighbor respectively
- left, right, top, and bottom neighbors: any cell who's boundary is touching the boundary of the cell is a neighbor of the cell, left is the closest cell to the left, right is the closest cell to the right
- integrin level: 0 if in layer 2 or 3, determined by the formula cell(n).integrin = cell(n).volume * cell(n).cellspace if in layer 1
- cadherin level: the number of neighbors a cell has
- diffusible promoter level: layer 1- a random number between 0 and .4, layer 2- a random number between .25 and .75, layer 3- a random number between .5 and 1
- volume: a random number between .6 and 1
- cell space: cell(n).rightboundary - cell(n).leftboundary
- actual space: cell(n).volume*cell(n).cellspace

It would be nice if we could find some research to validate the levels that we assigned. We made a guess at what levels seemed reasonable, but that does not mean they are accurate. For example our measurement of integrins and cadherins has almost nothing to do with the chemicals themselves, but rather what affects these chemical levels. The cell boundary attribute might be improved by considering volume and some other factors. We determined a lot of these levels by running the program several times and seeing the results. If a factor was not acting like we wanted it to we adjusted it.

Cell Adjust

Cell adjust is the most used function. It is called on by cell structure, cell death, cell move, and cell division. It is called after every change in the population. The purpose of cell adjust is to update the population and plot the population.

Cell adjust redefines each cells' boundaries, neighbors, and updates its cadherin levels, cell space, and actual space. After these levels are updated a plot of the population appears.

After each change in the population we need to make sure all the values stored are accurate. This function updates any of the items that might have changed.

Cell adjust calls cell boundary to update the boundary of each cell. It calls topbottomneighbor to determine the new neighbors of the cell. Cell adjust sets the cadherin level to the number of neighbors the cell has. It sets the cell space as the distance between the right and left boundaries of the cell. It calculates the actual space by multiplying the volume of the cell times the cell space. After it has done this it calls cell plot to plot the changed population.

Once again we are not very happy with our definition of cadherins. It would be nice to be able to determine cadherins a different way than by counting the number of neighbors the cell has. It would also be nice to determine cell boundaries using a different and more accurate method.

Cell Boundary

Each cell has an area it can move, grow, or divide in. The purpose of cell boundary is to determine this space.

We needed to define some type of boundary for each cell so it only has a finite space to work with. Using the midpoint is the simplest way to incorporate this.

If the cell is on the far left then its left boundary is .5. If the cell is on the far right then its right boundary is 10.5. Otherwise the boundary is determined by the midpoint between the cell and its left or right neighbor using their x-locations.

Cell boundary determines a left and right boundary for each cell.

We think this definition is the "easy way out". If we had more time we would try to incorporate things like volume in determining where to put the

boundaries between each cell. Even though it possibly is not the best way to define the boundaries it is effective and gives us the results we wanted.

Top and Bottom neighbor

The purpose of the topbottomneighbor function is to determine the top and bottom neighbors of each cell.

Assigns top and bottom neighbors for each cell.

One of the things we needed to model was cell communication. If two cells overlap, or are neighbors, then they can communicate. This function determines the neighbors so we can determine who it communicates with.

Uses subroutines to determine if the cell boundaries overlap at all. If they do, it adds the neighbor to the list of neighbors.

This function is pretty straight forward. The only problem with it is it uses cell boundaries overlapping, not the actual cell space. Maybe two cells boundaries overlap but possibly the actual cell space does not and therefor they could not communicate. We did not incorporate this in the model.

Cell Plot

The purpose of cell plot is to plot a graphic picture of our population so we can investigate visually what is happening.

Cell plot plots a visual representation of our population.

It is nice to be able to see the population and how the cells are reacting. With a visual representation of the population we can determine many things including the validity of our model.

Cell plot plots a line at y=.5, y=1.5, y=2.5, and y=3.5 to represent the different layers. It plots a line at x=.5 and x=10.5 to represent the sides of the population. It plots the location of each cell and the cell number for each cell, then it plots the left and right boundary for each cell.

This function is straight forward and produces the graph desired. It might be nice to incorporate some GUI to the plot so the user could click on a cell and see the properties of it without having to type in the cell number in the Matlab command window.

Cell Step

Cell step is the heart of this program. A user will start the program by setting up a population with cellstructure, after that the user will use cell step see

how the model reacts over time.

Once the user calls cell step the program asks how many steps do you want to do. After they enter the number the program will complete that many steps.

A "cell step" has cells perform the following operations:

1. Some cells die in Layer 3 (the exterior layer).
2. Some cells move from layer 2 to layer 3 to replace the dead cells.
3. Some cells move from layer 1 to layer 2 to replace the missing cells in layer 2.
4. Some cells divide to replace the missing cells in layer 1.
5. Volume and Diffusible promoter grow in each cell.

Once a cell dies, moves, or divides there is a new plot showing this and a line appears saying which cell died, moved, or divided.

Here is an example of the calls and the outputs of the program. We first call cellstructure.

Now we call cellstep and do 1 step. Here are the results:

Here are some plots of some populations after 10, 20, 50, and 100 steps:

This as a normal cycle for a healthy tissue to go through: something dies, things move up to replace it, and then cells divide. With each step a cell is alive, it grows and transforms. The last step of a cell step represents the growth in both the volume of the cell and the diffusible promoter (or desire to move or die). The rate at which it grows is according to its previous level and a constant which varies with the step number. We assumed the rate of growth to be exponential and picked constants according to what made our model stay in equilibrium (the number of cells remained close to constant).

Cell step calls cell death, then cell move, then cell divide. After it has run through all of these programs it updates the volume and diffusible promoter level in each cell as follows:

- Layer 1
  - cell(n).volume= cell(n).volume$*e^{.009*j}$ where j is the step number

  - cell(n).promoter= cell(n).promoter$*e^{.003*j}$ where j is the step number

- Layer 2
  - cell(n).volume= cell(n).volume$*e^{.006*j}$ where j is the step number
  - cell(n).promoter= cell(n).promoter$*e^{.006*j}$ where j is the step number

- Layer 3
  - cell(n).volume= cell(n).volume$*e^{.003*j}$ where j is the step number
  - cell(n).promoter= cell(n).promoter$*e^{.009*j}$ where j is the step number

Determining the growth constants was a trial and error process based on running the program. It would be nice to delve more deeply into how growth rates are determined (biologically) and see how those growth rates match up with our model.

Cell Death

The first process of a cell step is the death of cells in the third layer. The purpose of this function is to determine when a cell dies and then zero out or delete the cell.

Cell death is a function that determines when a cell dies. If a cell meets the requirements then the cell dies. Once a cell dies all of its entries are zeroed out, the left and right neighbor absorb the space of the dead cell. We reasoned that the vacuum created by the flaccid dead cell would be filled by its neighbors. Each one of the dead cell's neighbors receive a portion of its diffusible promoters (using cell signaling). A plot shows which cell is going to die and then shows the population after the cell dies. A line appears saying which cell died.

Here is an example of what the plot looks like before and after a cell dies.

A cell will die if it does not have too many neighbors connecting it to the model, if it has a high diffusible promoter level, and if it has a high volume. We put these factors together knowing the maximum sum is 2.5 so we adjusted it to a level where we saw cells dying at approximately the rate

we thought they should. We researched the concept of chemotaxis [47] and determined once a cell dies it signals its neighboring cells to get them to move, that is modeled with our concept of distributing the diffusible promoters.

A cell will die if 1/cell(n).cadherin + cell(n).promoter + cell(n).volume >= 1.95. Once a cell dies all of its entries are zeroed out and its old left and right neighbors absorb the space (we adjust the volume accordingly). If the cell had neighbors on all 3 sides (left, right, and bottom), then each side will receive 1/3 of the diffusible promoters (divided evenly on bottom) else the one side neighbor will receive half and the bottom neighbors will divide half. All the counters are updated and the new cell population is plotted.

We do not know exactly why a cell dies. It would be nice to have more research about exactly why a cell dies and get some realistic levels.

Cell Move

After cells die cells from the second layer need to move to the third layer to replace the dead cells and cells from the first layer need to move to replace the cells in the second layer. The purpose of cell move is to determine which cells move and actually move them.

If a cell meets the criteria outlined below, the cell will move up a layer. All the cells above it will shift to allow the new cell to move up and the old neighbors will absorb the space where the cell was. A plot will show which cell is moving and then show a plot of the modified population.

Here is an example of a few cells that moved. A cell moved from layer 2 to layer 3 and then a cell replaced that cell by moving from layer 1 to layer 2.

A cell is likely to move if it does not have ties to too many neighbors, it has a high promoter level, and a high volume. A cell cannot move if it does not fit so that is what the second check is for. Once a cell has moved we need to adjust the surrounding cells and that is what the program does. We divided the extra space equally.

A cell will move from layer 2 to layer 3 if 1/cell(n).cadherin + cell(n).promoter + cell(n).volume >= 1.25 and if the sum of the cells top neighbors actual space + cell(n)'s actual space fits within the boundaries of the cells top neighbors. A cell will move from layer 1 to layer 2 if 1/cell(n).cadherin + cell(n).promoter + cell(n).volume >= .8 and if the sum of the cells top neighbors actual space + cell(n)'s actual space fits within the boundaries of the cells top neighbors. When a cell moves its old neighbors will absorb the space

and we adjust the volumes proportionally. We determine whether cell(n) is closer to the left or the right boundary of its top neighbors and move it to the closest area. We take all the extra area within the left and right boundary and spread it evenly between each cell. We adjust the neighbors, boundaries, and volume for each cell involved.

Although we do have the key characteristics as to why a cell would move we are not sure if our model is that accurate. We interpreted levels for each aspect of the function logically at first and then adjusted them a little until we had something that we believed was accurate. It would be nice to have more concrete research on exactly what affects cell movement. Another problem with this function is when a cell moves up a layer it can effect neighboring cells boundaries because the way our boundaries are defined. It does not effect it by much but if it changes then there should be changes made to that cell (such as volume) which we do not consider.

Cell Divide

An important part of a healthy population model is having regular cell division to replace the cells which have moved and/or died. The purpose of this function is to model cell division.

If a cell meets the criteria it will divide. A new cell is created which is a replica of the previous cell and they split the area of the original cell. A plot shows which cell is going to divide and then what the model looks like after the cell divided.

Here is an example of what it looks like when a cell divides.

A cell will divide if it has high integrin levels, enough room, and the lack of desire to move. The newer cell won't have as much desire to move as the old cell so we adjusted the promoter levels accordingly.

If .4*cell(n).cellspace + .6*cell(n).integrin -cell(n).promoter >= .4 then a cell divides. A new cell is created that is identical to cell(n). We take the old boundary of cell(n), divide it into two, and place each cell in the middle of their half. Cell(n) receives 3/4 of the promoter level while the new cell only receives 1/4. The cells' volumes are divided in half for each cell.

There is not much as far as population modeling as to when and why a cell divides. If we had time we would incorporate the cell model's program to more accurately depict division. We are fairly certain that all of our elements effect cell division, but once again are not sure at the rate for which

they should effect it or if there are any other elements we need to consider. Another problem with this program is when a cell divides, it slightly affects its neighboring cells' cell boundary which would change the volume of the cell.

4.4 Conclusion

Our model is a fairly accurate depiction of how cells signal, move, divide, growand die in a tissue population. The biggest limitation of our model is the small number of attributes recorder at each cell. The fundamental aspects of how the cell cycle progresses and the mechanisms by which the tissue population remains ate equilibrium were well accounted for, though not neccessarily biologically accurate. The program needs to be run and rerun so that atypical cell formations can be explained and functions can be designed which will lead to such atypical formations being eliminated.

We set out to model healthy tissue so that we might model cancer and in turn develop tools for early diagnosis of cancer. Only once we are convinced that we properly understand the mechanisms by which a tissue population holds itself in homeostasis can we interpret and explain the nature of disruptions in this equilibrium. Because our model does not account for wounds which can sometimes lead to cancer, we can only can to have accurately depicted a healthy tissue population.

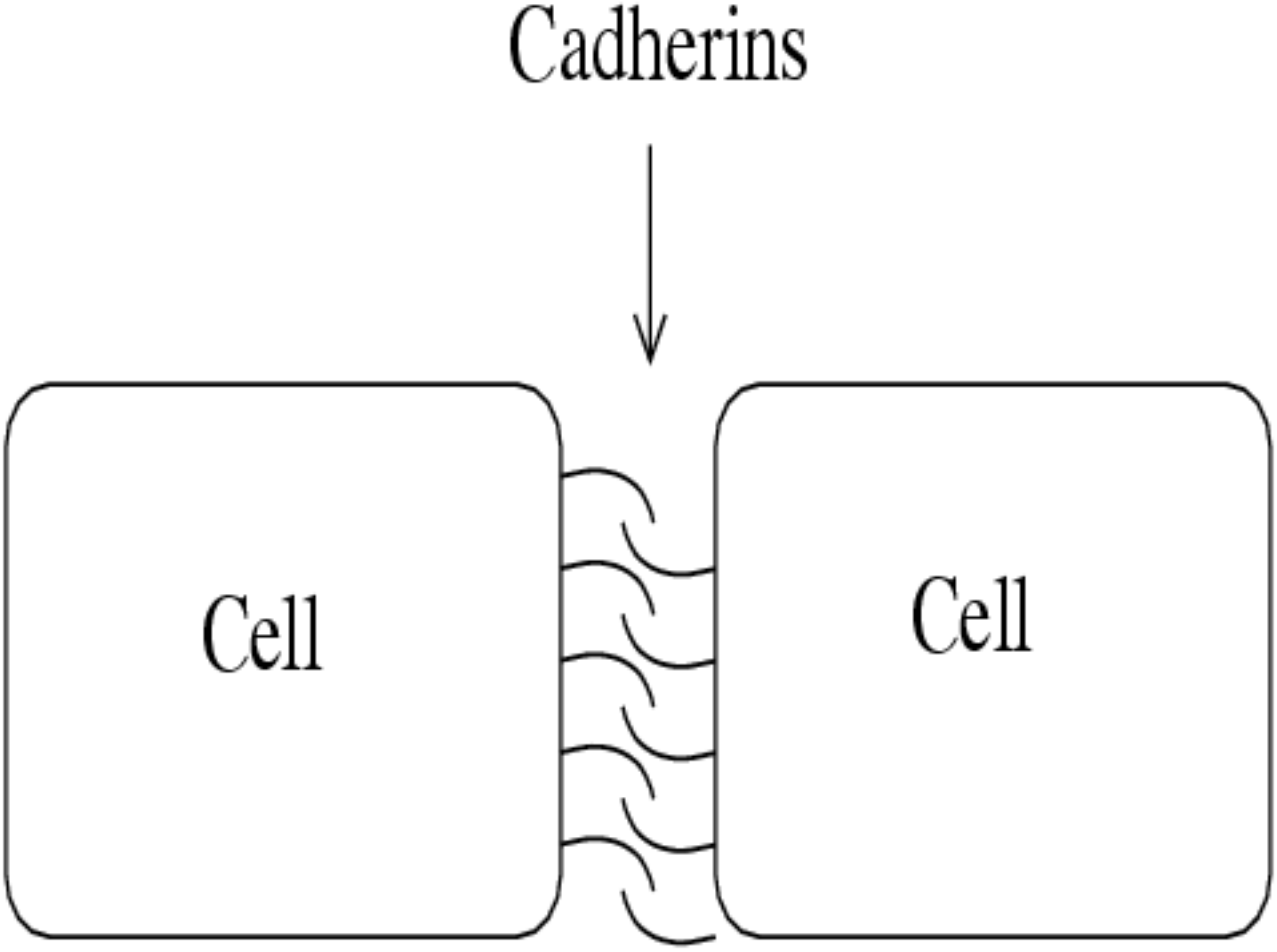


Figure 2.4: An example of how cadherins bond together like Velcro.

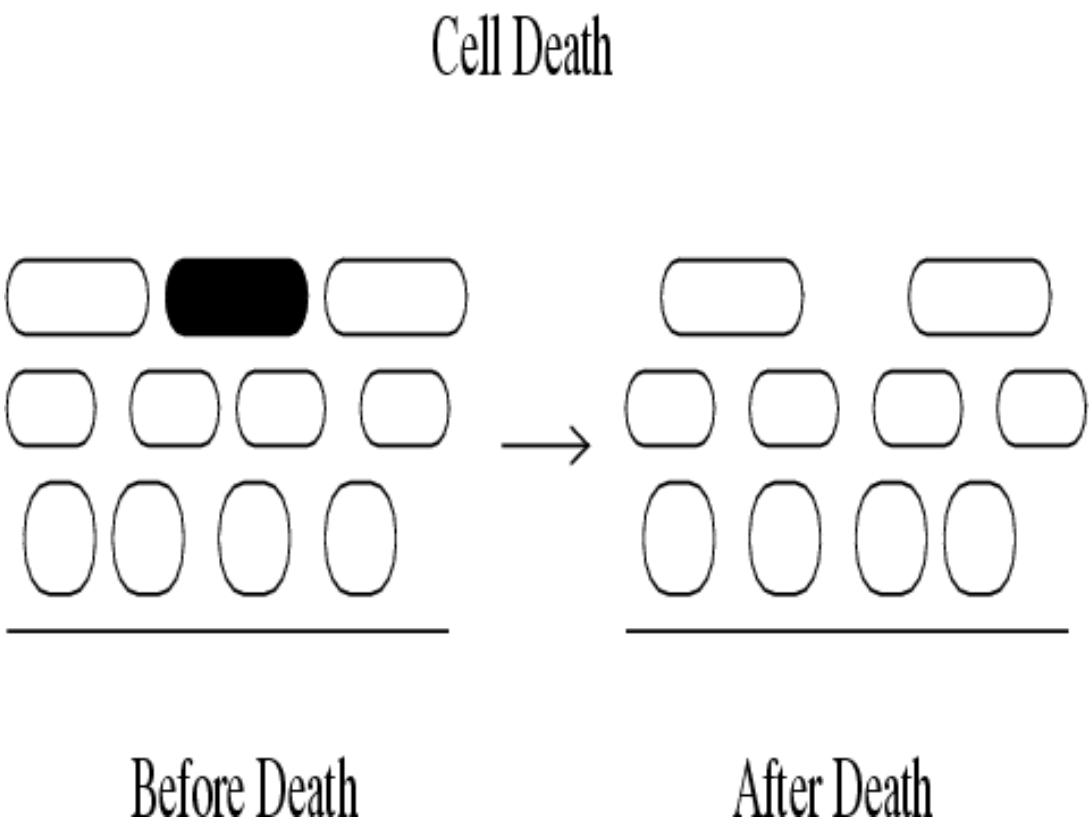


Figure 2.5: When cells die, they lose volume.

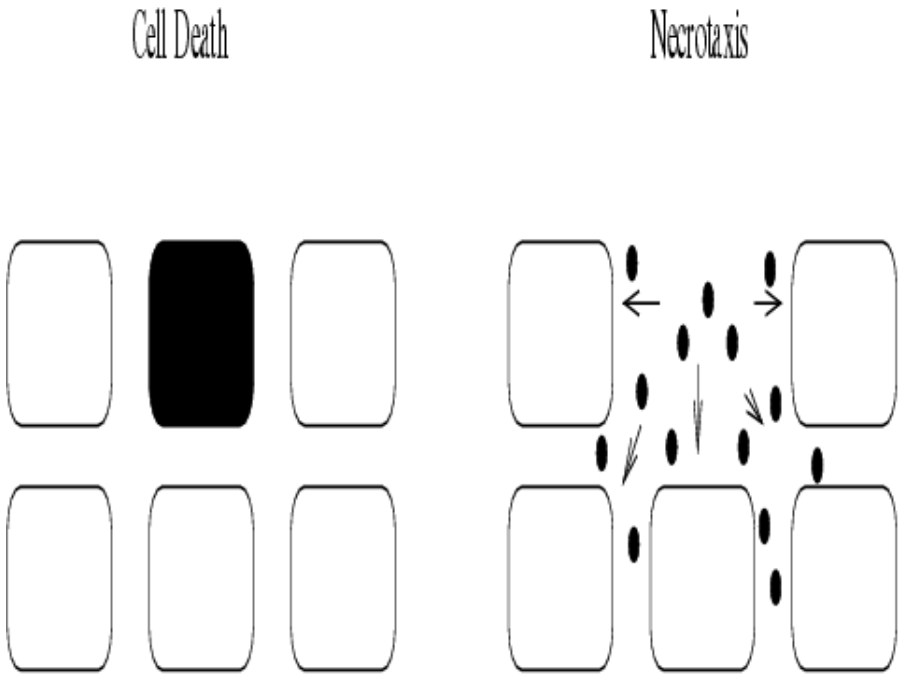


Figure 2.6: An example of chemotaxis that promotes cell movement

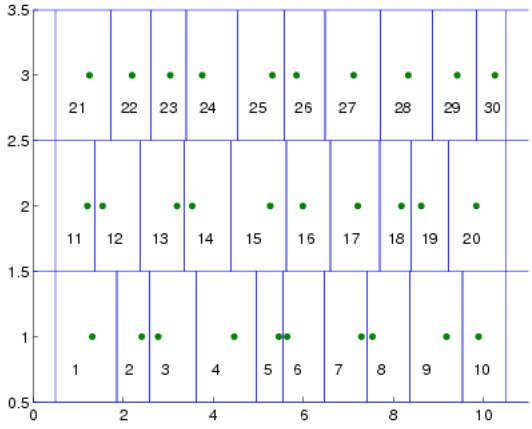


Figure 3.7: An initial population of cells

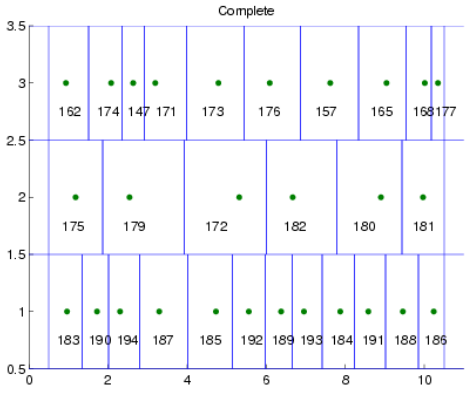


Figure 3.8: A population of cells after 100 steps

```
>> cellstructure
>> cell(1)
ans =
xlocation: 0.8477
ylocation: 1
leftneighbor: []
rightneighbor: 2
promoter: 0.0600
volume: 0.8344
integrin: 0.6717
leftboundary: 0.5000
rightboundary: 1.3049
topneighbor: 11
bottomneighbor: []
cadherin: 2
cellspace: 0.8049
actualspace: 0.6717
>>
```

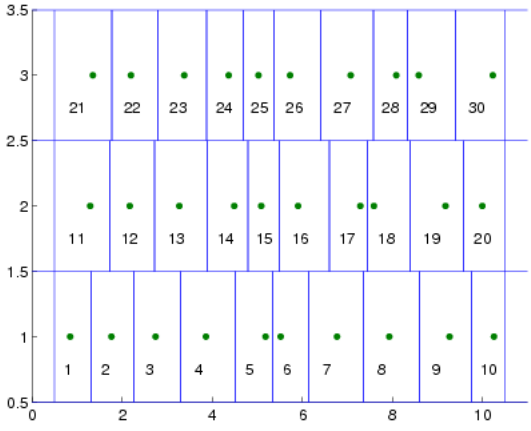


Figure 3.9: A set of characteristics for a typical cell

>> cellstructure

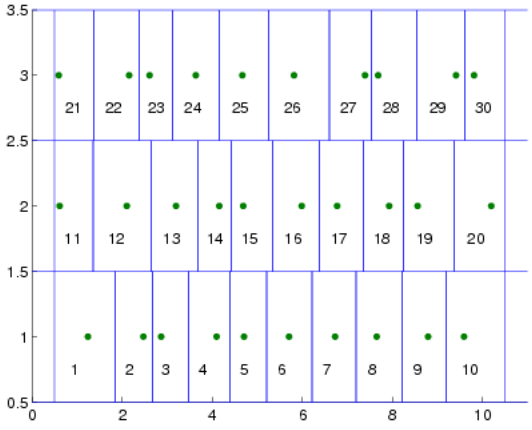


Figure 3.10: Cell Replacement

>>
[cell,a,b,c,k]=cellstep(cell,a,b,c,k);
How many steps do you want to go through? 1
cell step 1
cell 22 died
cell 12 moved
cell 2 moved
cell 3 moved
cell 1 divided
cell 4 divided
cell 5 divided
cell 6 divided
cell 7 divided
cell 9 divided
cell 10 divided
>>

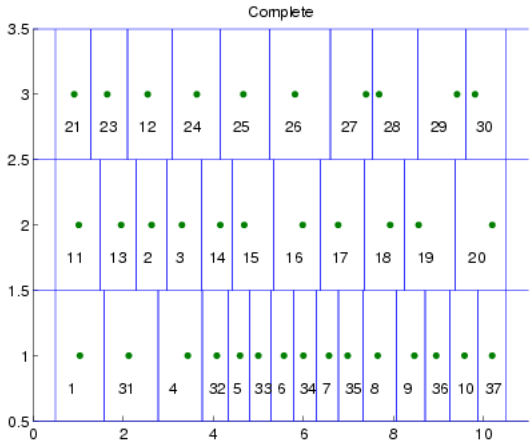


Figure 3.11: A list of cell movement and death

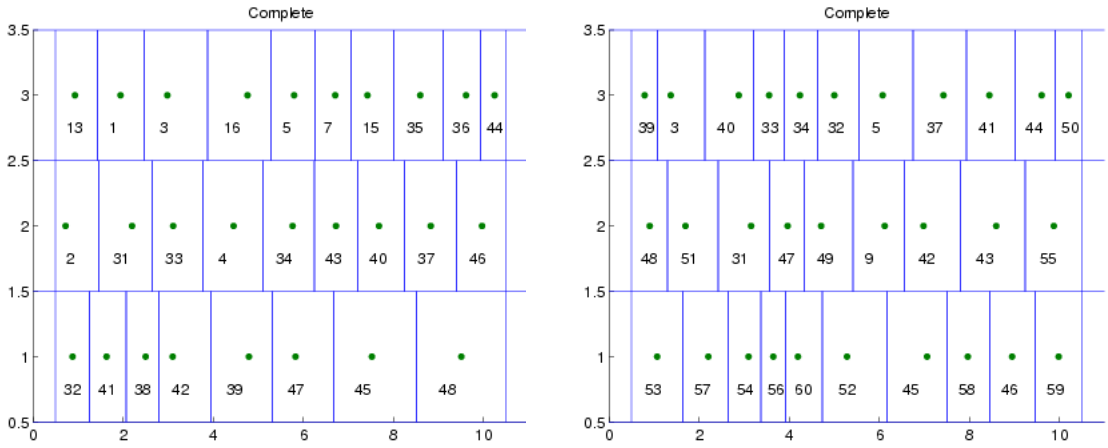


Figure 3.12: A population after 10 and 20 steps

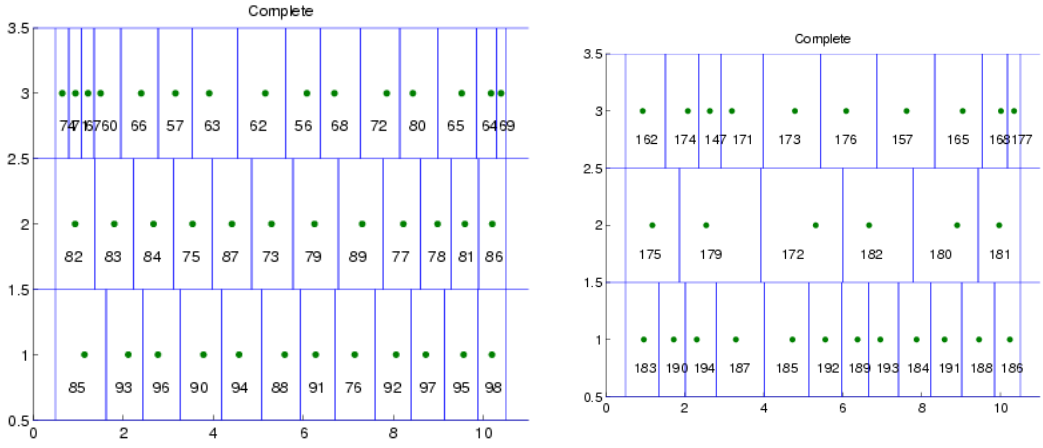


Figure 3.13: A population after 50 and 100 steps

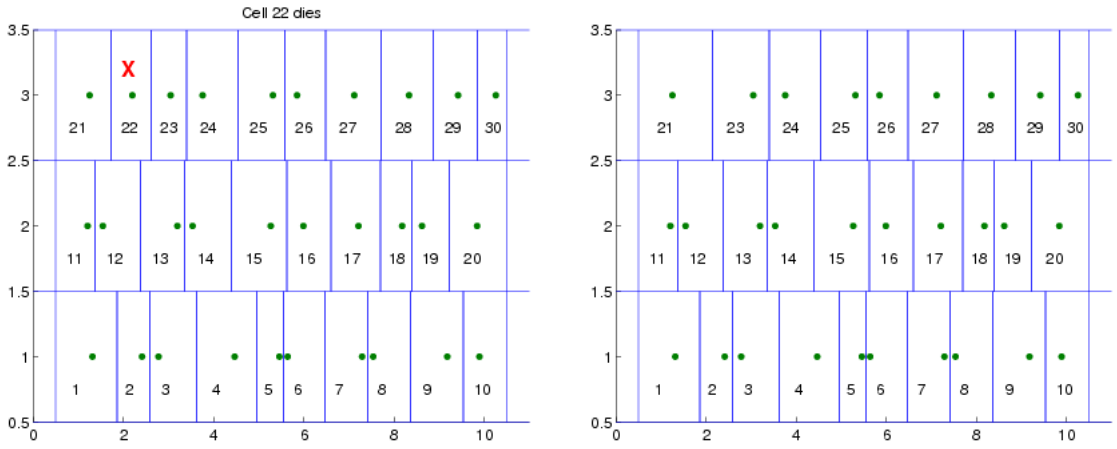


Figure 3.14: Before and after the cell dies

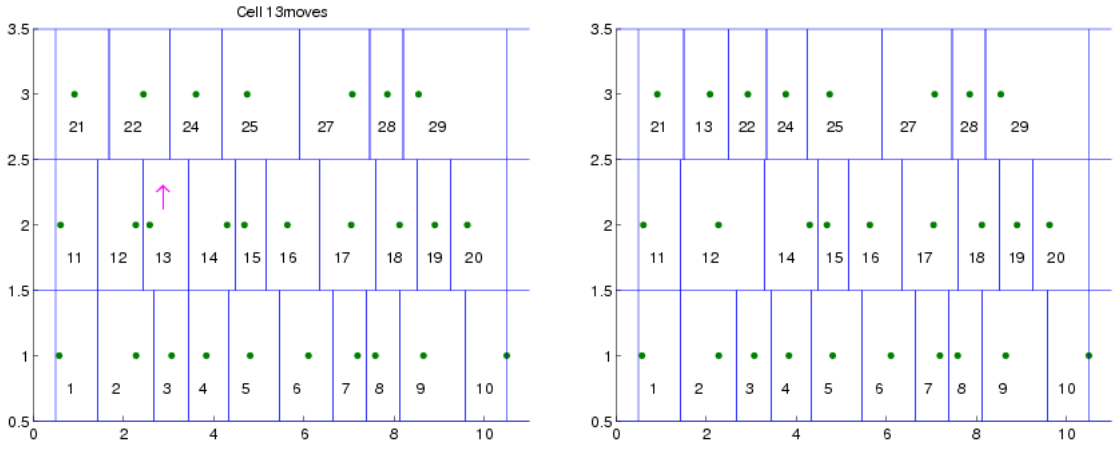


Figure 3.15: Before and after the cell moves from 2 to 3

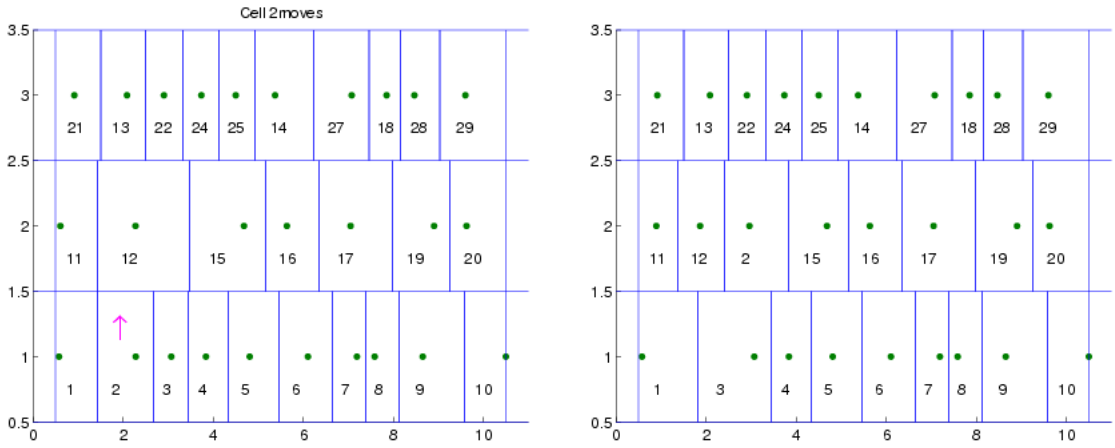


Figure 3.16: Before and after the cell moves from 1 to 2

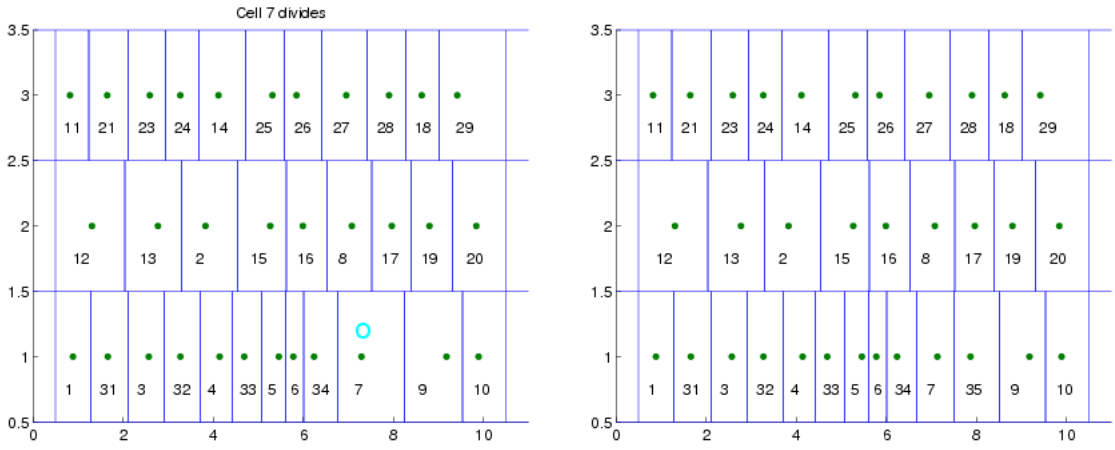


Figure 3.17: Before and After the cell divides

Chapter 5

Cell Population Model II: Tissue Modeling

By Chris Comiskey, Don McCuan, and Kim Story

5.1 Introduction

This report will explain in detail the model we created. The model attempts to represent epithelial tissue dynamics. It seeks to incorporate the important factors, while simulating the main processes that occur within epithelial tissue.
Our efforts are part of the larger objective of developing computational models to guide research in the area of epithelial cancer. It is hoped that structural changes in the tissue can be understood and thereby offer early cancer detection. Our project is important because it contributes to the growing body of knowledge available to researchers looking for direction in the lab environment. Even further, it contributes to future groups attempts to do exactly what we attempted to do. In other words, they can use the model we developed to pick up where we left off.
Just as the bulk of our efforts were aimed at developing this model, the bulk of the report will aim to explain this model. The explanation will be broken into three main parts: the biology of epithelial tissue, a detailed description of the model, and the code itself. The relevant biology will be integrated with the description of the model, where possible. Additional information gathered but not included in the code will also be provided,

and integrated where possible. A concluding section will summarize the report and its relevance to the overall researching efforts being made.

5.2 The Model

Our model attempts to simulate healthy epithelial tissue, in particular the tissue of the mouth and esophagus.

5.2.1 Overview and Motivation

Epithelial tissue separates the interior of the body from the exterior world. As such it includes in part the skin, lining of the mouth, esophagus, stomach, intestines, and respiratory track.
Epithelia are classified as **simple** (one cell thick) or **compound** (more than one cell thick). If the shape of the cells in the outmost layer are flattened and thin they are called **squamous**. If approximately square they are **cuboidal**. If taller than wide they are **columnar** [59]pg.52.
In our model we are concerned with compound squamous epithelia.
The tissue geometry in the model will be represented in two-dimensions (for simplicity) and consist of 3 cell layers, restricted to a finite width but allowing for layer thickness to vary within defined limits. Cells in layer 1 are generally cuboidal, in layer 2 are more flattened, and in layer 3 even more flattened; by which we mean that the height to width ratio of cells becomes smaller as we move up in the layers.

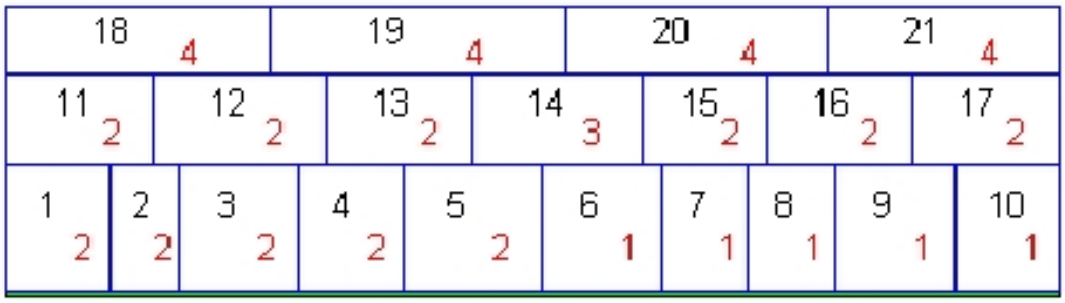


Figure 2.1: Tissue Geometry

While the model (for simplicity) is two dimensional. We may refer for

clarity to cell volume and area of contact as if the model were 3 dimensional.
The function of epithelial tissue is to protect the internal environment of the body from the external environment while at the same time allowing for passage of selected chemicals both ways.
Epithelia in general are composed only of cells with little or no extracellular space, fibres, or blood-vessels. This forces the exchange of chemicals between exterior and interior environment to go through the cells and be regulated by cellular control mechanisms. In other words the epithelial tissue is very selective about what gets in and what gets out [59]pg.43. Our model reflects this in consisting only of cells with no extracellular space.
The bottom layer of cells in the tissue attach on their basal surface to a carpet of fibres called a **basal lamina**. This basal lamina acts as physical support to the epithelium. While not rigid it is not fully elastic. [59]pg.44. We model this and simplify our model by limiting the number of cells along the basal lamina to be very limited (only 10 to 20) within this limit we considered the basal lamina to be rigid and flat. While the epithelial cells can allowed to slide along the basal lamina they cannot penetrate or deform it.
Cells in the tissue in general age, die, grow, divide, move, and differentiate. These actions are modeled by specific rules to be described later.
Not all cells can divide in the epithelia. Cells closer to the hostile external environment are more subject to cellular damage and as such generally do not divide [59]pg.46.
We model this by allowing for different cell types. Some allow for cell division and are restricted to the lower layers and some do not and they appear in the outer layers.
Because cells die and in particular are shed (rubbed off) from the top layer of the tissue, we must have a movement from the lower layers where cells are created to the upper layers where death is more frequent.
The model will execute multiple cycles in time. Within each of these cycles cell aging, death, growth, division, differentiation, and movement will occur. Each of these actions will be governed by rules, some of which will depend on parameters of the model.

5.2.2 Cell Types

The cells in the model will be of 6 types: Stem Cells, Transit amplifying cells, Layer two (non-dividing) cells, squamous cells, shed cells, and dead cells.
Stem cells can divide, have very long lives, and are restricted to layer 1. Stem cells divide to create either new stem cells or transit amplifying cells. Transit amplifying cells are shorter lived, can divide, and can exist in layers 1 or 2. Cell division creates either transit amplifying cells or "Layer two" cells.
"Layer two" cells are non-dividing cells which occur generally in layer 2 but are allowed to be created in layer 1 from which they will move up to layer 2. Squamous cells are the highly flattened cells in layer 3, are non-dividing. Cells are removed from the tissue either by dying or being shed from layer 3. To distinguish how they were removed from the tissue they are assigned a cell type of either "dead" or "shed" when this occurs.

5.2.3 Cell State

Each cell in the model is identified by a unique number and has a cell state which consists of

- layer a cell is in
- sequence within the layer the cells is at
- cell size
- cell type
- cell age
- factor x
- the cell that created this one

The cell states of all of the cells completely determine the current geometry of the model tissue. Cell state and the model rules determine how the model will evolve in time.

5.2.4 Output

The layer, sequence, and size of all of the cells uniquely determine the tissue geometry and the implementation of the model in Matlab will display

this. This display at various points in the cycles is the primary form of output for the model.
For example, figure 1 shows a possible initial state for the model. Within each cell the cell number and cell type are displayed.

5.2.5 Cell Aging

Cell age is the number of model cycles occurring since cell creation. Cell age is incremented by one at the start of each cell cycle.

5.2.6 Cell Death

Cell death occurs after a set age that is dependent on cell type. Upon cell death cell type changes to "dead" and that cell is no long included in cell geometry or any cell actions.

5.2.7 Cell Growth

Cell growth occurs at a fixed rate limited by a maximum cell size that is dependent on cell type.

5.2.8 Cell Division

Only Stem cells and transit amplifying cells can divide. Cell division occurs when a cell reaches it's maximum size. Cell volume is divided equally between parent and daughter cell. Daughter cell is assigned a new number, is inserted into the tissue to the right of the parent, and is assigned a cell type based on parent cell type and a probability distribution.

5.2.9 Cell Differentiation

Cell differentiation is the change in a cells type. This occurs when cells move from one layer to another. Stem cells are normally in layer 1 and become transit amplifying cells if they move to layer 2. Transit amplifying cells may become nondividing layer 2 cells when moving to layer 2 or may stay transit amplifying based on a probability distribution. All cells in layer 3 are squamous cells.

5.2.10 Cell Movement

Movement is by far the most complicated aspect of the model. To do this we create an abstraction called factor x. Factor x controls cell movement. Factor x is a hypothetical combination of one or more chemicals. It is motivated by the following. Epithelial cells are tightly bound together. In order for them to move something must trigger a set of events resulting in the temporary breaking of these intra cellular bonds.
A cell can move by either forcing itself between two cells in the level above or by replacing a cell above it and pushing that cell into the level above (possibly causing it to shed).
In the first case, the factor x level per cell area in the cell to move must be high enough (exceed some set value, say X2) to break all of its intra cellular bonds. Also the factor x level per cell area of the cells to move between must be high enough (exceed some smaller value,say X1) to break the bonds between those two cells. That is these cells don't require quite as much factor x since we aren't breaking all of their intra cellular bonds.
In the second case, since both cells must move they both must have the higher level of factor x and we have a cascading sequence of events since the cells above the cell to be pushed up must have sufficient factor x to allow this.
The models accounts for creation, diffusion, and destruction of factor x. Factor x is split equally between cells during cell division.
Every cell has an ideal shape whose measure is its height to width (h/w) ratio. Factor x is produced in response to the amount of cell distortion from this ideal shape. We model growth in factor x within a cell as being proportional to the deviation from ideal shape times the volume of the cell. The proportionality factor is a function of cell type.
We allow for factor x to diffuse between cells. We think this is necessary since cell movement is a cooperative process between cells, so we expect the "stress" on a cell to move to affect the cells around it. This diffusion is modeled to be proportional to the product of the difference in factor x concentration between cells times the area of contact between cells. Based on the biology we expect diffusion within a layer to be greater than diffusion across layers and so will have different proportionality constants for these two cases.
The process of breaking of bonds and moving consumes all of the factor x in a cell. Moving between cells consumes half of the factor x in the cells

moved between.

5.3 Implementation of the Model

The model is implemented using Matlab. It consists of a m-file with one primary function containing numerous sub functions. This architecture was chosen for simplicity and allows the primary data arrays of the model "Cell" and "Type" to be accessible by all of the functions without having to pass them as parameters.
The model is highly parameterized with respect to data that define the rules. Array "Type" is used to contain type Cell type dependent information. Type specifies whether a cell can divide and how, its maximum size, growth rate, life span, factor x growth rate, and ideal shape. Several other rule control variables that are not type specific are coded for. "Cell" is an array that contains the cell state for each cell. This data changes constantly as the tissue evolves from cycle to cycle.
Both "Type" and "Cell" are setup when the tissue model starts execution. This simplest way to execute the model is to open it in edit mode and run it. The model was written with the expectation of adding a GUI interface at some point (not done for this version). This would mean the rule parameters could be adjusted via GUI controls.
The setup for "Cell" defines the initial state of the model. The variable "net" define the initial number of cells in layers 1,2, and 3 and the rest of the information is assigned randomly.
The number of cycles to perform is specified and an options suppress display, display after each cycle, or display after every action is provided. Then model then executes the functions for the tissue actions in succession for the specified number of cycles. These functions are f_age, f_death, f_grow, f_divide, f_produce_x, f_diffuse_x, and f_move_x.
These functions are generally very simple with the exception of f_diffuse_x and f_move_x.
f_diffuse_x allows for different diffusion rates within a layer and across layers. In both cases the amount of diffusion within between 2 cells in one cycle depends on the amount of contact area between the cells. For the case where the cells are in different layers this must be handled by a comparison of left/right cell wall positions.
f_move_x also faces this problem of computing contact area and has to

handle multiple movement scenarios depending on whether we can move a cell via pushing a cell above up or pushing between two cells above. Note in particular the usage of routines f_get_geometry to obtain cell positions and f_adj_seq to recompute cell layer sequences after movement.

5.4 Biology of Integrins

Primarily, these molecules bind the basal layer cells to the basal lamina. The basal lamina, also sometimes called the basement membrane, is the wall upon which the structure of epithelial cells are built. Less prominently, integrins also act to bind cells to other cells.

An important facet of integrin function lies in how they operate. Integrins only bind when they are in concentration at certain focal points [38]. In other words, while still "sticky" when diffusely distributed over a cell surface, actual adhesion will not occur unless the integrins sufficiently cluster together on the cell surface [38]. This adhesion strategy seems to allow cell movement without completely breaking a cell's bond to its immediate environment. In this way integrins seem to allow a cell to be neither completely attached, nor completely detached from the surrounding environment. They are strong, flexible, and adaptable molecules that are very effective in achieving their goals.

The primary mode in which integrins will be incorporated into the model will be through cellular adhesions to the basement membrane. While we know integrins play a role in cell to cell adhesions, these ideas will not be addressed at this time. Given that only basal layer cells rest against the basement membrane, it is only these cells that will be given integrin based characteristics.

Our model contains two types of basal layer cells: stem cells, and transit amplifying cells. Each cell type will be assigned an integrin level at the beginning of the model. Along with the other characteristics loaded into the "Cell" array and called by the Type Array structure, now "v7" will be included. The variable "v7" will denote a given cell's integrin level. This level is going to influence a cell's likelihood of movement from layer one to layer two.

Research indicates that stem cells are less likely to advance to layer two than their sister transit amplifying cells, so we would expect them to have higher levels of integrins. Consequently, we'll probabilistically assign a

slightly higher level of integrins to stem cells than TA cells. The v7 array will contain rules for assigning these levels. For cell group 1, stem cells, we'll assign a random number between .40 and 1.00. For Cell group 2, TA cells, we'll assign a random number between 0.00 and .59. Cell groups 3, 4, 5, and 6 will get integrin levels of 0.00.

For groups one and two we'll want to establish a relationship between this semi-random level and that particular cell's surface area. It is inconclusive what this exact relationship would be, but based on intuition we'll assume a larger surface area means more integrins, thus more stickiness.

In this fashion, each cell will have an integrin level between 0.00 and 1.00. Using this parameter, now in tandem with the other movement determining variables, the cell will make a movement decision. In the model's earliest incarnations, the height to width ratio determined movement. Now the cell would have to pass the test on both the h/w ratio requirement, and the integrin level requirement. This integrin test would consist theoretically, of three categories of movement likelihood. An integrin level from 0.00 to 0.39 would confer a high likelihood of movement. A level from 0.40 to 0.69 would allow movement, but not encourage it. A level from 0.70 to 1.00 would positively prevent movement.

In conjunction with a cell's h/w ratio (or other determining characteristic), a decision would be made by the cell about its movement. Here we would probably want to establish categories of likelihood for the other determining factors so they would be able to interact probabilistically. We model fluctuations in integrin levels from cycle to cycle with an unobtrusive set of fluctuations, perhaps plus or minus 0.10 per cycle, each half the time. Or perhaps stem cell integrin levels would go up by 0.10 slightly more than half the time, discouraging movement, while the integrin levels of TA cells decrease by 0.10 slightly more than half of the time, encouraging movement. Division also needs to be addressed. In its present state, the model allows horizontal and vertical division. Quite simply, integrin levels would be divided equally between two cells resulting from a vertical division, and the bottom cell would retain all of its integrins in the even of a horizontal division.

The literature indicates that integrins play a role in signal transduction. For instance: "The integrins also act as cellular sensor and signaling molecules." [48] This idea of integrins as a signaling molecule abounds in the literature. However, the details are consistently fuzzy. No thorough description of this part of integrin's function was to be found. This is an

area future groups could explore further. Another prevalent function of integrins not incorporated into the programming ideas described is their role in cell to cell adhesion. While we know cadherins act heavily in this fashion, it is clear integrins contribute. What is not clear is how, and to what extent. Once again, future research could be usefully devoted to this topic. Another unknown is what drives the integrin fluctuations between cell cycles? In our model, as described, a rather arbitrary mechanism would govern this activity. Other questions include: do the strength of adhesions (ie their stickiness) determine movement, or is the stickiness a function of cell signaling meant to carry out a "decision" the cell has already made. In other words, which came first: differentiation or stickiness. Does the stickiness inform the decision, or does the decision inform the stickiness?

5.5 Biology of Cadherins

A specific focus for our project, as recommended by Randy Tagg, was the function and influence of cadherins. A basic definition of cadherins is that they are a chain of proteins connected together for the purpose of joining the body's cells together. More specifically, they "provide the glue that gives form to our different tissues" [26]. A cadherin reaches from the surface of a cell and comes into contact with a cadherin coming from the surface of another cell. The structure of a cadherin is that of a chain of proteins. The cadherin molecule is comprised of three major regions [84]. These are; an extracellular region that controls specific adhesion, a transmembrane domain that spans the cell membrane, and a cytoplasmic domain that extends into the cell. The extracellular area is of particular importance. This is the region that contains the adhesive properties. As a visual, the appearance of two adjacent cells could be stated as that of a zipper. Each cell could be represented as having strings coming out of it towards another cell with its own strings. The strings from these two cells then lace together to form a bond.
There are some basic facts about cadherins that should be noted. There are 4 different basic types of cadherins. The focus of this project focuses on the E-cadherin or ones found in epithelial tissue. The cadherin type is important because of the fact that cadherins functioning properly will only join to those of the same cadherin type. This results in cell clusters. Another important fact is that they are dependent on Calcium. The

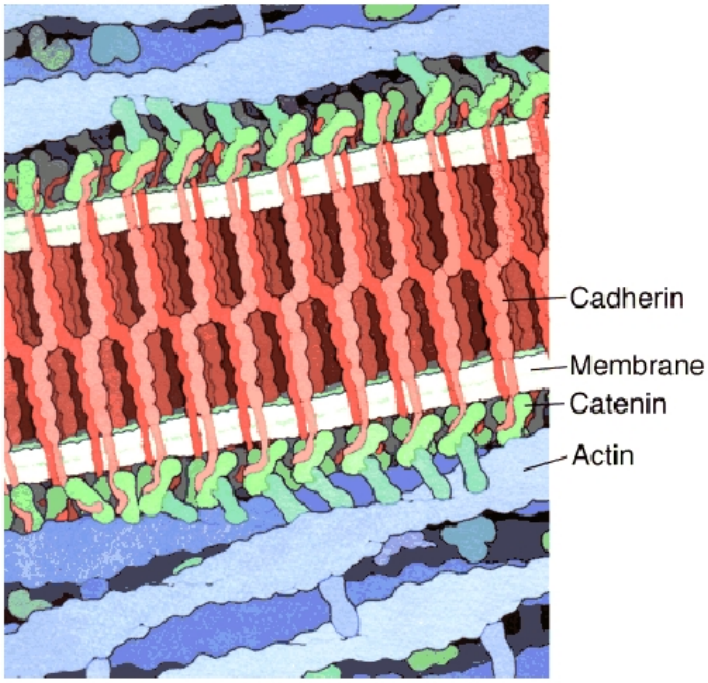


Figure 5.2: Cadherins

Cadherin Resource states that the removal of calcium abolishes adhesive activity and renders cadherins vulnerable to proteases [46]. Another important point is that cadherins are mainly located at cell junctions. Changing the amount of amino acids can also affect the cadherin functioning properly.
The primary function of the cadherin is the cell to cell bonding. Cadherins have an impact also on cell differentiation, individual cell growth, and cell to cell signaling. As quoted by Ivanov, et al, "To Date, numerous data indicate that cell adhesion receptors can affect cell form, motility, and growth not only due to mechanical attachment of the cells to each other or to the substrate, but also by activating internal signaling" [40]. This appears to be due to the affect the cadherin has over β-catenin. β-catenin has been shown to play a role in cell to cell signaling. According to Ivanov, Philippova, and Tkachuk,

> "It was shown that over expression of cadherins in the embryos Xenopus laevis and Drosophila inhibited signal

> transduction via β-catenin. In Xenopus embryos, the inhibition is due to β-catenin binding with C-cadherin on the inner surface of the cell membrane. As this takes place, β-catenin is removed from its cytoplasmic pool, becoming inaccessible for participation in signaling. Thus, cadherins can regulate β-catenin signaling activity by changing it's distribution in the cell" [40].

From this, one can clearly see the importance of cadherin function not only for it's primary job of cell to cell bonding but also it's affects on other cell functions.

Loss of cadherin production and/or a breakdown in cadherin function have shown detrimental effects. Studies have shown that low adhesion capabilities relate directly to individual cell movement or drifting. Once cadherins fail to work properly, the cells will lose their ability to bond to those adjacent. In the case of cancer, this allows individual cells to separate from a solid tumor. They are free to wander through the body and form metastases [26]. People have also been found to develop *pemphigus vulgaris*. This is a disease where the cell to cell adhesion capabilities fail and the epidermal cells fall off the body [84]. Other instances of cadherin function failure are mutations in the proteins, mutation of the cadherin itself, and the creation of a protein-cutting enzyme that actually attacks cadherin. Overall, any of these failures result in the loss of the adhesion function of the cadherins.

Unfortunately we were unable to find specific cadherin levels in either a healthy/stable condition or in a test subject with failing cadherin levels. While studies have identified 4 specific cadherin types and some of the molecules involved in these complex chains, the quantities were not found. For further progress in this area, research for future groups must be focused on finding actual test data which demonstrates:

- cadherin levels in a normal system
- amount of Ca++ (calcium) present
- source/amount of Ca++ available
- source/amount of amino acids
- further research into the β-catenin

- exchange possibilities cell to cell

With this information, an addition to the model could be created to represent the importance of cadherin function in tissue. An amount for each cell in the model could be determined and depending on the level of the surrounding cells a number could be associated to represent the strength of the bond between the adjacent cells. If the cell to cell bond were a high level the cells would remain where they are located. If the cell to cell bond is a moderate level a cell would be able to possibly differentiate or move up a layer. This would happen if there were the proper cadherin levels on the next layer up so that it could bond to the adjacent cells on that layer. However, if the bond is way too low, a cell would have the tendency to break apart from all those surrounding it and thus migrate out of the model (representing a possibility of metastases). To begin to represent this in the model more research is needed in the areas listed above.

5.6 Summary/Conclusion

One measure of success of the model is that it reaches stability. By stability we mean that The number of cells in each layer does not show wide variation.
This model does not achieve that. It is very sensitive to the various rates chosen. If cell division is too slow cells move up faster than they can be replenished and this leads to complete destruction of the lower layers. If too fast with respect to cell movement they pile up in layer 2.
The problem seems to be that we need feedback loops between the division rate and movement rate.
The model described in this paper was Version 2. The original Version did achieve stability, but required an assumption (relief of height to width ratio) on movement that not appear to be realistic from a biological perspective. We can now see that it achieved this stability due because movement rate was driven by division rate, in fact was tightly coupled to division rate. Cell division may occur at some low steady rate and this likely does drive cell movement somehow to keep the cell density in layer near a steady value. But it seems probable that opposite is also true. Should the cell density fall below a certain level it stimulates cell division.
While it still seems reasonable to postulate a factor x that is required to break the cell bonds and allow for movement, it may be that enough factor

x is produced in a short period of time to do this, and that its production is a respond to local densities and not the driving factor in cell movement. That is once some threshold of density is reached cells must move to relieve it.
However, the conditions defining local density (height/width ratio) in Version 1 still do not seem reasonable in that this leads to a very strong preference for moving up the cells smallest in size.
One of the desires of client was to have a model that might demonstrate "interesting behavior". In this sense the model is partially successful. It gives insight into tissue dynamics, what works and what doesn't, and it is insight that is not readily apparent without running the model to see what happens.

5.7 For Future Groups Pursuing this Problem

Where possible we have let the biology inform our choices. Not surprisingly, this left quite a few holes in our model. As this was expected, one of our main goals in writing the program was to achieve flexibility. We hope the program can accommodate future changes due to an increase in the next group's understanding of the underlying biology.
The model could be modified in the following ways.
More than 3 tissue layers. The behavior in a 3 layer is probably too limited.
Cell gaps should be allowed, that is there might be empty spots within the tissue or on the surface where cells are missing. Right now the model allows for an arbitrarily wide cell. This is not realistic. There must be some limit on how wide a cell can be. This being the case cell gaps must naturally occur as cells are shed or die, either due to age or damage. The model needs to handle these gaps both in the display and by filling them as rapidly as possible. Repair of tissue damage is a necessity for modeling healthy tissue. A mechanism should be coded for creating cell damage (possibly via GUI interface). The model should then be able to repair it.
The growth and division algorithms are too simple. Both of these actions must respond in some way to cell damage and cell movement. This implies some kind of diffusion of chemicals between cells which acts as feedback loops affecting these processes.

Finally, a word of advise to the next group. Whatever the solutions you come up with, program it and check it out as fast as possible.

5.8 Appendix: Glossary of Terms

The following general terminology was found useful in working our way through some of the literature on the subject.
Morphology: outward appearance, including shape, color, structure,and pattern
Amplifying Cells: cells whose function is to increase the number of cells available for subsequent maturation and mitosis
Cytokine: signaling compound
Desquamation: shedding of outer layers
Masticatory: Chewing
Integrin: Responsible for attachment to extra cellular matrix (ECM) and other cells. Plays vague role in signal transduction, especially from ECM to the cells
Epithelium: lining tissue/cells, and layers of cells; one of four primary types of body tissues
Differentiation: process whereby cells acquire a type, and the cell morphology changes
Squamous Epithelium: epithelium named after top, surface layer of cells
Squamous Cells: flat, outermost cells
Cadherin: responsible for cell adhesion, plays role in ensuring that cells within tissues remain bound together
Growth Factor: Cells that stimulate cellular proliferation and differentiation
Kinase: type of signaling protein
Transit Amplifying Cells: An alternative to the more pure stem cells, existing usually within one of the lower layers, close to the basal lamina. This cell type is slightly more differentiated than the true stem cells, with less capacity for indefinite division
Basal Lamina: foundation, or "bottom", upon which the epithelial cells are built/situated

Chapter 6

Review of Net Logo and V-Cell

By John Boren and Jon-Michael Brinkerhoff

6.1 Introduction

For the entirety of the spring 2007 semester, The University of Colorado at Denver's Math Clinic (4/5779) has been researching models of cells and tissue to suit the task of Dr. Randall Tagg (UCDHSC, Dept. of Physics.) His efforts to develop and improve computational models for tissue dynamics are important in his research of early oral cancer detection.

As a whole, the clinic's objectives were to develop a rudimentary modeling system with which further investigation by other Clinics could be built upon. Specifically, this was achieved by breaking the clinic into five groups. Four of these groups would begin the challenge of creating these modeling systems using the research of cellular biologists and their mathematical models of cell division and tissue dynamics.

Our group's (group 5) task was to find, test and review programs which already exist and determine their feasibility as suitable alternatives to MATLAB, the programming language that had been chosen by the clinic to model tissue and cells. This review is specifically helpful at the beginning of this modeling process as it may reduce the number of work hours needed to program models in the future, giving more time for researchers to develop models.

This report is a review of these programs, it is our group's opinions of functionality as we mimicked the four other group's programming, using

outside software. There is no theoretical nor empirical data in this report to speak of. However, these reviews are important as they may help guide the direction of future clinic's programming languages and in turn, determine the amount of time they spend programming rather than modeling.

Our key findings show the limitations of the current software available and potential advantages of using these software packages. In general, we've found certain outside programs to have large initial setup times but reduced programming times for beginner programmers once those initial setups have taken place. However, to the astute programmer, none of the programs tested had a major advantage over MATLAB. Many times it was in fact the case that MATLAB was a superior programming language.

Objectives of the Review Our objectives in reviewing outside software were to answer five main questions of each language or application we encountered.

- What are the capabilities and limitations of this programming language?
- What is the learning curve of this programming language?
- What applications of this language may be helpful for further clinics?
- Should this language be further studied or does this language have potential for use?

6.2 Methods

To review, we found two outside languages with which we would program those models found by the four other research groups. Each model was programmed an run using the perspective software and notes were made during those sessions which lead us to our conclusions. The two models of interest were those which were discussed in the clinic by both cellular modeling groups. Namely, we programmed the models found in:

- Modeling the Cell Division Cycle:cdc2 and Cyclin Interactions" [67]
- Dynamics of the Cell Cycle: Checkpoints, Sizers, and Timers" [55]

The latter paper being more important to us as reviewers because of its breadth of differential equations which were required for the model to work.

6.3 Results

Our testing began with an initial screening of available software. From the myriad of collaborative projects on cellular modeling found, few had implemented working programs with which to model. There were however, four programming packages which seemed to be comprehensive and well developed, suited for our needs in the math clinic. These were:

- Net Logo
- Virtual Cell (V-Cell)
- E-Cell
- Cell Electrophysiology Simulation Environment (CESE)

Unfortunately, there was not enough time to test all of the programming packages. Net Logo and V-Cell seemed to be best for our time constraints and both were seemingly designed to accomplish different tasks. Net Logo had the potential to be the better program for tissue modeling and V-Cell for cells. E-Cell, though our impressions led us to believe it had potential, was put at the bottom of our list for testing and regrettably never tested due to time constraints. CESE was never investigated further because of its lack in development compared to the others listed. However, CESE is highly active and should develop greatly in the coming years. Hence, it is worth noting in this review.

The two programs which were reviewed in the course of this semester were Net Logo and V-Cell. Both are freely available to the general public and were created out of an apparent need for them to exist. However, many of the features outlined in these programs are not unique to them.

Net Logo This program was developed by Uri Wilensky as a way to dynamically change variables and constants to see real time results and graphs. It is based heavily on Java. The idea of this program is that it's open to model anything the user wants. That is to say the user creates his program for a specific model. This is a start from scratch language in that whatever the user would like the program to do needs to be coded specific to that task.

Trials and Experimentation In our review of the Net Logo, we did not have a tissue model with which to code. We evaluated this software by creating and using a user's interface that may have potential for future Clinics.

Capabilities

- *GUI-* The Graphical User Interface was set up very well. Not only was the editor easy to comprehend but the user defined GUI was easy to set up and is the reason for the existence of this software. Whereas setting up visuals and user interfaces in MATLAB is quite difficult, only simple commands and buttons were needed to do so in Net Logo. An example of the editor is shown as Figure 1.

- *Sliders-* The programming language is designed to use mouse-drag sliding scales. We found the ability of the user to set up a slider and slide the rates of change and constants of his equations, almost infinitely within a user defined interval, to be a valuable option. In witnessing the cellular group's need to "play around" with these numbers to find a suitable limit cycle, we began an attempt to implement these sliders. If these sliders were implemented before other groups completed their tasks, this process could have greatly reduced the trial and error time the other groups lost.

- *Real Time Animated Graphs-* Use of the sliders can determine the speed at which an animated graph is running. This made it easy to manipulate the visuals for better interpretation by an audience. The graphs could just as easily be sped up to witness long term results.

- *Multiple Graphs-* Since Net Logo's primary function is a GUI, it is easy and potentially helpful to the user to set up multiple graphs running at user defined rates. Though this isn't something MATLAB cannot do, it is very easy to set up and can be done to produce very understandable and good looking graphs. Whereas a MATLAB graph is simple, these graphs can be extravagant and organized in a way that is clearest to the audience.

- *Ease of Use for End User-* Once a program is created, it is extremely simple for the end user to enter in the information they wish to model

and push a button to begin the program. For broader research, such as the kind performed in this clinic, functionality such as this would be very helpful. If the program was implemented before the semester begun, either of the models mentioned above would have been programmed in a few simple clicks. Considering the time used to program the models, this type of programming is very advantageous.

Limitations

- *Java Based-* This application is Java based and comes with all the same limitations as Java. It is not well designed to handle the sorts of mathematics which could be programmed in MATLAB. For example, the matrix functions are not nearly as effective in Net Logo nor are there as many solvers as are in MATLAB.

- *Steep Learning Curve-* Although this is arguable because Java is less difficult to learn than many languages and widely used throughout the world, the product intended for the end user is not easy to change without heavy coding. This means that should someone be using an end product made specifically for ease of use as described above, the user would not be able to change the capability of the program without new code. Though the end product is made to resemble and indeed functions like other easy to use programs, there are no GUI options to change its capabilities.

- *Poorly Documented-* Net Logo is freeware and like most freeware, there is no incentive to make the product marketable or well documented for users. There is documentation but compared to the thousands of pages of manuals and bound books describing MATLAB, Net Logo is severely lacking.

Virtual Cell (V-Cell) V-Cell was developed by the National Resource for Cell Analysis. It is a unique modeling environment developed for quantitative cell biological research. The modeling language, like the program, is unique but similar to many modeling languages being used in the numerous fields of mathematics.

Trials and Experimentation In our trials, we tested a few systems of differential equations. Some were very basic systems to test our understanding of the language and we also coded the larger ones found in the "Dynamics of Cell Cycle", "Modeling the Cell Division Cycle" and some which were presented during class by group 2. You can see an example of our code and some of our graphs in figures 2-4. A more advanced graph is shown as an example in Figure 5.

Capabilities

- *Effective Differential Equation Solver*- Each system tested was solved quickly with very little coding. Comparatively, the amount of coding was less that that of MATLAB to solve the same problem. The user need not include commands to produce solutions, plot graphs etcetera. Rather, he simply plugs in the equations he wishes and runs the program. See Figure 2 for sample code.

- *Easy to Learn Language*- Because of the simplicity in entering systems of differential equations, the basics of the program can be conquered quickly by novice users with the help of an example. There exist higher functions, but we've seen no use for these in testing the models presented during the clinic and hence, they were not used.

- *Excellent Graphs*- Once the program solves a system of differential equations, the data is output to a cached file and graphs can be created using the graphing function. This allows the user to select the plot he requires from mouse activated drop down menus. Each variable can be plotted against another with the push of the button and consequently, the data and graphs are easily exported to other applications if required. Figure 3 and 4 show some examples of these graphs.

- *Organize and Draw Functions*- The cell is represented as a picture in the software. To keep track of items the user wishes to model, he can draw these in the cell using the tools available within the software. These drawings are not animated but in larger systems do a good job of keeping order visually of what's going on should the user invest the time to create them. There are also system drawing tools with which the user can draw reactions and movements but again, these are not animated and are up to the user to produce rather than being produced automatically.

Limitations

- *Limited Programming Capabilities-* The software and its language are very useful for solving equations and plotting solutions however it is limited to these sorts of models and doesn't have the breadth of programming opportunities that MATLAB or Net Logo have. The usefulness of the software ends at what the limited modeling language can accomplish.

- *Does Not Solve Locally-* V-Cell is not a software package but is actually an application that connects and transmits to a solver on a server at the University of Connecticut Health Center. This has two implications. First, the user must have a working internet connection to solve problems and the server on which the solver resides must be up and running. Second, the user is at the whim of application changes made by the National Resource for Cell Analysis. Upgrades to language and solvers are forced rather than acquired as in MATLAB and Net Logo.

- *Poor Editor-* The application uses a very poor editor. Unless a program is written flawlessly on the first try, an inevitable error checking tool will begin to run. The error checker gives a line and error name but it's up to the user to find which line and what went wrong. This is not easily accomplished because there are no line numbers on the editor nor is there a readily accessible or mentioned description of the error name. Further, no color coding or animation to indicate parentheses sets or anything else, makes writing out lengthy differential equations difficult, ensuring you will receive an error message. This could be solved with a copy and paste from another editor, however our attempts to do so were unsuccessful, creating fake spaces. This is one of the largest drawbacks of V-Cell and added lots of wasted time to the modeling process.

MATLAB Though MATLAB was never actually used by our group, as a standard among many university math departments, it is well documented and heavily used. The functionality is directly related to solving the mathematical problems which arose in our clinic's research. The other teams in our clinic were able to successfully code their problems and obtain data and graphs suitable for export. Now that the hurdle of initial coding has been passed, testing future models using similar code will only be a matter of code editing, rather than code writing. There has been nothing our group was able

to achieve with Net Logo and V-Cell that the other groups using MATLAB weren't able to do.

Conclusions Through our testing we've seen a potential in each of the programming packages. Some though, are better than others.

V-Cell is the quickest to learn and makes wonderful graphs with little effort but its lack of documentation, time consuming editor and dependency on external solvers may make it a difficult program to implement in future math clinics.

Net Logo has the most potential to become an integrated part of future clinics. It is dynamic, feature rich and produces very good looking output. The caveat being that a large amount of initial work would need to be done in order to get a proper program coded and running. This is the best long term solution for future math clinics where a turn key system would help students discover more about the problem and less about the coding.

MATLAB has shown to be the most reliable solution. It was implemented within the time constraints of our clinic and was able to produce very good results with minimal coding. It has a strong network of users, documentation and availability. Should future clinics wish to run similar models, the coding infrastructure is already at their disposable for editing, decreasing the initial learn time by new students.

6.4 Recommendations

Based on our review and findings we recommend MATLAB be the primary language used for further study of cellular and tissue dynamics. However, future clinics may also want to consider a few other options:

- Research the possibility of using E-Cell. Its programming language is based in C++, similar to that of MATLAB and is directed towards complex systems in cellular biology.
- Consider setting up an easy to use Net Logo program designed to quickly solve systems of differential equations through a GUI to assist novice programmers in expanding their cellular and tissue dynamics research.

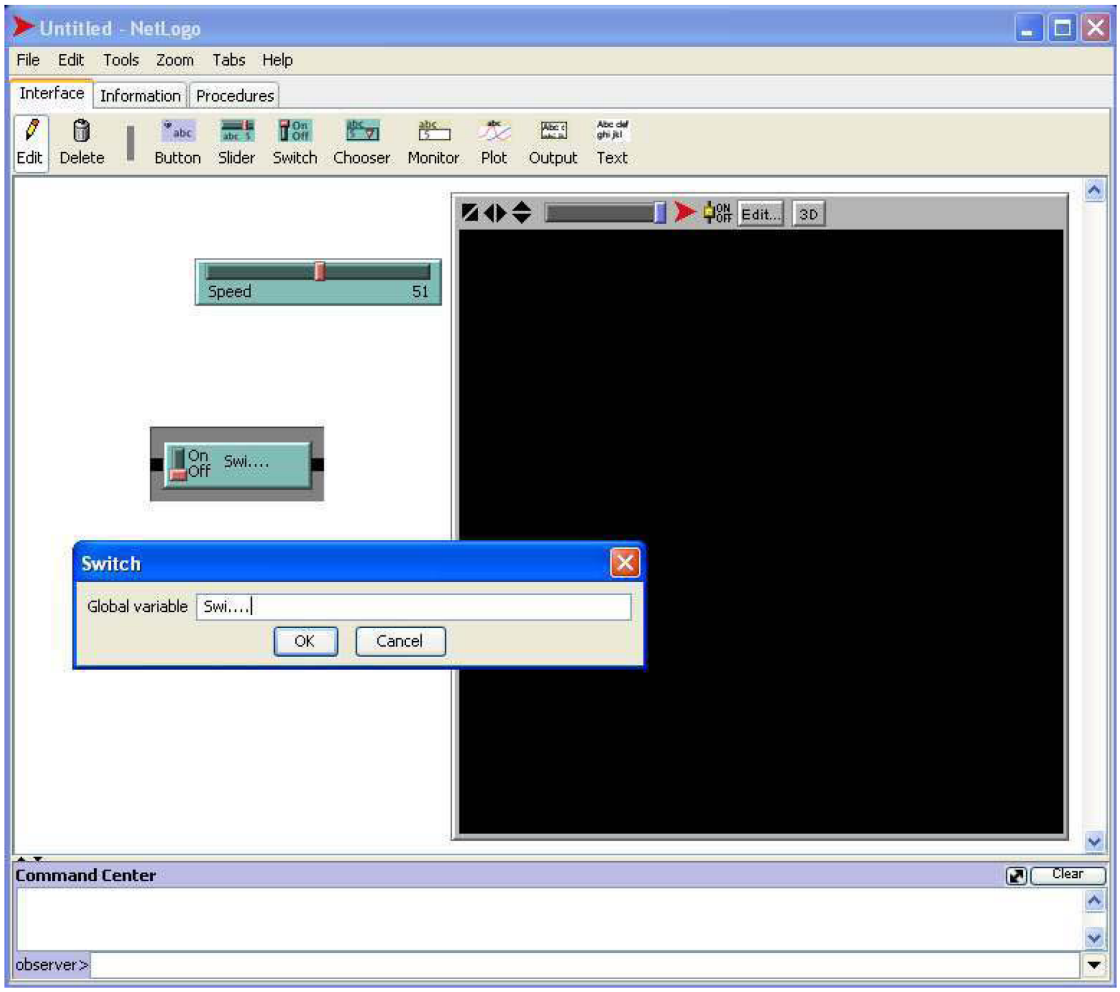
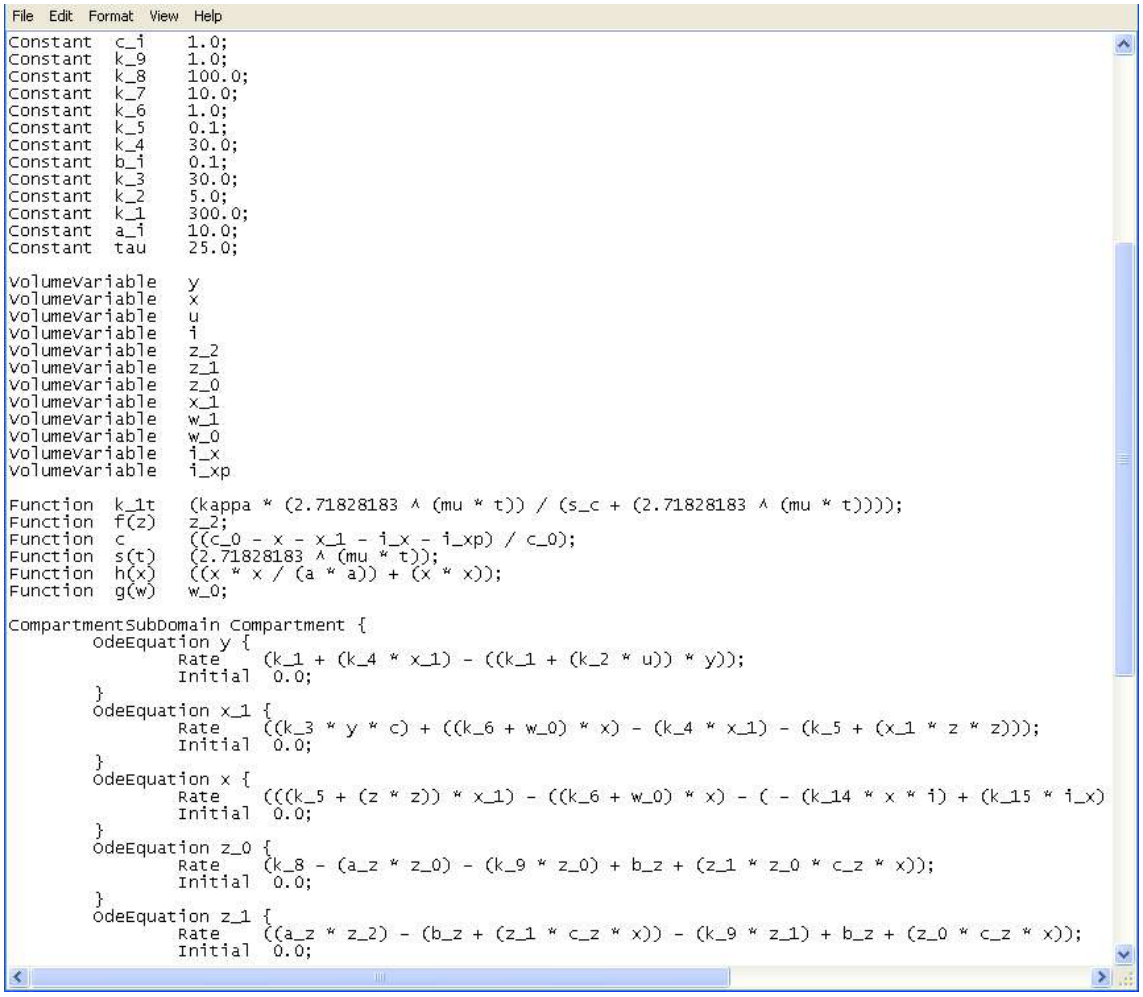


Figure 4.1: Creation of buttons and sliders are done in this editor. Everything is click, drag and label. Here, I'm making a switch labled "swi..."



```
File  Edit  Format  View  Help
Constant  c_i    1.0;
Constant  k_9    1.0;
Constant  k_8    100.0;
Constant  k_7    10.0;
Constant  k_6    1.0;
Constant  k_5    0.1;
Constant  k_4    30.0;
Constant  b_i    0.1;
Constant  k_3    30.0;
Constant  k_2    5.0;
Constant  k_1    300.0;
Constant  a_i    10.0;
Constant  tau    25.0;

VolumeVariable   y
VolumeVariable   x
VolumeVariable   u
VolumeVariable   i
VolumeVariable   z_2
VolumeVariable   z_1
VolumeVariable   z_0
VolumeVariable   x_1
VolumeVariable   w_1
VolumeVariable   w_0
VolumeVariable   i_x
VolumeVariable   i_xp

Function  k_1t   (kappa * (2.71828183 ^ (mu * t)) / (s_c + (2.71828183 ^ (mu * t))));
Function  f(z)   z_2;
Function  c      ((c_0 - x - x_1 - i_x - i_xp) / c_0);
Function  s(t)   (2.71828183 ^ (mu * t));
Function  h(x)   ((x * x / (a * a)) + (x * x));
Function  g(w)   w_0;

CompartmentSubDomain Compartment {
        OdeEquation y {
                Rate    (k_1 + (k_4 * x_1) - ((k_1 + (k_2 * u)) * y));
                Initial  0.0;
        }
        OdeEquation x_1 {
                Rate    ((k_3 * y * c) + ((k_6 + w_0) * x) - (k_4 * x_1) - (k_5 + (x_1 * z * z)));
                Initial  0.0;
        }
        OdeEquation x {
                Rate    (((k_5 + (z * z)) * x_1) - ((k_6 + w_0) * x) - ( - (k_14 * x * i) + (k_15 * i_x)
                Initial  0.0;
        }
        OdeEquation z_0 {
                Rate    (k_8 - (a_z * z_0) - (k_9 * z_0) + b_z + (z_1 * z_0 * c_z * x));
                Initial  0.0;
        }
        OdeEquation z_1 {
                Rate    ((a_z * z_2) - (b_z + (z_1 * c_z * x)) - (k_9 * z_1) + b_z + (z_0 * c_z * x));
                Initial  0.0;
```

Figure 4.2: Sample Code for system of Diff EQ's found in "Dynamics of the Cell". As you can see, this is an easy to understand language. Here I am writing my equations and variables.

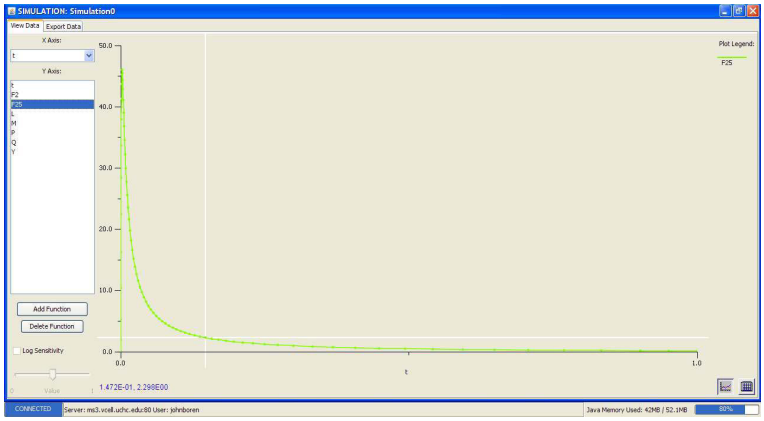


Figure 4.3: A graph of t vs F25 as defined by the code in Figure 4

```
File  Edit  Format  View  Help
MathDescription {

Constant  k25     0.03;
Constant  k3      50.0;
Constant  k2      0.0080;
Constant  k025    50.0;
Constant  k1      0.0040;
Constant  k02     3.0;
Constant  C       1.0;
Constant  kwee    1.0;

VolumeVariable    Y
VolumeVariable    M
VolumeVariable    L

Function  Q       (C + L - Y);
Function  P       (Y - L - M);
Function  F25     (k25 + (k25 * M * M));
Function  F2      (k2 + (k2 * M * M));

CompartmentSubDomain Compartment {
        OdeEquation L {
                Rate    (k1 - (F2 * L) - (k3 * L * (C + L - Y)));
                Initial  k3;
        }
        OdeEquation M {
                Rate    ((k3 * L * (C + L - Y)) - (F2 * M) + (F25 * (Y - L - M)) - (kwee * M));
                Initial  kwee;
        }
        OdeEquation Y {
                Rate    (k1 - (F2 * Y));
                Initial  k1;
        }
}

}
```

Figure 4.4:

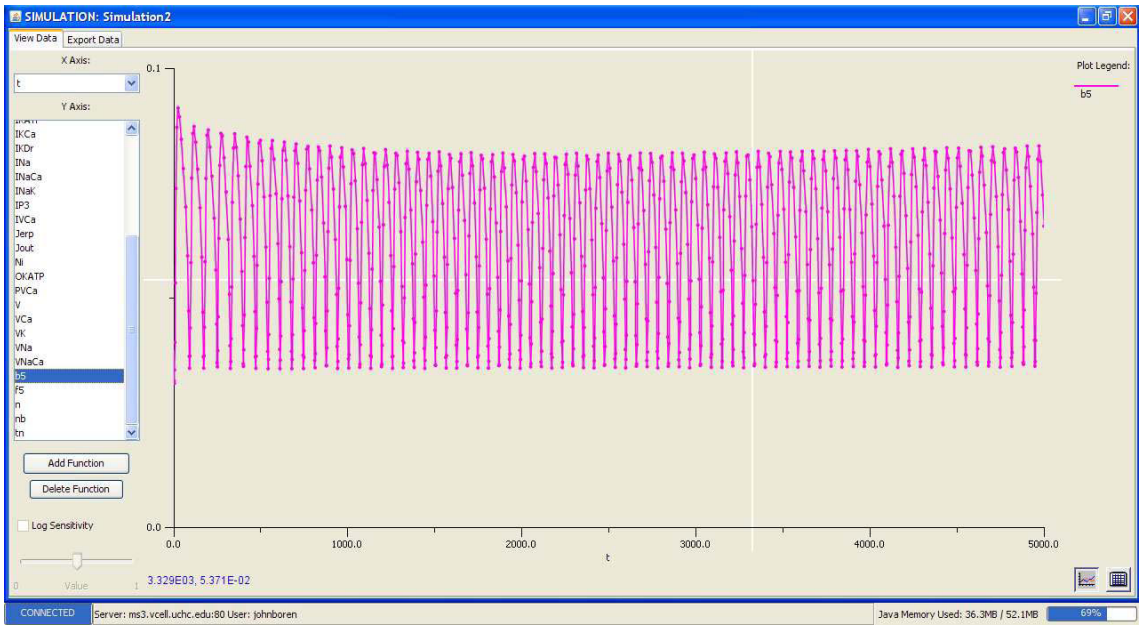


Figure 4.5: Here we see a more complex graph. This is an example model called "Simulation 2" and indeed all models submitted by users of this package are available as examples to others.

Bibliography

[1] Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., and J.D.Watson. *Molecular Biology of the Cell*. Garland Publishing, Inc., New York, New York, 2nd edition, 1989.

[2] Roger Alperin and Robert Lang. One-, two-, and multi-fold origami. World Wide Web, December 2006.

[3] Wolfgang Alt. Biased random walk models for chemotaxis and relaxed diffusion approximations. *Journal of Mathematical Biology*, 9:147–177, 1980.

[4] Nakhle Asmar. *Partial Differential Equations and Boundary Value Problems*. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[5] Irina Baran. Calcium and cell cycle progression: possible effects of external perturbations on cell proliferation. *Biophysical Journal*, 70:1198–1213, March 1996.

> This article has more to do with the effects of the addition of extracellular calcium on the cell cycle progression. It does have a series of differential equations that can possibly be adapted to a variety of different chemicals or ions.

[6] Barnes, Cox, Gray, McGinty, Pham, Sosnicki, and Wicks. Integrins-how to move by getting stuck. World Wide Web, 2007.

[7] R. Baserga. *Cell Growth and Division: A Practical Approach*. IRL Press, NY, NY, 1989.

[8] Boler, C. B. *Numerical Computing with Matlab*. Society for Industrial and Applied Mathematics, Philadelphia,PA, 2004.

[9] William E. Boyce and Richard C. DiPrima. *Elementary Differential Equations*. John Wiley and Sons, Inc., NY, NY, 1997.

[10] Boyce, W. E. and R. C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. John Wiley & Sons, Hoboken, New Jersey, 1992.

[11] Christopher T. Capaldo and Ian G. Macara. Depletion of e-cadherin disrupts establishment but not maintenance of cell junctions in madin-darby canine kidney epithelial cells. *Molecular Biology of the Cell*, 18:189–200, 2007.

[12] Raymond Chang. *Chemistry*. McGraw Hill Higher Education, eighth edition, 2005.

This is a general chemistry book. It is very well written and very helpful in terms of learning basic chemistry applications.

[13] Pierre Charles Romond, Mauro Rustici, Didier Gonze, and Albert Goldbeter. Alternating oscillations and chaos in a model of two coupled biochemical oscillators driving successive phases of the cell cycle. *Annals of the New York Academy of Sciences*, 879:180–193, 1999.

This article discusses the presence of chaos in the cell cycle by evaluating the strength of the inhibition between two cellular components, mainly different types of cyclins and cyclin-dependent kinases. It has excellent graphs demonstrating the effects that small perturbations have on thet shape of the phase plane as well as a series fo differential equations to model this behavior.

[14] Ian Conlon. Checking how cells grow. *Journal of Biology*, 24, 2003.

[15] P. Michael Conn and Anthony R. Means, editors. *Priniciples of Molecular Regulation*. Humana Press, Totowa, NJ, 2000.

[16] Peter R. Cooke. *Principles of Nuclear structure and Function*. John Wiley and Sons, 2001.

[17] P Crespon and J León. Ras proteins in the control of cell cycle and cell differentation. *Cellular and Molecular Life Sciences*, 57:1612–1636, 2000.

This article had very indepth information on the structure and function of Ras proteins, as well as descriptions of laboratory techniques. The level of this paper is probably above what needs to be understood by the clinic participants. However, Ras may have a place in cell-to-cell signaling.

[18] Csikász-Nagy, A., Battogtokh, D., Chen, K.C., Novak, B. and J.J. Tyson. Analysis of a generic model of eukaryotic cell cycle regulation. *Biophysical Journal*, 90:4361–4379, 2006.

[19] Wikipedia entry for cyclin-dependent kinase. World Wide Web, 2007.

[20] Teri G Boulton et al. ERKs: A family of protein-serine/threonine kinases that are activated and tyrosine phosphorylated in response to insulin and ngf. *Cell*, 65:663–675, May 1991.

ERK refers to extracellular signal-regulated kinase; this article appears to have potential use for the incorporation of both extracellular events that affect cellular reproduction and differentiation as well as the signaling pathway that controls the activation of certain cyclin/Cdk complexes.

[21] Farlow,, J., Hall, J. E., McDill, J. M. , and B. H. West. *Differential Equations & Linear Algebra*. Prentice Hall, Upper Saddle River, New Jersey, 2002.

[22] Wolframmathworld entry for fourier series. World Wide Web, 2007.

[23] Gamba and et al. Diffusion-limited phase separation in eukaryotic chemotaxis, 2007.

[24] Paul Glendinning. *Stability, Instability and Chaos: an Introduction to the theory of nonlinear differential equations*. Cambridge University Press, New York, NY, 1994.

[25] Goldbeter,A. *Biochemical Oscillations and Cellular Rhythms: The molecular bases of periodic and chaotic behaviour*. Cambridge University Press, New York, 1996.

[26] David S. Goodsell. Fundamentals of cancer medicine, the molecular perspective: Cadherin. *The Oncologist*, 7(5):467–468, 2002.

[27] Roy O. Greep and Leon Weiss. *Histology*. McGraw-Hill, third edition, 1973.

[28] Hall, M.N., Raff, M., and G. Thomas, editor. *Cell Growth: Control of Cell Size*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York, 2004.

[29] Bruce Hannon and Matthias Ruth. *Modeling dynamic biological systems*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[30] Duane C. Hanselman and Bruce Littlefield. *Mastering MATLAB: A Comprehensive Tutorial and Reference*. Prentice Hall, Upper Saddle River, New Jersey, 1996.

[31] Kanthen K Harikrishnan and Hiroaki Ishii. Single machine batch scheduling problem with resource dependent setup and processing time in the presence of fuzzy due date. *Fuzzy Optimization and Decision Making*, 4:141–147, 2005.

[32] Jane V Harper and Gavin Brooks. The mammalian cell cycle. In Tim Humphrey and Gavin Brooks, editors, *Cell Cycle Control*, volume 296 of *Methods in Molecular Biology*, chapter 6, pages 113–153. Humana Press, Totowa, New Jersey, 2005.

> A large store of information can be found in this paper. A detailed description of each aspect of the cell cycle can be found here. It also contains good references for more in-depth research for the interested reader. The rest of the book may also be useful for extension of cell cycle control to other types of cells or species.

[33] Stephan Hoffman, Shikun He, Manlin Jin, Marianne Ehren, Peter Wiedemann, Stephen J. Ryan, and David R. Hinton. A selective cyclic integrin antagonist blocks the integrin receptors $\alpha v\beta 3$ and $\alpha v\beta 5$ and inhibits retinal pigment epithelium cell attachment, migration and invasion. *BMC Ophthalmology*, 5(16), 2005.

[34] Shouchuan Hu. *Discrete and continuous dynamical systems*. AIMS, Springfield, 2004.

The book is about dynamical systems. It is a result of extensive studies on non-autonomous dynamical systems with VV. Chepyzhov. This book is good and can be followed for someone that has had dynamical systems already. It will be difficult to follow for a beginner, though.

[35] Hunt, B., Lipsman, R. L., Osborn, J. E. and J. M. Rosenberg. *A Guide to Matlab for Beginners and Experienced Users*. Cambridge University Press, New York, New York, 2001.

[36] Nicholas T Ingolia and Andrew W Murray. The ups and downs of modeling the cell cycle. *Current Biology*, 14:771–777, September 2004.

This article contains a discussion of the various attempts that have been made to model the cell cycle. It serves mostly as a critique paper, not suggesting any ideas nor providing any ranking for the models discussed.

[37] Wikipedia entry for integrin. World Wide Web, 2007.

[38] The integrin page: Basic integrin info. World Wide Web, 1997.

[39] Zinat Ismael. Personal interview, April 2007.

[40] D.B. Ivanov, M.P. Philippova, and V.A. Tkachuk. Structure and functions of classical cadherins. *Biochemistry*, 66(10):1174–1186, 2001.

[41] Kaneko, K. *Life: An Introduction to Complex Systems Biology*. Springer, New York, 2006.

[42] Wikipedia entry for kinase, 2007.

[43] William S Klug and Michael Cummings. *Essentials of Genetics*. Prentice Hall, Upper Saddle River, New Jersey, 4th edition, 2002.

[44] Kurt W Kohn. Functional capabilities of molecular network components controlling the mammalian G1/S cell cycle phase transition. *Oncogene*, 16:1065–1075, 1998.

This article models the G1/S phase transition using a alternative method - instead of using ODEs, Kohn uses a programming language to model the network components (defined as

the molecular compounds involved in phase transition). There are good graphs present to verify other models, as well as the presence of parameters that may potentially be used within other ODE models.

[45] Bernard Kolman. *Introductory Linear Algebra with Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1997.

[46] Ikura Lab. The cadherin resource: Molecular and structural properties. World Wide Web, 2002.

[47] Douglas A. Lauffenburger and Jennifer J. Linderman. *Receptors*. Oxford University Press, NY, NY, 1993.

[48] Medicinenet.com definition of integrin. World Wide Web, 2007.

[49] Meijer, L., Guidet, S. and H.Y. Lim Tung, editor. *Progress in Cell Cycle Research*, volume 1. Plenum Press, New York, 1995.

[50] Morel, D., Marcelpoil, R., and G. Brugal. A proliferation control network model: The simulation of two-dimensional epithelial homeostasis. *Acta Biotheoretica*, 49:219–234, 2001.

[51] Mostow. *Mathematical Models for Cell Rearrangement*. Yale University Press, 1975.

[52] Murray, A.W. *The Cell Cycle: an introduction*. W.H. Freeman and Company, New York, 1993.

[53] D.A. Lauffenberger P.A. DiMillla, K. Barbee. Mathematical model for the affects of adhesion and mechanics on cell migration speed, 1991.

[54] Eva Part-Enander and Anders Sjoberg. *The MATLAB Handbook 5*. Addison-Wesley, Reading, Massachussets, 1999.

[55] Weiss Qu, MacLellan. Dynamics of the cell: Checkpoints, sizers,and timers. *Biophysical Journal*, December 2003.

[56] Zhilin Qu, W Robb MacLellan, and James N Weiss. Dynamics of the cell cycle: checkpoints, sizers and timers. *Biophysical Journal*, 85:3600–3611, December 2003.

This article discusses an alternative method to Tyson's 1995 approach to modeling the G1/S and G2/M checkpoints. The authors feel that their model is better than Tyson's because it takes into account the possibility of asymmetrical cell division. The other cell cycle group appears to be interested in this paper, so we haven't spent much time reviewing and attempting to code the thirteen equations within it.

[57] Qu, Z., MacLellan, W.R., and J. N. Weiss. Dynamics of the cell cycle: Checkpoints, sizers, and timers. *Biophysical Journal*, 85:3600–3611, 2003.

[58] Qu, Z., Weiss, J.N. and W.R. MacLellan. Coordination of cell growth and cell division: a mathematical modeling study. *Journal of Cell Science*, 117:4199–4207, 2004.

[59] Andrew W. Rogers. *Cells and tissues : an introduction to histology and cell biology*. Academic Press, New York, 1983.

[60] Pierre-Charles Romond, Mauro Rustici, Didier Gonze, and Albert Goldbeter. Alternating oscillations and chaos in a model of two coupled biochemical oscillators driving successve phases of the cell cycle, 1999.

[61] Robert J Sheaff. Regulation of mammalian cyclin-dependent kinase 2. In William G Dunphy, editor, *Cell Cycle Control*, volume 283 of *Methods in Enzymology*, chapter 13, pages 173–193. Academic Press, San Diego, 1997.

This paper did not seem to be particularly helpful for modeling the overall cell cycle or even a checkpoint within the cycle. However, it does discuss different laboratory techniques that are used to determine regulation of Cdk2 in mammalian cells. This would be a better paper to look at if the biology of each player in the cycle became important.

[62] Jill C Sible and John J Tyson. Mathematical modeling as a tool for investigating cell cycle control networks. *Methods*, pages 238–247, 2007.

This paper utilizes a different type of program to model cell cycle. The coding appears to be similar to MATLAB, per the

coding sheet within the paper, but not similar enough to cut-and-paste between the two programs. The benefit of this paper is that it gives an algorithm for creating your own model given a wiring diagram of the chemical reaction. Sible and Tyson also discuss the accuracy of the model by comparing predicted results with observed results. A third item of interest is the brief foray into the available technology and programs to build models of cells. Four such programs are discussed (including VCell, which the programming group has expressed interest in) with a brief description of the capabilities and restrictions of each.

[63] Smallwood,R.H., Holcombe, W.M.L., and D.C. Tyson. Development and validation of computational models of cellular interaction. *Journal of Molecular Histology*, 35:659–665, 2004.

[64] Steven H Strogatz. *Nonlinear dynamics and chaos with applications to physics, biology, chemisty, and engineering.* Westview Press, Cambridge, 2000.

This was the book describes linear and nonlinear dynamics and chaos. I consider it a treasure. It is very well written, but simple enough to quickly understand. I found myself going back to this book for references even after I was done with its class. So far, I have not found another book that can replace it when it comes to these topics.

[65] Strogatz, S. H. *Nonlinear Dynamics and Chaos.* Perseus Books Publishing, Cambridge, MA, 2001.

[66] John J Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proceedings of the National Academy of Sciences of the United States of America*, 88:7328–7332, August 1991.

This article reviews the specific cyclin/Cdk complex that is active at the metaphase/anaphase checkpoint. It also refers strongly to MPF and its influence/presence throughout the rest of the cell cycle. Our first working model used the equations and parameters in this paper.

[67] John J Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proc. Natl. Acad. Sci. USA*, August 1991.

[68] John J Tyson, Katherine C Chen, and Bela Novak. Sniffers, buzzers, toggles, and blinkers: dynamics of regulatory and signaling pathways in the cell. *Current Opinion in Cell Biology*, 15:221–231, 2003.

> This paper has served to be a helpful starting point to an understanding of different types of signaling pathways can be mathematically modeled. These basic graphs can be utilized if a new model, based off a different wiring diagram than any presented in current research, needed to be created.

[69] John J Tyson, Bela Novak, Kathy Chen, and John Val. Checkpoints in the cell cycle from a modeler's perspective. *Progress in Cell Cycle Research*, 1:1–8, 1995.

> This article discusses the G1/S and G2/M checkpoints. The equations that they use for each are similar. It appears to build off the model presented in the Tyson 1991 paper on the metaphase/anaphase checkpoint.

[70] Tyson, J. J., Csikasz-Nagy, A., and B. Novak. The dynamics of cell cycle regulation. *BioEssays*, 24:1095–1109, 2002.

[71] Tyson J.J., and B. Novak. Regulation of the eukaryotic cell cycle: Molecular antagonism, hysteresis and irreversible transitions. *Journal of Theoretical Biology*, 210:249–263, 2001.

[72] unknown. The gene factor: Gene p53. World Wide Web, 2001.

[73] Unknown. Chaos in the cell defect gene disrupts signal processing within the cell and damages the immune system. World Wide Web, January 2007.

[74] National resource for cell analysis and modeling. World Wide Web, May 2007.

[75] Frank Wattenberg. *Calculus in a Real and Complex World*. PNS Publishing Company, Boston, MA, 1995.

[76] Elmer G Wiens. Egwald mathematics: Nonlinear dynamics: Bifurcations and two dimensional flows. World Wide Web.

[77] World Wide Web, 2007.

[78] Uri Wilensky. Netlogo. northwestern.edu, May 2007.

[79] Peter C. Wilkonson. *Chemotaxis and Inflammation*. Churchhill Livingstone, London, England, 1974.

[80] Hao Wu and Zhi-Xin Wang. The mechanism of p21-activated kinase 2 autoactivation. *Journal of Biological Chemistry*, 278(43):41768–41778, October 2003.

[81] Ling Yang, W Robb MacLellan, Zhangang Han, James N Weiss, and Zhilin Qu. Multisite phosphorylation and network dynamics of cyclin-dependent kinase signaling in the eukaryotic cell cycle. *Biophysical Journal*, 86:3432–3443, June 2004.

This article has more in-depth biology and lab-quality information on phosphorylation.

[82] Yang, L., Han, Z., MacLellan, W.R., and Z. Qu. Linking cell division to cell growth in a spatiotemporal model of the cell cycle. *Journal of Theoretical Biology*, 241:120–133, 2006.

[83] Kristina R Yu, Robert J Duronio, and William Sullivan. Cell cycle checkpoints: safe passage through mitosis. In Sharyn A Endow and David M Glover, editors, *Dynamics of Cell Division*, pages 1–41. Oxford University Press, Oxford, 1998.

This chapter has a basic overview of cell division, but then goes into information that does not seem pertainent to modeling the actual cycle. The chapter is heavy on the biological processes.

[84] Zygote website: The biochemistry of cadherins. World Wide Web, 1998.