Mathematical Analysis of a Cell Cycle Model

Kevin DeHoff

December 4, 2003

Abstract

We present an *ab initio* mathematical analysis of the model presented by DeHoff and Obeyesekere [3], including a summary of the model itself and the biological significance of its solution. To determine the worthiness of its solutions, an analysis of the Runge-Kutta class of numerical methods is presented, along with the derivation of a third-order method. The model was tested for the existence and uniqueness of solutions, as well as for the existence of equilibrium solutions, which were not found. The "natural responses" of each member of the system were also determined and documented by removing all other influences and finding the solution curve. We present a series of directions in which this project could move in the future. Finally, we found that the model itself is mathematically and biologically sound, and that the Runge-Kutta class of numerical methods is a standard numerical technique used in a variety of different applications.

Background and Model Design

In [3], a model was designed to simulate the concentrations of different proteins in the nucleus of a cell going through the G2-M phase transition of the cell cycle. During this transition, the DNA of the cell is completing its replication and the cell itself is preparing divide. In this model, $y_i, i \in \{1, \ldots, 12\}$ are known proteins that play a crucial role in this transition. During the transition, each protein interacts with other proteins.

Proteins interact with each other in a variety of ways. Some act by increasing or decreasing the natural production rate of other proteins. Other pairs of proteins join to form different complexes, thereby decreasing the levels of both proteins involved. Proteins transform to different states through a process known as phosphorylation. These phosphorylated proteins were considered to be new proteins.

Our goal was to model the activity of each of the different proteins, and for simplicity, can be thought of as the concentration of the protein in the nucleus. However, since proteins are constantly moving throughout the cell, this becomes an open system[1]. As time passes, one of five things happens to the proteins.

1. A certain amount of the protein is produced.
2. A certain amount of the protein moves out of the nucleus.
3. A certain amount of the protein moves into the nucleus.
4. A certain amount of protein is degraded (destroyed).
5. A certain amount of protein is phosphorylated to become a different protein.

With these interactions and movements in mind, a system of autonomous ordinary differential equations was created to track each protein. [3]

In this system, only two protein levels are biologically important. y_5 is a collection of proteins that kill the cell. If these proteins reach a threshold level, the cell dies and the simulation is over. y_7 is the trigger into the M-phase and cell division. If the concentration of y_7 reaches a certain level, everything "resets", with the levels of proteins returning to their initial concentrations[2] and the cycle continues. Otherwise, the cell cycle stops, and the cell is considered to be *arrested*.

The notation used in the model is as follows:

- y_i is the level of protein i.

[1] An open system is any chemical system where particles and energy can enter or leave the system.

[2] The intial concentration being referred to here is the concentration at the beginning of the cell cycle, not the beginning of the simulation, as it takes time for the simulation to settle into a pure cyclic solution.

- f_i is a constant representing the natural formation rate of protein i.
- e_i is a constant representing the natural degredation rate of protein i (generally proportional to itself)
- $a_{i,j}$ is a constant representing the activation rate of protein i on protein j (increases production of protein j dependent on the level of protein i.) If the subscript contains an e, then the degradation of j is increased dependent on the level of protein j. Although this is mathematically equivalent to the inhibition of the formation rate of the same protein, it is more correct for the biological and chemical description of the process.
- $\dot{xi}_{i,j}$ is a constant representing the inhibition of some reaction involving protein j by protein i. Inhibitions are always dependent on the level of protein i.
- $p_{i,j}$, $q_{i,j}$ are constants used to describe the phosphorylation of proteins[3].
- $r_{i,j}$ is a constant representing the a transformation rate that involves protein i and protein j combining to create an entirely different protein. This is different from the phosphorylation reaction in which a single protein is altered. The resulting proteins are considered to be sinks, and their only fate is to be degraded. Therefore, their resulting levels are unimportant and hence ignored.
- $STIM$ is a function representing an external damage stimulus. Its parameters are α, the intial amount of stimulus applied and β, the half-life of the stimulus.
- *severe* is a constant which determines a cutoff for the severity of the damage stimulus. For damage levels above a certain severity, the cell is modeled to die. Otherwise, the cell arrests and is repaired.

For example, the equation for y_9' is

$$\frac{dy_9}{dt} = \frac{f_9}{1 + \dot{xi}_{6,9} y_6} - e_9 y_9 y_7.$$

In this equation, f_9 is the natural formation rate of protein 9, while the inhibiting term $\dot{xi}_{6,9}$ slows the formation relative to the amount of y_6. Additionally, e_9 is the natural degradation of y_9, which depends on both y_9 and y_7.

The entire model is seen below.

$$\frac{dy_1}{dt} = f_1(1 + a_{2,1} y_2) - \frac{r_{1,2} y_1 y_2}{1 + y_{13} \dot{xi}_{b,r1,2}} - e_1 y_1 \tag{1}$$

[3]This phosphorylation is described by Michaelis-Menten enzyme kinetics

$$\frac{dy_2}{dt} = f_2\frac{1+a_{b,2}y_{13}^5}{severe^5+y_{13}^5} - \frac{r_{1,2}y_1y_2}{1+y_{13}\dot{xi}_{b,r1,2}} - \frac{p_{2,3}y_2}{q_{2,3}+y_2}(1+a_{2,3}y_2)(a_{7,3}y_7+a_{3,3}y_3) - e_2y_2 \quad (2)$$

$$\frac{dy_3}{dt} = \frac{p_{2,3}y_2}{q_{2,3}+y_2}(1+a_{2,3}y_2)(a_{7,3}y_7+a_{3,3}y_3) - e_2y_3 \quad (3)$$

$$\frac{dy_5}{dt} = f_5(1+a_{4,5}y_4^2) - e_5y_5 \quad (4)$$

$$\frac{dy_6}{dt} = f_6(1+a_{2,6}y_2) - e_6y_6(1+a_{5,e6}y_5) \quad (5)$$

$$\frac{dy_7}{dt} = \frac{f_7}{1+\dot{xi}_{6,7}y_6} - \frac{p_{7,8}y_7}{q_{7,8}+y_7} + \frac{p_{8,7}y_8y_{11}}{q_{8,7}+y_8+y11} - e_7y_7^2 \quad (6)$$

$$\frac{dy_8}{dt} = \frac{p_{7,8}y_7}{q_{7,8}+y_7} - \frac{p_{8,7}y_8y_{11}}{q_{8,7}+y_8+y11} - e_7y_7y_8 \quad (7)$$

$$\frac{dy_9}{dt} = \frac{f_9}{1+\dot{xi}_{6,9}y_6} - e_9y_7y_9 \quad (8)$$

$$\frac{dy_{10}}{dt} = f_{10} + \frac{p_{11,10}y_{11}y_{12}}{q_{11,10}+y_{11}+y_{12}} - y_9\frac{p_{7,11}y_7y_{10}}{q_{7,11}+y_7+y_{11}} - e_{10}y_{10} \quad (9)$$

$$\frac{dy_{11}}{dt} = -\frac{p_{11,10}y_{11}y_{12}}{q_{11,10}+y_{11}+y_{12}} + y_9\frac{p_{7,11}y_7y_{10}}{q_{7,11}+y_7+y_{11}} - e_{11}y_{11} \quad (10)$$

$$\frac{dy_{12}}{dt} = f_{12}(1+a_{2,12}y_2) - e_{12}y_{12}(1+a_{5,e12}y_5) \quad (11)$$

$$\frac{dy_{13}}{dt} = a_{s,b}STIM - \dot{xi}_{6,b}y_6 \quad (12)$$

$$STIM = \beta e^{-\alpha t} \quad (13)$$

Due to the nonlinearity of the model, it is impossible to use standard techniques to find solution curves and characterize the system. However, models are of no use if we cannot see the solutions they produce, and we turn to numerical techniques to obtain solution estimates. The algorithm used was a $6-7$ order Runge-Kutta obtained from the IMSL libraries produced by Visual Numerics. To get an idea of how these libraries worked, we study the Runge-Kutta algorithms, including a derivation of a third order Runge-Kutta system based on information from [1] and [5].

Numerical Methods

The general idea behind numerical methods is to estimate what the solution to an unintegrable function is for any given point in time. These processes work by examining the current point in the solution. Since the slope of the solution at that point is known, it is used in some way to forecast what the next point in the solution is going to be. However, due to the very nature of the algorithms, there is always some inherent error. Each numerical method has an item known as the step size. In general, as the step size get closer to zero, the algorithm becomes more and more accurate. However, each decrease in step size includes an

increase in computational time and an increase in the amount of data output by the algorithm. We know that for small step sizes, the maximum error in the algorithm is bounded by a polynomial in the size of the step. There are two functions which are often used in dealing with error.

- $O\big(g(x)\big)$: We say that $f(x)$ is $O\big(g(x)\big)$ if and only if for all x, $f(x) \leq \lambda g(x)$, where λ is independent of x. This is the same as saying $f(x)$ is $O\big(g(x)\big)$ if and only if

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} \leq \lambda.$$

  Note that if λ is a constant and $f(x)$ and $g(x)$ are polynomials, then $\deg f(x) = \deg g(x)$.

- $o\big(g(x)\big)$: $o\big(g(x)\big)$ is a similar idea. However, we say that $f(x)$ is $o\big(g(x)\big)$ if and only if

$$\lim_{x\to 0} \frac{f(x)}{g(x)} = 0.$$

  In this case, $f(x)$ decays faster than $g(x)$ and it is easy to show that $\deg f(x) > \deg g(x)$ for two nonzero polynomials.

We define a numerical solution to be order p if its error is $O(h^{p+1})$ for a given step size h.

Consider the differential equation $y' = f(x, y)$ with initial value $y'(x_0) = y_0$. Given the initial value $\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}$, theories of numerical analysis presented in [1] assume that the solution to y' can be written as

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \Delta x \Phi \begin{pmatrix} x \\ y \\ \Delta x \end{pmatrix}. \tag{14}$$

In the simplest of cases, Euler's Method, the function Φ in (14) is simply $f(x)$. Euler's Method tends to accrue error quickly, because it is a first order method, i.e. the error is $O\big(h^2\big)$. The easiest way to reduce the error of these methods is to increase their order. For example, for an order one approximation, we know that the highest possible error is accumulating proportional to the square of the step size h with each step. So, assuming a step size of 0.5 units, the order after the first five steps could be as high as 1.25. However, if an order two approximation is used, the maximum error accumulates proportional to the cube of h with each step. So after five steps with an order two approximation, the maximum error could only be 0.625—half of the maximum error of the order one approximation.

The desire is to find a method that is of a sufficiently high order to give believable results without making them computationally infeasible. Euler's Method uses the slope at the current value and a uniform step size to determine the next value. It is possible, however, to approximate future slopes, and take a weighted average of these slopes. This weighted average could then be used as the slope in Euler's Method. This approach to increasing the order of the approximation has yielded a class of algorithms commonly known as the Runge-Kutta methods. The most well known and widely used version of these methods is known as the "Standard Runge-Kutta" method, which is the weighted average of the current slope and three forecasted slopes. The Standard Runge-Kutta method is order four, due to the method of its derivation. To give an example of how this works, we will derive a third order Runge-Kutta method, rather than a corresponding order four technique[4]. The basis for the derivation of a Runge-Kutta method as set forth by [1] and [5] is to set the Taylor expansion of a generalized function equal to another approximation of the same solution with a known order. Before we begin the derivation, we will prove the following theorem.

Theorem 1 *The n^{th} degree Taylor expansion of $f(x)$ is order n for any rational function $f(x)$.*

Proof Let $T_n(x)$ be the n^{th} degree Taylor expansion of $f(x)$. We know that this can be written as

$$T_n(x) = f(x_0) + f'(x_0)(x - x_0) + \cdots + \frac{1}{n!}f^{(n)}(x_0)(x - x_0)^n,$$

so that $f(x) = T_n(x) + \varepsilon(x)$, where $\varepsilon(x)$ is the error in the Taylor expansion. Because we know that $\varepsilon_n(x)$ is the remaining terms of the Taylor expansion, it is possible to write

$$\varepsilon_n(x) = \frac{1}{(n+1)!}f^{(n+1)}(x_0)(x - x_0)^{n+1} + \cdots .$$

If we can show that there must be some k between the arbitrary x_0 and x so that

$$\varepsilon_n(x) = \frac{1}{(n+1)!}f^{(n+1)}(k)(x - x_0)^{n+1}, \tag{15}$$

we can begin to get an upper bound on $\varepsilon_n x$. Assume without loss of generality that $x < x_0$. Because (15) is a continuous function, we know that it must attain its maximum and minimum over the compact set $[x, x_0]$. Let u be that maximum and v be that minimum. Hence we know that

$$\frac{1}{n!}f^{n+1}(v)|x - x_0|^n \leq \frac{1}{n!}f^{n+1}(\hat{x})|x - x_0|^n \leq \frac{1}{n!}f^{n+1}(u)|x - x_0|^n$$

[4]The derivation of the fourth order and higher methods require an advanced knowledge of graph theory [1, 5]

for some $\hat{x} \in [x, x_0]$. However, we want $|x - x_0|^{n+1}$, so we will integrate the above equation over the interval $[x, x_0]$. This gives us

$$\int_x^{x_0} \frac{1}{n!} f^{n+1}(v)|\hat{x} - x|^n d\hat{x} \leq \int_x^{x_0} \frac{1}{n!} f^{n+1}(\hat{x})|\hat{x} - x|^n d\hat{x} \leq \int_x^{x_0} \frac{1}{n!} f^{n+1}(u)|\hat{x} - x|^n d\hat{x}.$$

We can pull out the constant term $\frac{1}{n!}$ and the function evaluations to yeild

$$f^{n+1}(v) \int_x^{x_0} \frac{1}{n!}|\hat{x} - x|^n d\hat{x} \leq \frac{1}{n!} \int_x^{x_0} f^{n+1}(\hat{x})|\hat{x} - x|^n d\hat{x} \leq f^{n+1}(u) \int_x^{x_0} \frac{1}{n!}|\hat{x} - x|^n d\hat{x}.$$

But we know that

$$\int_x^{x_0} \frac{1}{n!}|\hat{x} - x|^n d\hat{x} = \frac{1}{(n+1)!}|\hat{x} - x|^{n+1}\Big|_x^{x_0} = \frac{1}{(n+1)!}|x - x_0|^{n+1}$$

and

$$\int_x^{x_0} \frac{1}{n!} f^{(n+1)}(\hat{x})|\hat{x} - x|^n d\hat{x} = \frac{1}{(n+1)!} f^{(n+1)}(x_0)|x - x_0|^{n+1} = \varepsilon(x).$$

Hence we have that

$$\frac{1}{n+1} f^{(n+1)}(v)|x - x_0|^{n+1} \leq \epsilon_n(x) \leq \frac{1}{(n+1)!} f^{(n+1)}(u)|x - x_0|^{n+1}.$$

However, because we know that $\varepsilon_n(\hat{x})$ is continuous over $[x, x_0]$, we know that by the Intermediate Value Theorem there must be some k to make (15) hold.

Now, to show that $\varepsilon_n(x)$ is $O\big(h^{n+1}\big)$, we need to show that for $h = |x - x_0|$, $\varepsilon_n(x) \leq \lambda^* h^{n+1}$. Because we know that $f^{(n+1)}(\hat{x})$ attains its maximum over $[x, x_0]$, we know that $\big|f^{(n+1)}(\hat{x})\big| \lambda$ for some λ. This implies that $|\varepsilon_n(x)| \leq \frac{\lambda}{(n+1)!} h^{n+1}$. So let $\lambda^* = \frac{\lambda}{(n+1)!}$. For sufficiently small h, we know that

$$|\varepsilon_n(x)| \leq \lambda^* h^{n+1},$$

which implies that

$$\frac{|\varepsilon_n(x)|}{h^{n+1}} \leq \lambda^*,$$

so $\varepsilon_n(x)$ must be $O\big(h^{n+1}\big)$, and by definition has order n. ■

Derivation of a Third Order Runga Kutta Method

The Runge-Kutta methods use a weighted average of the current and forecasted slopes to determine the step direction Φ. Since we can't use normal integration to figure out the integral, our other option is to try and write it as a finite sum. To do this, we turn to the ideas of Gaussian quadrature laid out in [5]. Let η be a nonnegative function in the interval (a, b) so that

$$0 < \int_a^b \eta(\phi)d\phi < \infty, \qquad \left|\int_a^b \phi^j \eta(\phi)d\phi\right| < \infty, \ j \in \mathbb{N}.$$

Then for some function f,

$$\int_a^b f(\phi)\eta(\phi)d\phi \approx \sum_{j=1}^{v} \omega_j f(\theta_j)$$

for some combination of ωs and θs. To use this for a single variable system $y' = f\begin{pmatrix} x \\ y \end{pmatrix}$, we integrate f from x_n to $x_{n+1} = x_n + \Delta x$. So

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f\begin{pmatrix} \phi \\ y(\phi) \end{pmatrix} d\phi = y(x_n) + \Delta x \int_0^1 f\begin{pmatrix} x_n + \phi\Delta x \\ y(x_n + \phi\Delta x) \end{pmatrix} d\phi.$$

We can now turn to quadrature and replace the integral of $f\begin{pmatrix} x_n + \phi\Delta x \\ y(x_n + \phi\Delta x) \end{pmatrix}$ with a sum to get

$$y(x_{n+1}) = y(x_n) + \Delta x \sum_{j=1}^{v} \omega_j f\begin{pmatrix} x_n + \zeta_j\Delta x \\ y(x_n + \zeta_j\Delta x) \end{pmatrix}.$$

However, because the values of $y(x_n + \zeta_j h)$ are unknown for any $j > 1$, we must approximate them. If we let the i^{th} approximation of $\begin{pmatrix} \zeta_j\Delta x \\ y(x_n + \zeta_j h) \end{pmatrix}$ be denoted θ_i, then we have created an iterative process to find the next point in the estimate of the solution. Hence we have that for (14)

$$\Phi\begin{pmatrix} x \\ y \\ \Delta x \end{pmatrix} = \sum_{j=1}^{m} \omega_m k_m,$$

where

$$k_1 = f(\hat{x} + \theta_1 \Delta x),\ k_2 = f(\hat{x} + \theta_2 \Delta x),\ k_3 = f(\hat{x} + \theta_3 \Delta x),$$

where

$$\theta_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \theta_i = \begin{pmatrix} \zeta_i \\ \gamma_{i,1}k_1 + \cdots + \gamma_{m,m-1}k_{m-1} \end{pmatrix}, \quad \text{and} \quad \hat{x} = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$$

where each ζ_j, and $\gamma_{j,i}$ are arbitrary constants. Now, to derive a third order Runge-Kutta method, (as is stated in [1] and [5]) we need the Taylor expansion of the method to be equal to the Taylor expansion of the true solution, up to the $(\Delta x)^4$ term of the solution's Taylor expansion at the point $x + \Delta x$. For the following equations, let T_f^n be the n^{th} order Taylor expansion of f. So, if we expand the solution about the point $x + \Delta x$, we find the following:

$$T_y^3(\hat{x} + \Delta x) = y(x) + \Delta x f(x) + \frac{1}{2}\Delta x^T \nabla f(x) \Delta x + \frac{1}{6}\sum_i \Delta x_i (\Delta x^T \nabla_i^2 f(x) \Delta x) + O\big((\Delta \hat{x})^4\big).$$

However, for a simple one-dimensional system, the Hessian of f can be calculated, and, substituting h for Δx, the Taylor expansion becomes

$$\begin{aligned} T_y^3(x+h) &= y + hf\begin{pmatrix} x \\ y \end{pmatrix} + \frac{1}{2}h^2\left(\frac{\partial}{\partial x}f\begin{pmatrix} x \\ y \end{pmatrix} + f\begin{pmatrix} x \\ y \end{pmatrix}\frac{\partial}{\partial y}f\begin{pmatrix} x \\ y \end{pmatrix}\right) \\ &+ \frac{1}{6}h^3\left(\frac{\partial^2}{\partial x^2}f\begin{pmatrix} x \\ y \end{pmatrix} + 2f\begin{pmatrix} x \\ y \end{pmatrix}\frac{\partial^2}{\partial x \partial y}f\begin{pmatrix} x \\ y \end{pmatrix} + f\begin{pmatrix} x \\ y \end{pmatrix}^2\frac{\partial^2}{\partial y^2}f\begin{pmatrix} x \\ y \end{pmatrix}\right. \\ &+ \left.\frac{\partial}{\partial x}f\begin{pmatrix} x \\ y \end{pmatrix}\frac{\partial}{\partial y}f\begin{pmatrix} x \\ y \end{pmatrix} + f\begin{pmatrix} x \\ y \end{pmatrix}\left(\frac{\partial}{\partial y}f\begin{pmatrix} x \\ y \end{pmatrix}\right)^2\right) + O\left(h^4\right) \end{aligned}$$

Similarly, we can expand the forecasting terms of the Runge-Kutta by the Taylor formula, and find that each of the three terms k_1, k_2, and k_3 expand into their own polynomials.

$$\begin{aligned} T_{k_1}^2(x+\Delta x) = f(\hat{x} + \theta_1 \Delta x) &= f(\hat{x}) \\ T_{k_2}^2(x+\Delta x) = f(\hat{x} + \theta_2 \Delta x) &= f(\hat{x}) + \nabla f(\hat{x})(T_{\theta_2}^1 \Delta x) + \frac{1}{2}(T_{\theta_2}^1 \Delta x)^T \nabla^2 f(\hat{x})(T_{\theta_2}^1 \Delta x) + O\left((\Delta x)^3\right) \\ &= f(\hat{x}) + \Delta x \nabla f(\hat{x})(T_{\theta_2}^1) + \frac{1}{2}(\Delta x)^2 (T_{\theta_2}^1)^T \nabla^2 f(\hat{x})(T_{\theta_2}^1) + O\left((\Delta x)^3\right) \\ T_{k_3}^2(x+\Delta x) = f(\hat{x} + \theta_3 \Delta x) &= f(\hat{x}) + \nabla f(\hat{x})(T_{\theta_3}^0 \Delta x) + \frac{1}{2}(T_{\theta_3}^0 \Delta x)^T \nabla^2 f(\hat{x})(T_{\theta_3}^0 \Delta x) + O\left((\Delta x)^3\right) \end{aligned}$$

$$= \quad f(\hat{x}) + \Delta x \nabla f(\hat{x})(T^0_{\theta_3}) + \frac{1}{2}(\Delta x)^2 (T^0_{\theta_3})^T \nabla^2 f(\hat{x})(T^0_{\theta_3}) + O\left((\Delta x)^3\right)$$

where each θ_i is a vector of weights ζ and γ defined as above. Once again, for a simple one-dimensional system, we can substitute in h for Δx, and the appropriate weights for the elements of θ_2 and θ_3, we find that k_1, k_2, and k_3 are actually as follows. Note that In these equations, $f\begin{pmatrix} x \\ y \end{pmatrix}$ has been simplified to merely f, and the expansion point of the taylor polynomials is understood to be $(x+h)$.

$$\begin{aligned}
T^2_{k_1} &= f \qquad (16)\\
T^2_{k_2} &= f + h\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)\begin{pmatrix} T^1_{\zeta_2} \\ T^1_{\gamma_{2,1}k_1} \end{pmatrix} + \frac{1}{2}h^2 \begin{pmatrix} T^1_{\zeta_2} \\ T^1_{\gamma_{2,1}k_1} \end{pmatrix}^T \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \begin{pmatrix} T^1_{\zeta_2} \\ T^1_{\gamma_{2,1}k_1} \end{pmatrix} + O\left(h^3\right)\\
&= f + h\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)\begin{pmatrix} \zeta_2 \\ \gamma_{2,1}T^1_{k_1} \end{pmatrix} + \frac{1}{2}h^2 \begin{pmatrix} \zeta_2 \\ \gamma_{2,1}T^1_{k_1} \end{pmatrix}^T \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \begin{pmatrix} \zeta_2 \\ \gamma_{2,1}T^1_{k_1} \end{pmatrix} + O\left(h^3\right)\\
&= f + h\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)\begin{pmatrix} \zeta_2 \\ \gamma_{2,1}f \end{pmatrix} + \frac{1}{2}h^2 \begin{pmatrix} \zeta_2 \\ \gamma_{2,1}f \end{pmatrix}^T \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \begin{pmatrix} \zeta_2 \\ \gamma_{2,1}f \end{pmatrix} + O\left(h^3\right)\\
&= f + h\left(\zeta_2 \frac{\partial f}{\partial x} + \gamma_{2,1} f \frac{\partial f}{\partial y}\right) + h^2\left(\frac{1}{2}\zeta_2^2 \frac{\partial^2 f}{\partial x^2} + \zeta_2\gamma_{2,1} f \frac{\partial^2 f}{\partial x \partial y} + \frac{1}{2}\gamma_{2,1}^2 f^2 \frac{\partial f}{\partial y}\right) + O\left(h^3\right) \qquad (17)\\
T^2_{k_3} &= f + h\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)\begin{pmatrix} T^1_{\zeta_3} \\ T^1_{\gamma_{3,1}k_1 + \gamma_{3,2}k_2} \end{pmatrix}\\
&\quad + \frac{1}{2}h^2 \begin{pmatrix} T^1_{\zeta_3} \\ T^1_{\gamma_{3,1}k_1 + \gamma_{3,2}k_2} \end{pmatrix}^T \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \begin{pmatrix} T^1_{\zeta_3} \\ T^1_{\gamma_{3,1}k_1 + \gamma_{3,2}k_2} \end{pmatrix}^T + O\left(h^3\right)\\
&= f + h\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)\begin{pmatrix} \zeta_3 \\ \gamma_{3,1}T^1_{k_1} + \gamma_{3,2}T^1_{k_2} \end{pmatrix}\\
&\quad + \frac{1}{2}h^2 \begin{pmatrix} \zeta_3 \\ \gamma_{3,1}T^1_{k_1} + \gamma_{3,2}T^1_{k_2} \end{pmatrix}^T \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \begin{pmatrix} \zeta_3 \\ \gamma_{3,1}T^1_{k_1} + \gamma_{3,2}T^1_{k_2} \end{pmatrix}^T + O\left(h^3\right)\\
&= f + h\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)\begin{pmatrix} \zeta_3 \\ \gamma_{3,1}f + \gamma_{3,2}\left(f + h\left(\zeta_2 \frac{\partial f}{\partial x} + \gamma_{2,1} f \frac{\partial f}{\partial y}\right) + O\left(h^2\right)\right) \end{pmatrix}
\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{2}h^2 \begin{pmatrix} \zeta_3 \\ \gamma_{3,1}f + \gamma_{3,2}\left(f + O(h)\right) \end{pmatrix}^T \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \begin{pmatrix} \zeta_3 \\ \gamma_{3,1}f + \gamma_{3,2}\left(f + O(h)\right) \end{pmatrix}^T + O\left(h^3\right) \\
= \quad & f + h\left(\zeta_3 \frac{\partial f}{\partial x} + (\gamma_{3,1} + \gamma_{3,2}) f \frac{\partial f}{\partial y}\right) + h^2 \Bigg(\frac{1}{2}\zeta_3^2 \frac{\partial^2 f}{\partial x^2} + \zeta_3(\gamma_{3,1} + \gamma_{3,2}) f \frac{\partial^2 f}{\partial x \partial y} \\
& + \frac{1}{2}(\gamma_{3,1} + \gamma_{3,2})^2 f^2 \frac{\partial^2 f}{\partial y^2} + \gamma_{3,2}\left(\zeta_2 \frac{\partial f}{\partial x} + \gamma_{2,1} f \frac{\partial f}{\partial y}\right)\Bigg) + O\left(h^3\right). \qquad (18)
\end{aligned}
$$

Note that these equations are for a general derivation of a 3^{rd} order Runge-Kutta method in one variable, and that the numbered equations match those found in [1]. Now, we can use these values in the function for Φ in the Runge-Kutta algorithm, and we find that

$$T_{y(x+h)} = y + h\left(\omega_1 T_{k_1} + \omega_2 T_{k_2} + \omega_3 T_{k_3}\right),$$

and if we set the coefficients equal to each other, we find that we only have to solve a simple system of equations.

$$
\begin{aligned}
\omega_1 + \omega_2 + \omega_3 &= 1, && (19)\\
\zeta_2\omega_2 + \zeta_3\omega_3 &= \frac{1}{2}, && (20)\\
\gamma_{2,1}\omega_2 + (\gamma_{3,1} + \gamma_{3,2})\omega_3 &= \frac{1}{2}, && (21)\\
\frac{1}{2}\zeta_2^2\omega_2 + \frac{1}{2}\zeta_3^2\omega_3 &= \frac{1}{6}, && (22)\\
\zeta_2\gamma_{2,1}\omega_2 + \zeta_3(\gamma_{3,1} + \gamma_{3,2})\omega_3 &= \frac{1}{3}, && (23)\\
\frac{1}{2}\gamma_{2,1}\omega_2 + \frac{1}{2}(\gamma_{3,1} + \gamma 3,2)^2\omega_3 &= \frac{1}{6}, && (24)\\
\zeta_2\gamma_{3,2}\omega_3 &= \frac{1}{6}, && (25)\\
\gamma_{2,1}\gamma_{3,2}\omega_3 &= \frac{1}{6}. && (26)
\end{aligned}
$$

From equations 25 and 26, we can immediately see that $\zeta_2 = \gamma_{2,1}$. By substituting ζ_2 for $\gamma_{2,1}$ throughout the system, we can then see that from 20 and 21 that $\zeta_3 = \gamma_{3,1} + \gamma_{3,2}$. Now, by making these substitutions in equations 23 and 24, we find that

$$\zeta_2\omega_2 + \zeta_3^2\omega_3 = \zeta_2^2\omega_2 + \zeta_3^2\omega_3.$$

Hence we have that $\zeta_2 = \zeta_2^2 = 1$. Notice that $\zeta_2 \neq 0$ due to (25). This leaves us in a system of four equations.

$$\begin{aligned} \omega_1 + \omega_2 + \omega_3 &= 1 && (27) \\ \omega_2 + \zeta_3\omega_3 &= \frac{1}{2} && (28) \\ \omega_2 + \zeta_3^2\omega_3 &= \frac{1}{3} && (29) \\ \gamma_{3,2}\omega_3 &= \frac{1}{6} && (30) \end{aligned}$$

If we now examine equations 20 and 21, we find that we can solve each for ω_2 and set them equal, giving

$$\frac{1}{2} - \zeta_3\omega_3 = \frac{1}{3} - \zeta_3^2\omega_3.$$

We can solve for ζ_3, and using the quadratic equation gives us

$$\zeta_3 = \frac{\omega_3 \pm \sqrt{\omega_3^2 - \frac{2}{3}\omega_3}}{2\omega_3}.$$

For these to be real solutions, we know that $\omega_3^2 \geq \frac{2}{3}\omega_3$, so $\omega_3 \geq \frac{2}{3}$. For simplicity, let $\omega_3 = \frac{2}{3}$. Putting this back into the second system yeilds values of $\frac{1}{2}$ and $\frac{1}{4}$ for ζ_3 and $\gamma_{3,2}$ respectively. We can then substitute these into the original system to find that the remaining parameters.

Parameter	Value
α_2	1
α_3	$\frac{1}{2}$
$\gamma_{2,1}$	1
$\gamma_{3,1}$	$\frac{1}{4}$
$\gamma_{3,2}$	$\frac{1}{4}$
ω_1	$\frac{1}{6}$
ω_2	$\frac{1}{6}$
ω_3	$\frac{2}{3}$

Using these, we find that we have an overall 3^{rd} degree Runge-Kutta formula of

$$y_{i+1} = y_i + \frac{1}{6}k_1 + \frac{1}{6}k_2 + \frac{2}{3}k_3,$$

where

$$\begin{aligned} k_1 &= f(x_i, y_i), \\ k_2 &= f(x_i + h, y_i + hk_1), \\ k_3 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{4}hk_1 + \frac{1}{4}hk_2\right). \end{aligned}$$

Existence and Uniqueness

Based on the Existence and Uniqueness Theorem for differential equations [2], we know that each equation has its own solution in the first quadrant. Because the system is made up of rational functions with strictly positive coefficients, we know that they must be defined over the entire first quadrant, axes not included. Because we start at time $t = 0$, the ordinate (y-axis) is included, and because the denominators of the rational functions are modeled to always contain a constant term, it is never possible for the system to be undefined. Therefore, we know that the system is not only continuous, but its partial derivative with respect to each y_i is also continuous over the entire first quadrant. Hence we have that a solution curve exists within the first quadrant and that the solution is unique.

Natural Responses

If we examine the solution for y_i assuming that all other variables are zero, the natural response of y_i, then we find that one of four things can happen.

1. The natural response of y_i is a constant differential equation. The obvious solution to this is a linear function with slope f_i and y-intercept y_{i_0}. This reponse occurs only for y_7, as its degredation term is dependent on both the levels of y_7 and y_8.

$$y_i(t) = f_i t + y_{i_0}$$

2. The natural response of y_i is a quadratic differential equation. The solution curve for this type of differential equation is a hyperbolic tangent multiplied by a constant. This response only occurs for

y_8, as its degredation term is dependent on the square of its own level.

$$y_i(t) = \tanh\left(t\sqrt{e_i f_i}\right)\sqrt{\frac{f_i}{e_i}}$$

3. The natural response of y_i is for the level to drop to zero or for nothing to happen, given by a linear or constant zero differential equation. These are cases where the formation terms are dependent on the concentrations of other proteins, so none is made. If there is none already present, then none is made. If there is some present, then it will be degraded.

$$y_i(t) = 0 \qquad \text{or} \qquad y_i(t) = y_0 \mathbf{e}^{-e_i t}$$

4. The natural response of y_i is given by an affine differential equation. The solution curves for these proteins rises or falls exponentially to an equilibrium given by the ratio between their formation and degredation rates.

$$y_i(t) = -\frac{y_{i_0}}{e_i}\mathbf{e}^{-e_i t} + \frac{f_i}{e_i}$$

Equilibrium Solutions

It has been shown in the Poincaré-Bendixson Theorem [4] that in a two-dimensional system, each solution is going to fall into one of three classes.

(a) The solution is an equilibrium point.

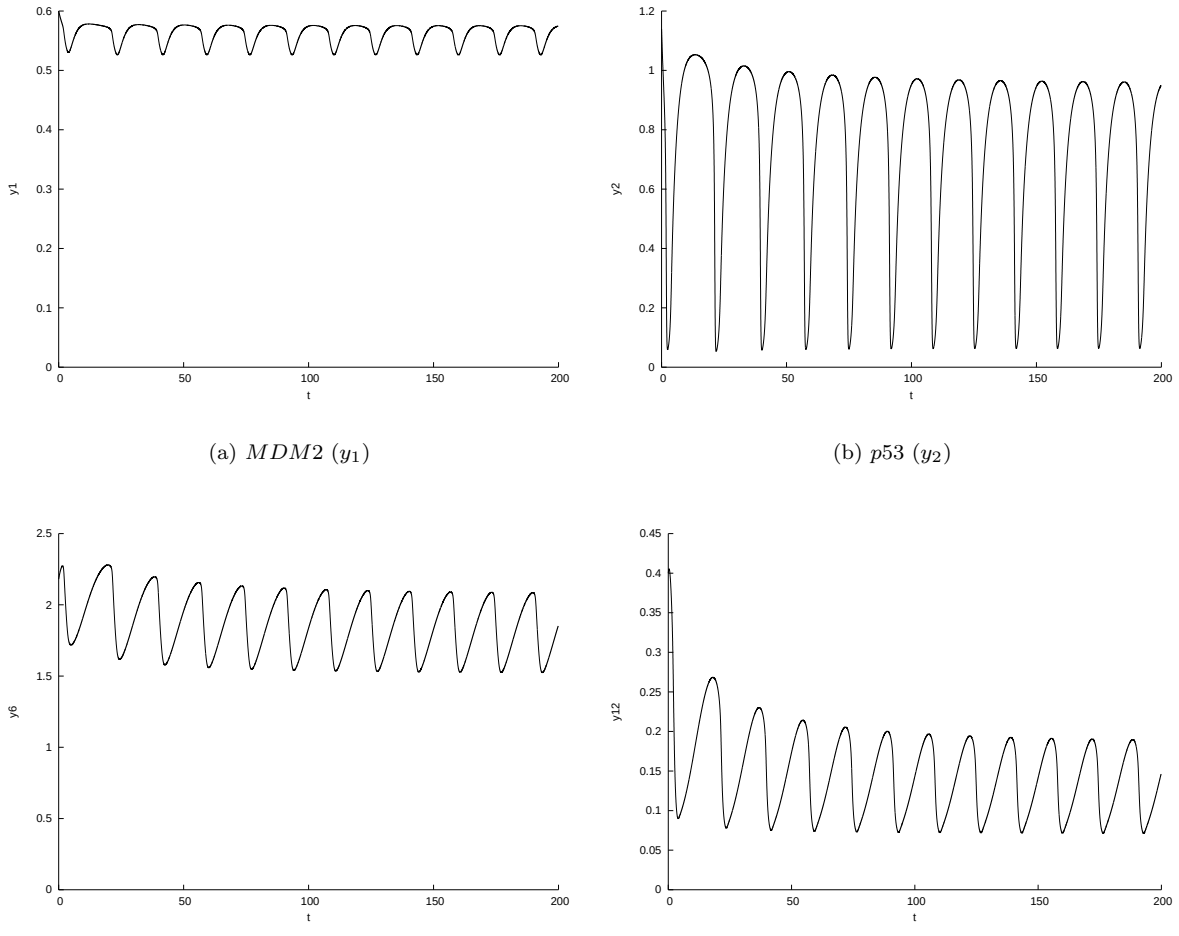
(b) The solution collapses to or spirals out from an equilibrium point.

(c) The solution falls into a periodic orbit around an equilibrium point.

However, in any system with more than two dimensions, it has been proven that there is no similar theorem. Hence, we must turn to algebraic or numerical techniques to determine equilibrium solutions for the system, if any exist. Using Maple 9, we attempted to find an exact solution to the system, with the parameter values set to those for which the model gives the expected qualitative results. However, after 12 days and 4 hours Maple was unable to find such a solution. Failing this, we tried the numerical algorithms of the GNU Octave software package. Using this package, $100,000$ different initial values within the expected range of the variables were determined, and the relative zeros for these starting values were numerically determined. As

these zeros were found, Octave screened them for strictly positive values. Over the entire range of $100,000$ different initial values, no feasible roots were found. This implies that the system is continuously changing, and never reaches equilibrium, which is a biologically sound idea. If the cell ever reached equilibrium, it would simply cease growing and changing. However, due to both external and internal influences, cells are always growing and always changing.

Solution Curves and Biological Significance



(a) $MDM2$ (y_1)

(b) $p53$ (y_2)

(c) $p21$ (y_6)

(d) 14-3-3 (y_{12})

Figure 1: Cell Arrest proteins

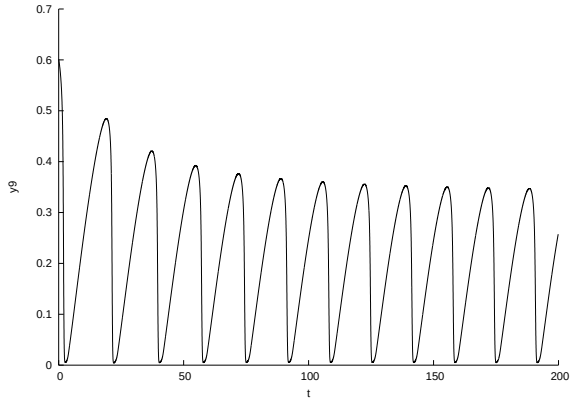
The two primary groups of proteins responsible for the viability of any given cell are the protein involved in the cell arrest pathway and the protein involved in the cell cycle engine. The idea behind the cell cycle

engine is to allow enough time for the DNA to completely replicate before the cell begins division. If this does not happen, then at least one of the two daughter cells will inherit an incomplete copy of the DNA and die. With this in mind, there are several things that can go wrong during the duplication of DNA. The primary downfall is the constant bombardment of each cell by high-intensity radiation. Although this radiation is for the most part minimal, it can still cause damage to the genomic structure which must be fixed. Depending on the severity of the damage, the cell cycle may be temporarily arrested, or the cell may simply kill itself.

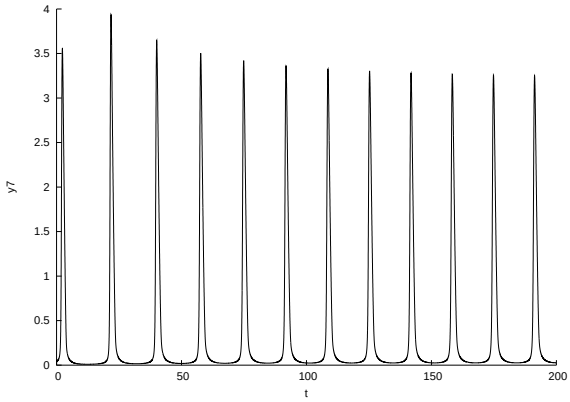
The proteins featured in Figure 1 are the primary cell cycle arrest proteins. Their function is to recognize damage to the genome and respond accordingly,temporarily halting the cell cycle so the damage can be repaired. The solution for the most important protein in this pathway, p53 (y_2), is shown as Figure 1(b). p53 is the beginning of the entire pathway. As is shown in Figure 4(c), as the levels of p53 rise, it triggers an increase in the levels of p21 (y_6) and 14-3-3 (y_{12}). As these increase, they slow down the production of cyclin A-K1 (y_9), the solution graph of which is shown in Figure 2(a), and MPF (y_7), the solution graph of which is shown in Figure 2(b). As the concentration of 14-3-3 increases, it prevents the activation of CDC25c (y_{10} inactive and y_{11} active) , as is shown in Figure 3(d). This indirectly slows down the positive feedback loop to activate MPF, which is shown in Figure 3(c).

The main proteins in the cell cycle engine are feature in Figure 2. The central protein in this interaction, and in the entire system is MPF. At concentrations of cyclin A-K1 lower than a threshold, the concentration of MPF is at a minimum. However, once cyclin A-K1 crosses that threshold, it overcomes the damping effect of 14-3-3 and activates CDC25c. This in turn causes pMPF to become MPF, which activates more CDC25c, which creates a positive feedback loop, as shown in Figure 3(c). This feedback loop and rapid activation can be seen as the sharp peaks in the solution graphs of MPF and CDC25c (Figures 2(b) and 3(b) respectively). Part of what allows this to happen is the buildup of an inventory of p-MPF (phosphorylated and inactive MPF, y_8). As the feedback loop occurs, the concentration of MPF must quickly spike—much more quickly than new MPF can be produced. At the same time however, MPF is responsible for resetting the model to the beginning of the next cell cycle. So, as MPF rises, it causes an increase in the degredation rates of cyclin A-K1 and itself, forcing them back to the starting levels. Withouth the cyclin A and the MPF to keep it activated, the CDC25c is rapidly inactivated, as can be seen in the solution curve of the inactive form (Figure 3(a)).

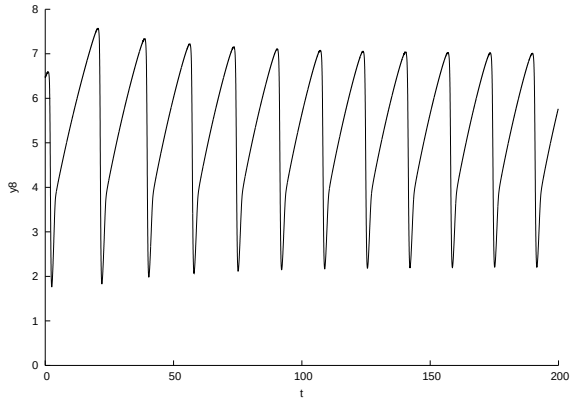
Despite the efforts to repair genomic damage, it is not uncommon for cells to suffer enormous amounts of damage and kill themselves. Although it is not yet clear exactly what mechanism starts this suicidal



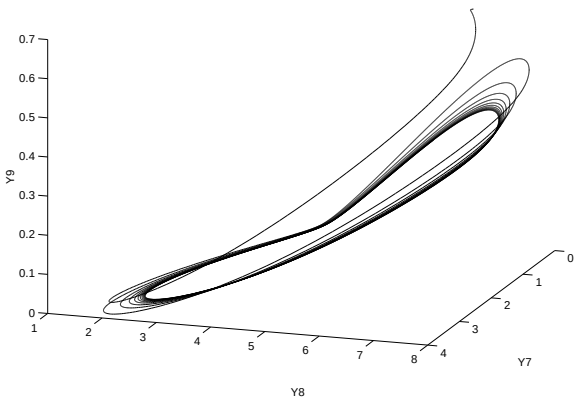
(a) Cyclin A-K1 (y_9)



(b) MPF (y_7)



(c) pMPF (y_8)



(d) Total interaction

Figure 2: Cell cycle Engine

pathway, several key players have been located, and included in this model. The first is a special form of p53, known as p53+ (y_3). It is believed that on severe radiation-based genomic damage, p53 is turned into p53+, which then goes on to activate the cell death pathway. Also included in our model is the evidence that MPF may have a role in the conversion of p53 to p53+. Due to the way this is modeled, a sharp increase in the level of p53+ can be seen at the end of each cell cycle, as is evidenced in Figure 4(a). In the model, the only role given to p53+ is to activate a group of proteins known as the caspases (y_5). The caspases themselves do all of the "dirty work," breaking down the machinery inside the cell, and eventually breaking open the protective membrane. Due to the increase in p53+, we noticed that the concentration of caspases

also increased at the time of cell division, although not by a large enough amount to destroy the cell on their own. This increase can be seen in Figure 4(b).

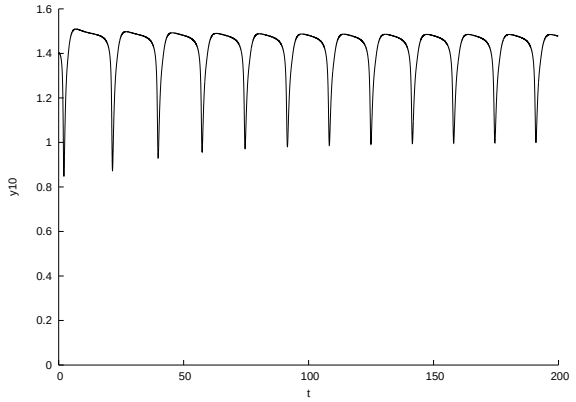
The above facts were taken from [3] and references therein.

Conclusion

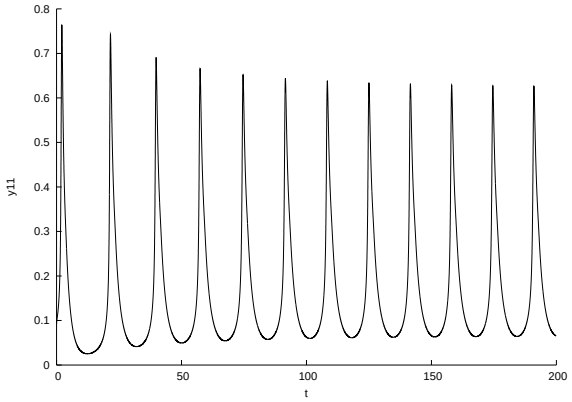
From the data discussed here, the model presented by DeHoff and Obeyesekere [3] is mathematically and biologically sound. The most interesting aspect of this model mathematically is the distinct lack of equilibrium solutions. However, this is is biologically sound concept, as it does not make sense for a cell ever to be at complete equilibrium. This is currently a highly simplified model, and could be expanded in the future to include more of the cell cycle. Additionally, more simulations should be run to determine bifurcation points for each of the parameters as well as a characterization of the effects introducing damage would have on the system. This characterization would compromise studying the effects different values for α and β in equation 13 have on the qualitative results of the model.

Acknowledgements

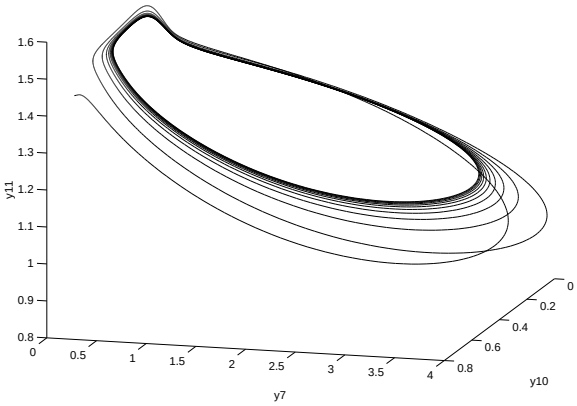
I would like to thank Dr. Obeyesekere for her mentorship during the formulation and initial testing of this model. Additionally, I would like to thank Dr. Holder, Dr. Ponomarenko, and Dr. Elyadi for being members of this committee.



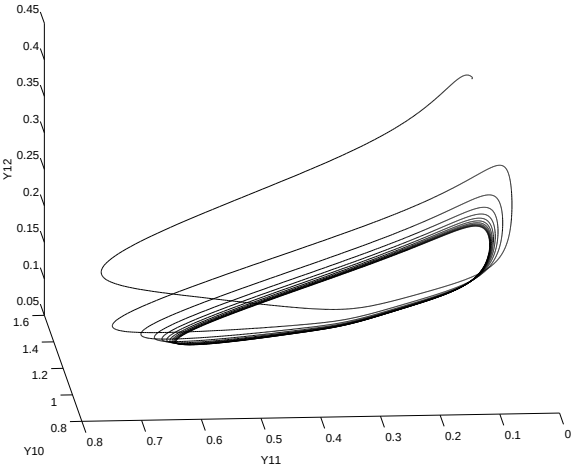
(a) CDC25c - inactive (y_{10})



(b) CDC25c - active (y_{11})

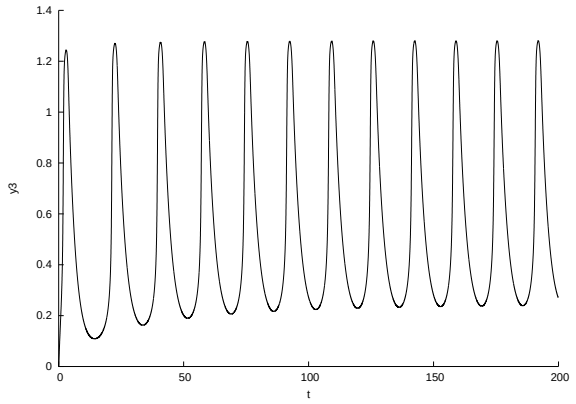


(c) MPF, CDC25c active and inactive

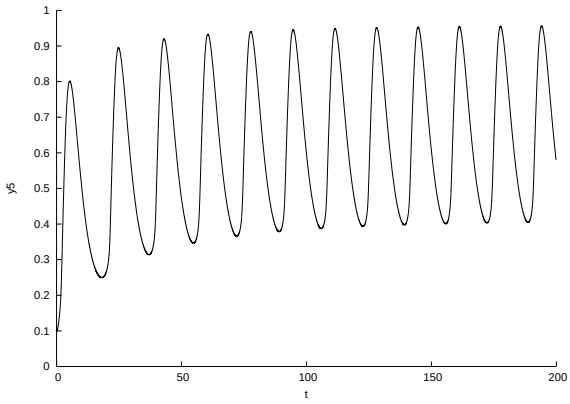


(d) 14-3-3 and CDC25c active and inactive

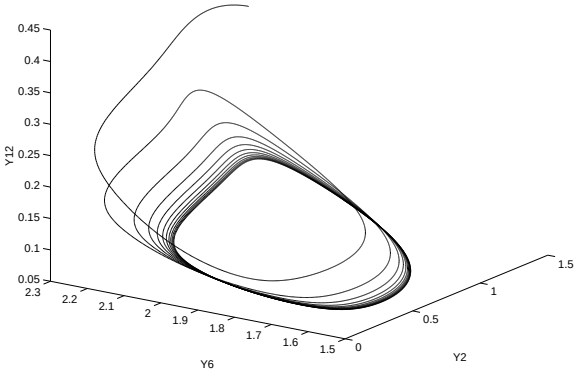
Figure 3: MPF positive feedback loop



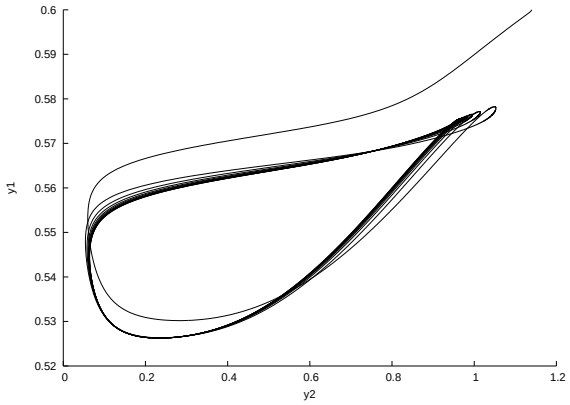
(a) p53+ (y_3)



(b) Caspases (y_5)



(c) p53, p21, 14-3-3



(d) MDM2 and p53

Figure 4: Apoptotic proteins and other interactions

Bibliography

[1] Ivo Babuska, Milan Práger, and Emil Vitásek. *Numerical Processes in Differential Equations*. Interscience Publishers, 1966.

[2] Paul Blanchard, Robert L. Devaney, and Glen R. Hall. *Differential Equations, Second Edition*. Brooks/Cole, 2002.

[3] Kevin DeHoff and Mandri Obeyesekere. p53 dependent apoptosis model. *Unknown*, Unpublished.

[4] Po-Fang Hsieh and Yatsutaka Sibuya. *Basic Theory of Ordinary Differential Equations*. Springer-Verlag, 1999.

[5] Arieh Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, 1996.