Project Number: MA-RYL-0506

A One-Dimensional Viscoelastic Cell Motility Model

A Major Qualifying Project

submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Andrew Port

Jeremy Ristau

April 25, 2007

Approved

Professor Roger Y. Lui
Major Advisor

Abstract

This project attempts to model the length, velocity, and internal stress experienced by a crawling cell as it moves on a substrate. We assume the cell's viscoelastic properties can be described by a Maxwell element. Through balance equations, we develop a Moving Boundary Problem. We solve this MBP numerically, as well as analyze its traveling wave solution. We then change our model to assume that the cell's actin concentration satisfies a second MBP and discuss our future plans for solving this new, more complicated model.

Acknowledgments

We would like to thank Professor Roger Lui of the Mathematical Sciences Department for his continuous advice and assistance throughout this project. We would also like to thank Kevin Chu of Princeton University for his ideas and direction.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Understanding the mechanics of cell crawling is necessary for understanding many cellular processes, including such natural phenomena as immune responses, wound healing, and cancer cell metastasis.

One of the most important examples of cell motility is available in our own immune response to bacterial infection. To fight off invading antigens the immune system increases production of white blood cells, which travel throughout the body looking for anything foreign. Once found, the foreign bacteria are actually engulfed by the white blood cells and are broken down using powerful digestive enzymes.

There can be down sides to cell movement as well. Motility is essential in the formation of new capillaries during angiogenesis, and during the metastasis of tumor cells. The understanding of cell motility is an essential part of the fight against many cancers and other such diseases.

Cellular migration is a dynamic process, which requires many internal mechanisms to work in unison. The motility of many eukaryotic cells can roughly be described as a cycle of protrusion, adhesion and contraction. A cell moving along a substrate continuously extends in the directional front due to actin filaments elongating against the plasma membrane in a certain area of the cell. This "front" edge of the cell is secured to the substrate by chemical bonds, while the opposite "rear" edge breaks previously formed substrate adhesions. Myosin motors attached to the cell's cytoskeleton contract to create an internal stress dynamic across the plasma membrane. This causes the newly-freed rear to be pulled in the direction of the front.

To accurately model this process is not an easy task. There are many unknown factors which must be described and analyzed. Some of these are the cell's viscoelastic properties, how the cell determines direction, and how the cell knows where to internally transport needed materials during locomotion.

Recently, progress has been made in understanding cell motility. Detailed research can now be found analyzing the cellular mechanics of protrusion, adhesion, and contraction, as well as the structural characteristics and protein concentrations of migrating cells.

One useful application of cell motility research has come from Dr. J.P. Trinkaus on embryonic development (Browder *et al.*, 1991). During embryonic development cells must move themselves to the correct places where they can form tissues and organs. However, cell motility has been largely studied only in cultured fibroblasts, and few embryonic cells are as amenable to investigation as cells in culture. Although, the deeply embedded cells of fish blastula are an exception because the fish embryo is transparent so individual cell behavior can be observed; and Dr. Trinkaus is one of the leading researchers on the embryonic cells of fish blastula. His studies have shown that the fibroblast model provides a good approximation of embryonic cell motility. This lends itself to even more areas of biological research and application that were before untouchable.

These studies are due in large part to the cooperation of the biological and mathematical science fields. Advances such as using traction force microscopy to measure cell-substrate traction fields in fibroblasts and in H-ras transformed NIH 3T3 cells ([3]; Munevar *et al.*, 2001a,b) have made a higher level of research and analysis possible. This showed that individual fibroblasts display a complex spatial distribution of traction forces throughout the cell.

An accurate mechanical model that describes the viscoelastic properties of a crawling cell, however, has yet to be discovered. Several models have been studied and compared with recorded data. Notably among them include the Kelvin-Voight model [6], the Standard Linear (SL) model, and the Standar Solid (SS) model (Palsson and Othmer, 2000).

The Kelvin-Voight model is a one-dimensional continuum model of amoeboid cell motility which represents the cytoplasm's viscoelastic properties as a spring and dashpot in parallel. Gracheva and Othmer hypothesized that the active stress generated by molecular motors was controlled by extracellular signals related to cell-substrate interaction. The model showed a relationship between cell velocity and cell-substrate interaction consistent with experimental observations [8].

The Standard Linear Solid has been used as a two-dimensional model to approximate the movement of Dd cells. Based on assumptions regarding cell-cell and cell-substrate interactions, the model accurately predicted much of the behavior observed in two-dimensional Dd slugs, including directional change due to external stimulus [6].

This MQP attempts to model in one dimension the stresses inside a fibroblast cell as it moves along a substrate. We assume the cell is a Maxwell material; that is, we use a Maxwell element to model the interaction between the cell and the substrate. From this model assumption and through mass balance laws we develop a Moving Boundary Equation.

Chapter 2

Biological Background

As previously mentioned, fibroblast cells crawl on a substrate using three mechanical processes: protrusion, adhesion and contraction. Actin polymers located in the front of the cell form the cell membrane into protrusions that push the cell's leading edge forward. This puts the plasma membrane under stress. The cell reduces this stress by weakening its adhesion in the rear. It accomplishes this by controlling the concentrations of certain protein triggers inside the cell. The weakened adhesion causes the rear of the cell to lose traction and become mobile. The rear is pulled forward by a collection of stresses. These are the stress along the cell, the contraction of internal myosin motors, and by a phenomenon called treadmilling. For more in-depth information about these cellular properties, we refer the reader to [1].

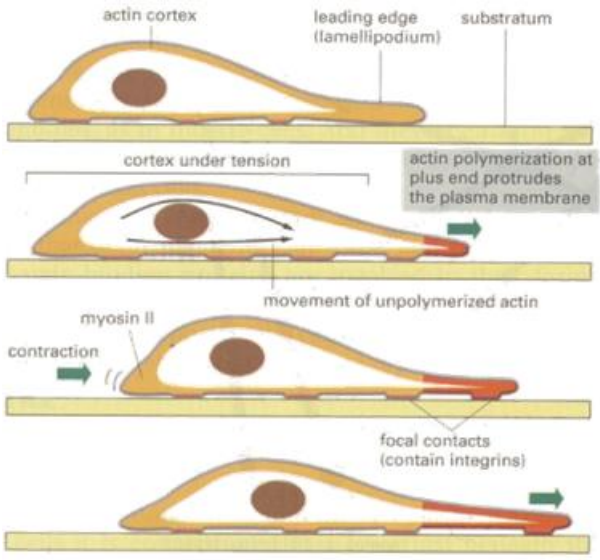

Figure 2.1: Stages in cell movement[1]

2.1 Actin

Actin is a globular protein structure which is found inside the cell as both free monomers floating inside the cell (G-actin), and as polymer structures known as filaments (F-actin). F-Actin is responsible for the structural integrity of the cytoskeleton, and is constantly being constructed and broken down. F-actin filaments are found in all eukaryotic cells and are essential for many of their movements, especially those involving the cell surface. G-actin is used as a building material for existing actin filaments, and in concentration allows for the easy creation of new filaments.

To compensate for the drop in concentration of G-actin during filament formation, filaments are broken down into G-actin at the rear of the cell. This keeps the G-actin concentration at a suitable level in the cell to allow for locomotion when it is needed. Since most filament construction happens at the front of the cell, and most decomposition happens at the rear, there is a concentration gradient of actin across the cell.

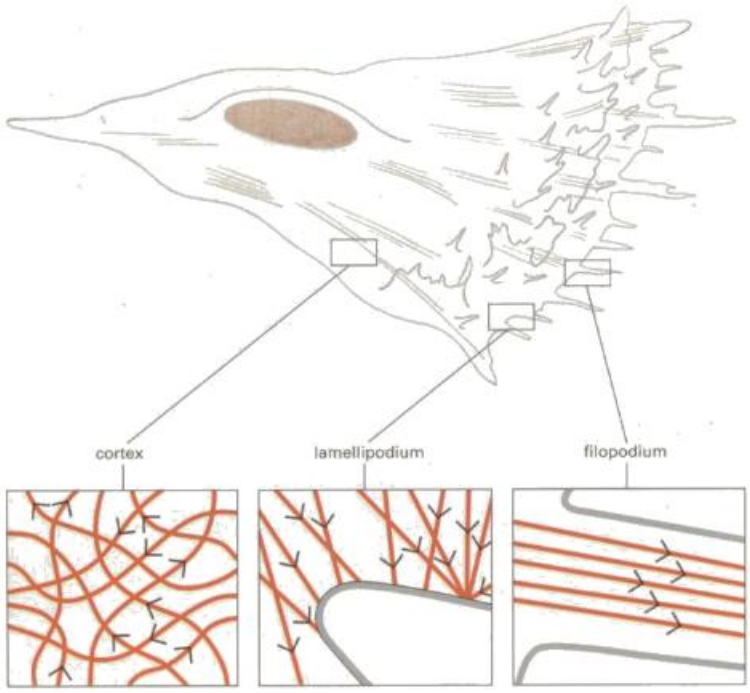

Figure 2.2: Actin filaments in different areas of a moving cell[1]

Actin filaments react with a variety of actin-binding proteins that allow the filaments to serve several functions. Depending on the protein, actin filaments can form rigid structures, or - more importantly to us - they can form temporary structures (e.g. the protrusions found at the leading edge of a crawling fibroblast).

Actin filaments are polar, having a fast growing "barbed" end and a slower growing "pointed" end. This allows for the two-dimensional expansion of filaments and development of a definable

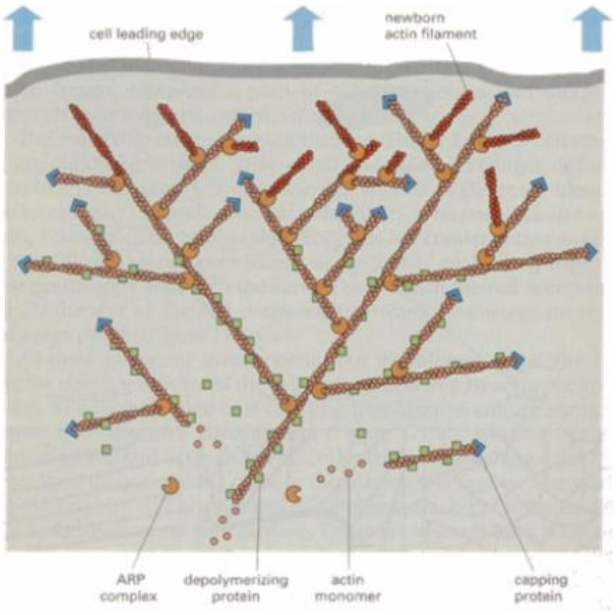

Figure 2.3: Actin structure at the cell's leading edge[1]

end of a filament's growth. When the polymerization rate at the "barbed" end equals the depolymerization rate at the "pointed" end, a treadmilling effect can be observed.

2.2 Treadmilling

The cell itself is an elastic solid, so it does not necessarily move uniformly throughout the body. To accurately describe a cell's movement and stresses this non-uniformity must be considered. One explanation of this is known as treadmilling.

Actin filaments in the front of the cell are constantly being lengthened by addition of G-actin, and this association takes place at a certain rate. Similarly, at the rear of the cell there is a breakdown of filament with the removal of G-actin, which also occurs at a certain dissociation rate. Since these are different chemical processes, their rates are not necessarily equal. These different rates allow for the length of the cell to vary with time.

Figure (2.4) is a simple representation of how treadmilling works. G-actin is removed from the rear of the cell and flows in the cytoplasm to the front of the cell. There it is added to the end of an F-actin filament. Fixing one G-actin monomer we can see lateral movement of the cell body through time.

Note that in cellular movement the rates of growth in the front and rear may not be equal, as they are in the diagram.

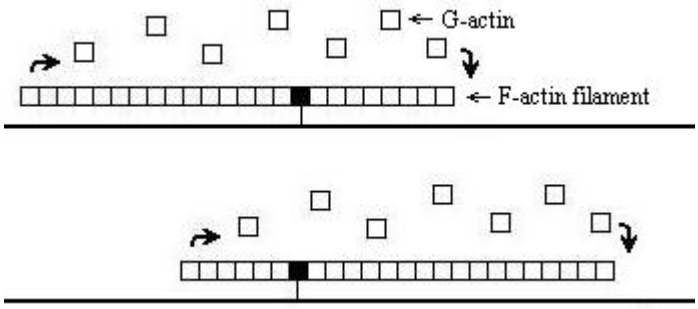

Figure 2.4: A simple graphical representation of treadmilling

2.3 Myosin II

Myosin motors are proteins that convert chemical energy into mechanical energy when they contract. They attach themselves to actin filaments in the cell so that when they contract, the plasma membrane also contracts.

Myosin II, responsible for skeletal muscle contraction, is perhaps the best-studied of all myosin proteins. In muscle cells, it is Myosin II that is responsible for producing the contractile force. Here, the long coiled tails of the individual molecules join together, forming thick filaments called the sarcomere. The force-producing heads stick out from the side of the thick filament, ready to move along the adjacent actin-based filaments in response to the proper chemical signals.

2.4 Integrin

Integrin receptors are transmembrane proteins responsible for the cell's adhesion to a substrate. When the cell membrane touches down on a favorable strip of substrate, the integrin located throughout the membrane adhere to molecules on the strip (e.g. The surface of another cell over which the moving cell is crawling). Conversely, if the substrate is unfavorable to the cell, the integrin will not adhere to it. Meanwhile, on the internal face of the membrane integrin capture actin filaments, creating a robust anchorage for the internal actin structure of the cell.

Chapter 3

Physical Background

3.1 Properties of Materials

Stress
Stress (σ) is the internal distribution of force per unit area that balances and reacts to external loads applied to a body. The simplest definition of stress is

$$\sigma = \frac{F}{A} \tag{3.1}$$

where F is the previously mentioned force and A is the cross-sectional area of the body perpendicular to the direction of F.

The SI unit for stress is Pascal (Pa).

Strain
Strain (ϵ) is the geometrical expression of deformation caused by stress on the cell. Strain therefore expresses itself as a change in size and/or shape, and is represented as the change in length with respect to the length at equilibrium.

$$\epsilon(x,t) = \frac{\partial u}{\partial x} \tag{3.2}$$

Strain has no units since the units of displacement and position are equal, so they cancel.

Elastic Modulus
Elastic Modulus (also known as Young's modulus or tensile modulus) is a measure of the stiffness of a material. For small strains, it is defined as the ratio of stress and strain:

$$E = \frac{\sigma}{\epsilon} \tag{3.3}$$

The SI unit of modulus of elasticity is Pascal.

Viscous Drag Coefficient
Viscous drag is a resistant force created by movement across a material. This form of drag is appropriate for very small objects moving in a fluid at slow speed, which is our situation.

$$F_d = -bv \tag{3.4}$$

where F_d is the drag force, v is the object's velocity, and b is a constant that depends on the properties of the fluid and the dimensions of the object. Since it is a force, the SI unit is Newton (N).

Contractile Stress
We denote τ to be the contractile, or active stress in the interior of the cell. The unit of contractile stress is Pascal.

3.2 Conservation Laws

Our cell is also governed by some fundamental physical laws that we will use to our advantage. The Laws of Conservation of Mass and Momentum are two very important properties used in our model's construction.

Conservation of Mass
The Law of Conservation of Mass is defined as (Fung, 1994):

$$\frac{D\rho}{Dt} + \rho\frac{\partial v}{\partial x} = 0 \tag{3.5}$$

where ρ is the cell's density, and the material derivative is defined as

$$\frac{D\phi}{Dt} = \frac{d\phi}{dt} + \frac{d\phi}{dx}\ . \tag{3.6}$$

Conservation of Momentum
The Law of Conservation of Momentum is mathematically defined as (Fung, 1994):

$$\rho\frac{Dv}{Dt} = \frac{\partial\sigma}{\partial x} + X \tag{3.7}$$

where X is the body force per unit volume.

3.3 1-D Viscoelastic Models

A spring is an elastic object used to store mechanical energy. Springs are stiff, meaning they deform and return to their original position very quickly. Thus, they are said to obey Hooke's law:

$$\gamma_S = E\eta_S \tag{3.8}$$

where γ_S is the spring's stress, E is the spring's modulus of elasticity, and η_S is the spring's strain.

A dashpot is a device that allows the stress and strain of an element to become time-dependent, and is analogous to a shock absorber for a car. The stress-strain relationship of the dashpot is represented by

$$\gamma_D = \mu \frac{d\eta_D}{dt} \tag{3.9}$$

where γ_D is the dashpot's stress, μ is the dashpot's viscosity, and η_D is the dashpot's strain.

The following models use different configurations of a spring and dashpot to represent viscoelasticity. The stiff spring allows for the representation of instantaneous movement. Through the introduction of a dashpot, they take into account that the relaxation occurs over time.

Maxwell Model
The Maxwell model can be represented by a dashpot and spring connected in series as shown in Figure 3.1.

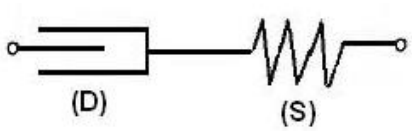

Figure 3.1: A diagram of a Maxwell Element

Under an applied axial stress, the total stress, σ, and the total strain, ϵ, can be defined as follows:

$$\begin{aligned} \sigma &= \gamma_D = \gamma_S \\ \epsilon &= \eta_D + \eta_S \end{aligned} \tag{3.10}$$

Differentiate (3.10b) with respect to t. Substitute (3.8) and (3.9). Finally use (3.10a) to obtain:

$$\frac{\partial \epsilon}{\partial t} = \frac{1}{E}\frac{\partial \sigma}{\partial t} + \frac{\sigma}{\mu} \tag{3.11}$$

Kelvin-Voight Model
The Kelvin-Voight (KV) model can be represented by a dashpot and a spring connected in parallel, as shown in Figure 3.2.

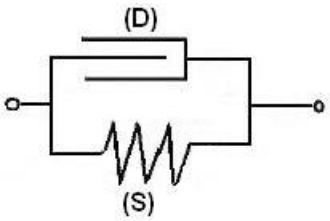

Figure 3.2: A diagram of Kelvin-Voight element

Under an applied axial stress, the total stress, σ, and the total strain, ϵ, can be defined as follows:

$$\begin{aligned} \sigma &= \gamma_D + \gamma_S \\ \epsilon &= \eta_D = \eta_S \end{aligned} \tag{3.12}$$

Differentiate (3.12a) with respect to t. Substitute (3.8) and (3.9). Finally use (3.12b) and re-integrate with respect to t and obtain:

$$\sigma = E\epsilon + \mu\frac{\partial\epsilon}{\partial t} \tag{3.13}$$

Standard Linear Model
The Standard Linear (SL) Model uses a spring connected in parallel with a Maxwell element to model the behavior of a viscoelastic material, as shown in Figure 3.3.

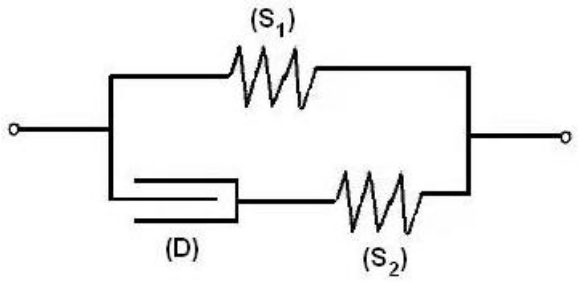

Figure 3.3: A diagram of a Standard Linear element

The spring is represented by Equation (3.8):

$$\gamma_{S_1} = E_1\eta_{S_1} \tag{3.14}$$

Since this is equivalent to a KV model where the dashpot is a Maxwell element we know that γ_M, η_M are the total stress and strain of the Maxwell element represented by

$$\begin{aligned} \gamma_M &= \gamma_D = \gamma_{S_2} \\ \eta_M &= \eta_D + \eta_{S_2} \end{aligned} \tag{3.15}$$

and the relationship between them is

$$\frac{\partial\eta_M}{\partial t} = \frac{1}{E_2}\frac{\partial\gamma_M}{\partial t} + \frac{\gamma_M}{\mu}\,. \tag{3.16}$$

Under an applied axial stress, the total stress, σ and the total strain, ϵ can be defined for the SL model as follows:

$$\begin{aligned} \sigma &= \gamma_M + \gamma_{S_1} \\ \epsilon &= \eta_M = \eta_{S_1}\,. \end{aligned} \tag{3.17}$$

Differentiate (3.17b) with respect to t. Substitute (3.16) and (3.14). Now use (3.17a) to obtain:

$$\frac{\partial \epsilon}{\partial t} = \frac{1}{E_2}\frac{\partial}{\partial t}(\sigma - \gamma_{S_1}) + \frac{\sigma - \gamma_{S_1}}{\mu}\,. \tag{3.18}$$

Now, substitute (3.14) and (3.17b), then rearrange terms to obtain the stress-strain relationship for the SL model:

$$\frac{\partial \epsilon}{\partial t} = (1 + E_1)^{-1}\left[\frac{\partial \sigma}{\partial t} + E_2\left(\frac{\sigma - E_1\epsilon}{\mu}\right)\right]\,. \tag{3.19}$$

Standard Solid Model

The Standard Solid (SS) Model uses a KV element and a spring connected in series to model the behavior of a viscoelastic material, as shown in Figure 3.4.

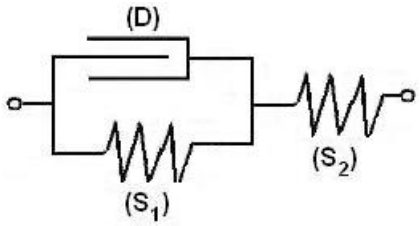

Figure 3.4: A diagram of a Standard Solid element

Since this is equivalent to a Maxwell model where the dashpot is a KV element we know that γ_{KV}, η_{KV} are the total stress and strain of the KV element represented by

$$\begin{aligned} \gamma_{KV} &= \gamma_D + \gamma_{S_1} \\ \eta_{KV} &= \eta_D = \eta_{S_1} \end{aligned} \tag{3.20}$$

and the relationship between them is

$$\gamma_{KV} = E_1\eta_{KV} + \mu\frac{\partial \eta_{KV}}{\partial t}\,. \tag{3.21}$$

The spring is represented by Equation (3.8):

$$\gamma_{S_2} = E_2\eta_{S_2} \tag{3.22}$$

Under an applied axial stress, the total stress, σ and the total strain, ϵ can be defined for the SLS model as follows:

$$\begin{aligned} \sigma &= \gamma_{KV} = \gamma_{S_2} \\ \epsilon &= \eta_{KV} + \eta_{S_2} \end{aligned} \tag{3.23}$$

Differentiate (3.23a) with respect to t. Substitute (3.21). Now use (3.23b) to obtain:

$$\sigma = E_1(\epsilon - \eta_{S_2}) + \mu\frac{\partial}{\partial t}(\epsilon - \eta_{S_2}) \tag{3.24}$$

.

Now, substitute (3.22) and (3.23a), then rearrange terms to obtain the stress-strain relationship for the SS model:

$$\left(1 + \frac{E_1}{E_2}\right)\sigma + \frac{\mu}{E_2}\frac{\partial \sigma}{\partial t} = E_1\epsilon + \mu\frac{\partial \epsilon}{\partial t}\,. \tag{3.25}$$

Chapter 4

1-D Viscoelastic Model

First, we must mathematically define our model structure. We consider a one-dimensional viscoelastic cell with *length*, denoted $\ell(t)$, in contact with a substrate. We define a spatial coordinate system relative to the stationary substrate, and denote the position of the cell *front* and *rear* by $f(t)$ and $r(t)$, respectively. Therefore, at any time t the cell can be represented by the interval $r(t) \leq x'(x,t) \leq f(t)$, where x' is a material point inside the cell at time $t \geq 0$. The displacement u of the cell from its initial position is then defined as

$$u(x,t) = x - x'(x,t) \tag{4.1}$$

We use a Maxwell element to describe the viscoelastic properties of our cell. This gives us the stress-strain relationship from Equation (3.11):

$$\frac{\partial \epsilon}{\partial t} = \frac{1}{E}\frac{\partial \gamma}{\partial t} + \frac{\gamma}{\mu}$$

where ϵ is the total strain across the cell, γ is the internal viscoelastic stress created by Maxwell elements, E is the elastic modulus, and μ is the viscosity.

4.1 Derivation of Model Equation

Since our cell is very small and moves slowly (microns/sec), we can make a justifiable assumption about the total stress. In general for small organisms moving slowly through fluids, the effect of inertial forces are negligible compared to that of the viscous drag forces. In other words, we can assume a low Reynolds number (Berg, 1984). Because of this the material velocity is independent of time, and Equation (3.7) becomes:

$$0 = \frac{\partial \sigma}{\partial x} + X \tag{4.2}$$

Also, we assume that all external forces (e.g. gravity) acting on the cell are negligible, except for viscous drag, due to the small size of the cell. We then have that

$$X = -\beta \frac{\partial u}{\partial t} \tag{4.3}$$

where β is the coefficient of viscous drag. Therefore, Equations (4.2) and (4.3) give us the following relationship between the stress and the velocity of the cell:

$$\frac{\partial \sigma}{\partial x} = \beta \frac{\partial u}{\partial t} \tag{4.4}$$

Our model for the internal stress of a moving fibroblast takes the form of a Moving Boundary Problem obtained from our model assumptions and from the Conservation of Momentum. We define the total stress, $\hat{\sigma}(x,t)$, as a sum of viscoelastic and contractile stresses.

$$\hat{\sigma}(x,t) = \gamma(x,t) + \tau \tag{4.5}$$

We can substitute Equation (3.2) into the LHS of Equation (3.11), to obtain:

$$\frac{\partial \epsilon}{\partial t} = \frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right) \tag{4.6}$$

then substitute (4.4) to get a new LHS of Equation (3.11):

$$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right) = \frac{\partial}{\partial x}\left(\frac{1}{\beta}\frac{\partial \hat{\sigma}}{\partial x}\right) \tag{4.7}$$

Substituting (4.5) for γ in (3.11), we obtain the following Moving Boundary Problem to represent the stress $\hat{\sigma}(x,t)$ in all time and throughout the cellular body.

$$\frac{\partial}{\partial x}\left(\frac{1}{\beta}\frac{\partial \hat{\sigma}}{\partial x}\right) = \frac{1}{E}\frac{\partial \hat{\sigma}}{\partial t} + \frac{\hat{\sigma}}{\mu} - \frac{\tau}{\mu} \tag{4.8}$$

In this model $\hat{\sigma}(x,t)$ is the total internal stress at a time t. Notice that we assume τ to be a function that is independent of time, and we have E as a non-constant function dependent on x.

Boundary Conditions

In order to solve this Moving Boundary Problem, we must also impose boundary conditions.

As the rear and the front of the cell advance at different rates, we define the following relations:

$$\begin{aligned} \frac{dr}{dt} &= V_d + \frac{\partial u}{\partial t}(r(t),t) \\ \frac{df}{dt} &= V_p + \frac{\partial u}{\partial t}(f(t),t) \end{aligned} \tag{4.9}$$

where V_d is the contraction rate of the rear and V_p is the protrusion rate of the front.

We also assume the biological boundary conditions, $\hat{\sigma}(r,t) = \hat{\sigma}(f,t) = 0$.

4.2 Change of Coordinates

In order to solve our Moving Boundary Problem, we must map the moving domain $\{(x,t) : r(t) \leq x \leq f(t), t \geq 0\}$ onto a fixed one by a change of variable. Figure 4.1 is a graphical representation of our change of coordinates.

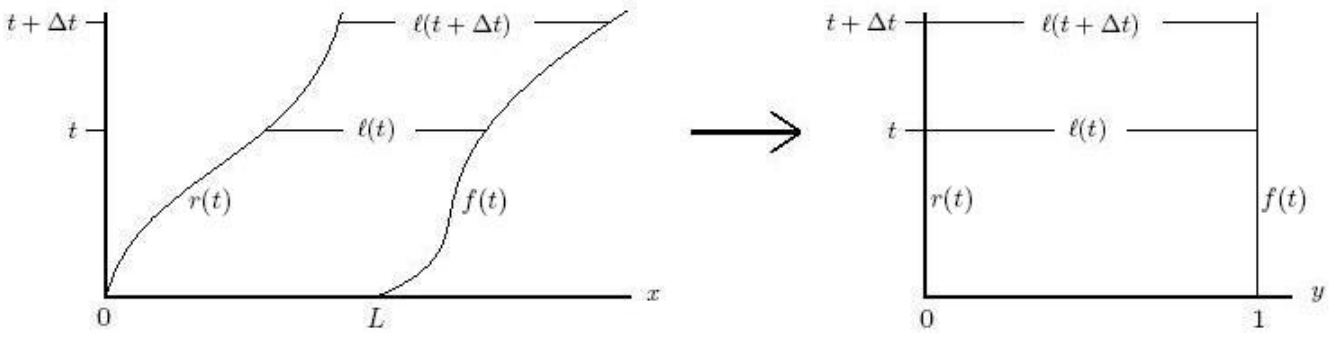

Figure 4.1: Changing from a moving domain to a fixed domain.

Let $y(x,t) = \dfrac{x - r(t)}{\ell(t)}$ where $\ell(t) = f(t) - r(t)$. Define $\sigma(y,t) = \hat{\sigma}(x,t)$. The chain rule then yields the following:

$$\begin{cases} \dfrac{\partial \hat{\sigma}}{\partial x} &= \dfrac{1}{\ell(t)} \dfrac{\partial \sigma}{\partial y}, \\ \\ \dfrac{\partial \hat{\sigma}}{\partial t} &= \dfrac{\partial y}{\partial t} \dfrac{\partial \sigma}{\partial y} + \dfrac{\partial \sigma}{\partial t} \end{cases} \tag{4.10}$$

To simplify notation, let $\bar{\beta} = \dfrac{1}{\beta}$. Substituting in (4.10), Equation (4.8) becomes

$$\frac{\partial \sigma}{\partial t} + \frac{\partial y}{\partial t}\frac{\partial \sigma}{\partial y} = E\left[\frac{1}{\ell(t)}\frac{\partial \bar{\beta}}{\partial y}\frac{\partial \sigma}{\partial y} + \frac{\bar{\beta}}{\ell^2(t)}\frac{\partial^2 \sigma}{\partial y^2} - \frac{\sigma}{\mu} + \frac{\tau}{\mu}\right]. \tag{4.11}$$

The boundaries (Equations (4.9)) are mapped in the same manner, and using (4.4), we have

$$\begin{aligned} \frac{dr}{dt} &= \left(V_d + \frac{\bar{\beta}}{\ell(t)}\frac{\partial \sigma}{\partial y}\right)\Bigg|_{y=0} \\ \frac{df}{dt} &= \left(V_p + \frac{\bar{\beta}}{\ell(t)}\frac{\partial \sigma}{\partial y}\right)\Bigg|_{y=1} \end{aligned} \tag{4.12}$$

Chapter 5

Model One

As found in the previous chapter, we have a way of describing the stress in our cell mapped onto a fixed domain by the Moving Boundary Problem with boundary conditions:

$$\begin{cases} \dfrac{\partial \sigma}{\partial t} + \dfrac{\partial y}{\partial t}\dfrac{\partial \sigma}{\partial y} &= E\left[\dfrac{1}{\ell(t)}\dfrac{\partial \bar{\beta}}{\partial y}\dfrac{\partial \sigma}{\partial y} + \dfrac{\bar{\beta}}{\ell^2(t)}\dfrac{\partial^2 \sigma}{\partial y^2} - \dfrac{\sigma}{\mu} + \dfrac{\tau}{\mu}\right] \\ \dfrac{dr}{dt} &= \left(V_d + \dfrac{\bar{\beta}}{\ell(t)}\dfrac{\partial \sigma}{\partial y}\right)\Big|_{y=0} \\ \dfrac{df}{dt} &= \left(V_p + \dfrac{\bar{\beta}}{\ell(t)}\dfrac{\partial \sigma}{\partial y}\right)\Big|_{y=1} \\ \sigma(0,t) = 0 \quad , \quad \sigma(1,t) &= 0 \end{cases} \tag{5.1}$$

We have two different numerical methods to solve (5.1), one explicit and one implicit. The comparison of results between these two methods allows us to more easily detect approximation error. First we must make some assumptions of our functions. We must note that many of the assumptions we make are taken from [6] and [7].

5.1 Assumptions for Material Properties

Actin
We assume that the concentration of actin across the cell is a linear function defined as

$$a(y) = y \tag{5.2}$$

Elastic Modulus
We assume the elastic modulus to be defined as

$$E(y) = E_0 a(y) \tag{5.3}$$

Figure (5.1) is a graphical description of our assumption for E.

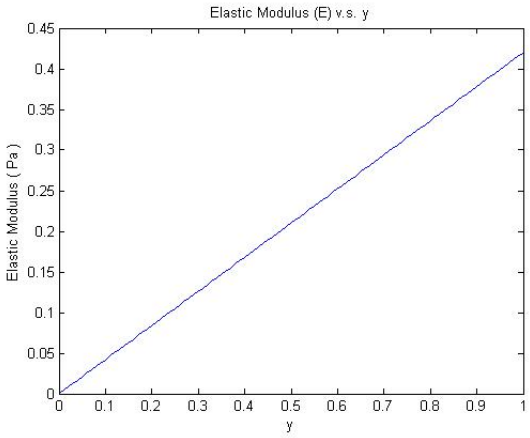

Figure 5.1: Elastic Modulus (E) for Model One

Viscous Drag Coefficient

We assume that the cell's viscous drag coefficient, β, is proportional to the density of the integrin receptors bound to the substrate, n_b.

$$\beta(y) = \beta_0 n_b(y) \tag{5.4}$$

Figure (5.2) is a graphical description of our assumption for β.

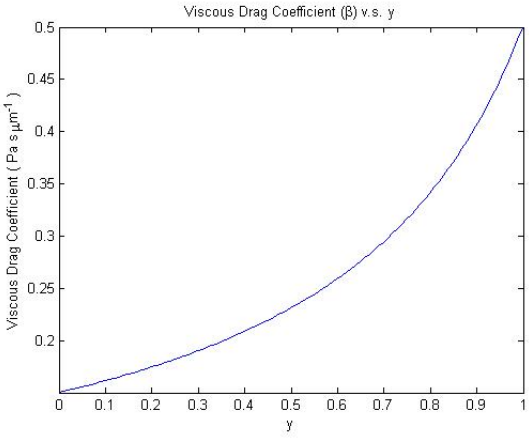

Figure 5.2: Viscous Drag Coefficient (β) for Model One

We assume that the density of integrin receptors bound to the substrate is proportional to the concentration of substrate ligands, n_s, and free (unbound) integrin receptors, n_f.

We keep the association rate, k_f, as well as the dissociation rate, k_{r0}, constant throughout the cell. Since the integrin-substrate bonds are chemically weakened in the rear of the cell we define a unitless position function

$$f_1(y) = \psi_1 + (1 - \psi_1)y \tag{5.5}$$

which linearly increases toward the rear. To simplify our notation, we denote $\kappa_s = \dfrac{k_f n_s}{k_{r0}}$. So we obtain a function for the bound integrin with respect to cell position:

$$n_b(y) = \frac{\kappa_s}{f_1(y)} n_f(y) \tag{5.6}$$

where $n_f(y) + n_b(y) = N_{Total}$, and N_{Total} is a constant total number of integrin in the cell.

Contractile Stress

We define the cell's contractile stress, τ to be proportional to the concentration of bound active Myosin II.

$$\tau(y) = \tau_0 m_b^+(y) \tag{5.7}$$

Figure (5.3) is a graphical description of our assumption for τ.

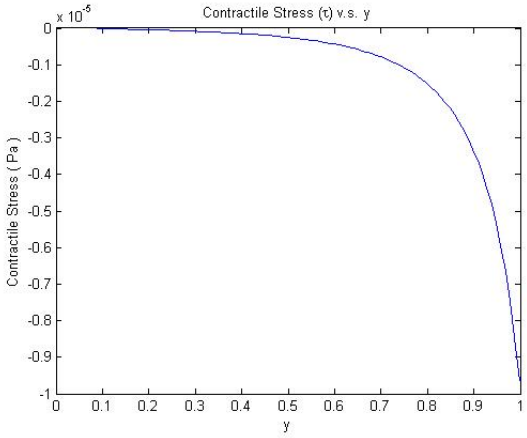

Figure 5.3: Contractile Stress (τ) for Model One

In this equation m_b^+ is the density of active Myosin II bound to actin, and τ_0 is the mean magnitude of stress produced by each motor. The equation for active bound myosin is

$$m_b^+(y) = \frac{k_{Reg}^+}{f_2(y) k_{Reg}^-} [Reg] m_b(y) \tag{5.8}$$

where $f_2(y)$, similarly to Equation (5.5), is defined below as

$$f_2(y) = \psi_2 + (1 - \psi_2)y. \tag{5.9}$$

The density of bound myosin is the ratio of myosin association (k_m^+) and dissociation (k_m^-) rates multiplied by the concentration of free myosin ($m_f(y)$) and actin,

$$m_b(y) = \frac{k_m^+}{k_m^-} m_f(y) a(y) \tag{5.10}$$

where $m_f(y) + m_b(y) = M_{Total}$, and M_{Total} is a constant total number of myosin in the cell.

5.2 Numerical Methods

We cannot solve this Moving Boundary Problem analytically, but in this section, we develop methods to solve it numerically.

Explicit Method
To obtain an explicit equation for $\sigma(y, t + \Delta t)$ we must discretize Equation (5.1) with respect to the current time t and then solve for $\sigma(y, t + \Delta t)$ algebraically. To help ease notation, we define $t0 = t$ and $t1 = t + \Delta t$. We use a central difference method to approximate derivatives.

Isolating $\dfrac{\partial \sigma}{\partial t}$, Equation (5.1a) discretizes into

$$\begin{aligned}
\frac{\sigma(y_i, t1) - \sigma(y_i, t0)}{\Delta t} &= E(y_i, t0)\left[\frac{1}{\ell(t0)}\frac{\partial \bar{\beta}(y_i, t0)}{\partial y}\frac{\sigma(y_{i+1}, t0) - \sigma(y_{i-1}, t0)}{2\Delta y}\right. \\
&+ \frac{\bar{\beta}(y_i, t0)}{\ell^2(t0)}\frac{\sigma(y_{i+1}, t0) + \sigma(y_{i-1}, t0) - 2\sigma(y_i, t0)}{\Delta y^2} \\
&- \left.\frac{\sigma(y_i, t0)}{\mu} + \frac{\tau(y_i, t0)}{\mu}\right] - \frac{\partial y}{\partial t}\frac{\sigma(y_{i+1}, t0) - \sigma(y_{i-1}, t0)}{2\Delta y} \\
&\equiv F(y_{i+1}, y_i, y_{i-1}, t0)\,.
\end{aligned} \tag{5.11}$$

We then solve for $\sigma(y_i, t1)$ to obtain the explicit equation:

$$\sigma(y_i, t1) = \Delta t\, F(y_{i+1}, y_i, y_{i-1}) + \sigma(y_i, t0)\,. \tag{5.12}$$

We use an explicit method to find the front and rear positions at the next timestep. First, we discretize Equations (5.1b) and (5.1c), and then solve for the desired positions.

We discretize to obtain an approximation for the rear at a time $t1$.

$$r(t1) = r(t0) + \Delta t\left(V_d + \frac{\bar{\beta}(0)}{\ell(t0)}\frac{\sigma(\Delta y, t0)}{\Delta y}\right) \tag{5.13}$$

Similarly, we do this for the front of the cell, but we know the cell increases in length, due to actin protrusions pushing out the leading edge, until the cell's motion reaches a steady state, so we let $V_p = \dfrac{V_0 L_0}{\ell(t0)}$.

$$f(t1) = f(t0) + \Delta t\left(\frac{V_0 L_0}{\ell(t0)} - \frac{\bar{\beta}(1)}{\ell(t0)}\frac{\sigma(1 - \Delta y, t0)}{\Delta y}\right) \tag{5.14}$$

Note that a sign in $f(t1)$ has changed, due to the fact that $\sigma(f(t0)) = 0$.

Implicit Method

Let us now take Equation (5.1a) and discretize it with respect to a future timestep $t1$. This will be similar to Equation (5.11), however the σ times $t0$ and $t1$ will be interchanged with one another. For the sake of notation, we rewrite Equation (5.11) in terms of coefficients of σ.

$$A_{i,i-1}\sigma(y_{i-1}, t1) + A_{i,i}\sigma(y_i, t1) + A_{i,i+1}\sigma(y_{i+1}, t1) = b \tag{5.15}$$

where the coefficients are:

$$\begin{aligned} A_{i,i-1} &= \frac{-1}{2E\Delta y}\frac{\partial y}{\partial t} + \frac{1}{2\ell(t0)\Delta y}\frac{\partial \bar{\beta}}{\partial y} - \frac{\bar{\beta}}{\ell(t0)^2\Delta y^2} \\ A_{i,i} &= \frac{1}{\mu} + \frac{2\bar{\beta}}{\ell(t0)^2\Delta y^2} + \frac{1}{E\Delta t0} \\ A_{i,i+1} &= \frac{1}{2E\Delta y}\frac{\partial y}{\partial t} - \frac{1}{2\ell(t0)\Delta y}\frac{\partial \bar{\beta}}{\partial y} - \frac{\bar{\beta}}{\ell(t0)^2\Delta y^2} \\ b_i &= -\frac{\sigma(y_i, t0)}{E\Delta t} + \frac{\tau}{\mu} \end{aligned} \tag{5.16}$$

Note that in all coefficients, the functions expressed in them are at $(y_i, t0)$ regardless of what y-step the corresponding σ value is. We then transform these equations into matrix notation so that:

$$\begin{pmatrix} \vdots \\ b_i \\ \vdots \end{pmatrix} = \begin{pmatrix} A_{1,1} & A_{1,2} & 0 & \dots & 0 \\ A_{2,1} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & A_{n-1,n} \\ 0 & \dots & 0 & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} \vdots \\ \sigma(y_i, t1) \\ \vdots \end{pmatrix}$$

where $A_{i,i-1}, A_{i,i}, A_{i,i+1}, b_i$ represent the coefficients of $\sigma(y_i, t1)$. We then solve this matrix using Gaussian Elimination to obtain $\sigma(y, t1)$.

The boundaries for the implicit method are calculated the same way as the explicit method. See Equations (5.13) and (5.14).

Numerical Results

We divided the cell interior into 100 discrete equidistant points. Taking the front and rear

into consideration, we obtain 101 sections each having length $\Delta y = \frac{1}{101}$. To ensure numerical convergence to the PDE solution we define a timestep $\Delta t = \frac{1}{2}\Delta y^2$ to be small enough to avoid compounding approximation error.

We ran the explicit method to the computational limit available to us and were unable to reach a steady state. This reason, and the lack of accuracy in the explicit method, drove us to not use this method in Model Two's development.

We ran the implicit method with several different initial stress values until there was a relative σ difference between timesteps smaller than 10^{-10}. Our results for Model One's Steady state solution found from a parabolic σ_0, found implicitly are displayed in Figure 5.4.

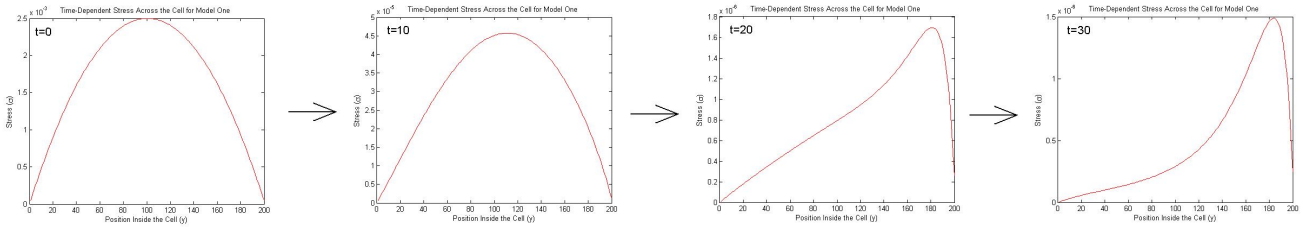

Figure 5.4: Stress (σ) vs. Position (y) for t = 0, 10, 20, 30 time-steps.

Note, t = 30 is the steady-state solution.

Model One's Steady state solution found from a sinusoidal σ_0, found implicitly is shown in Figure (5.5) to make sure our model converged to the same solution with very different initial conditions.

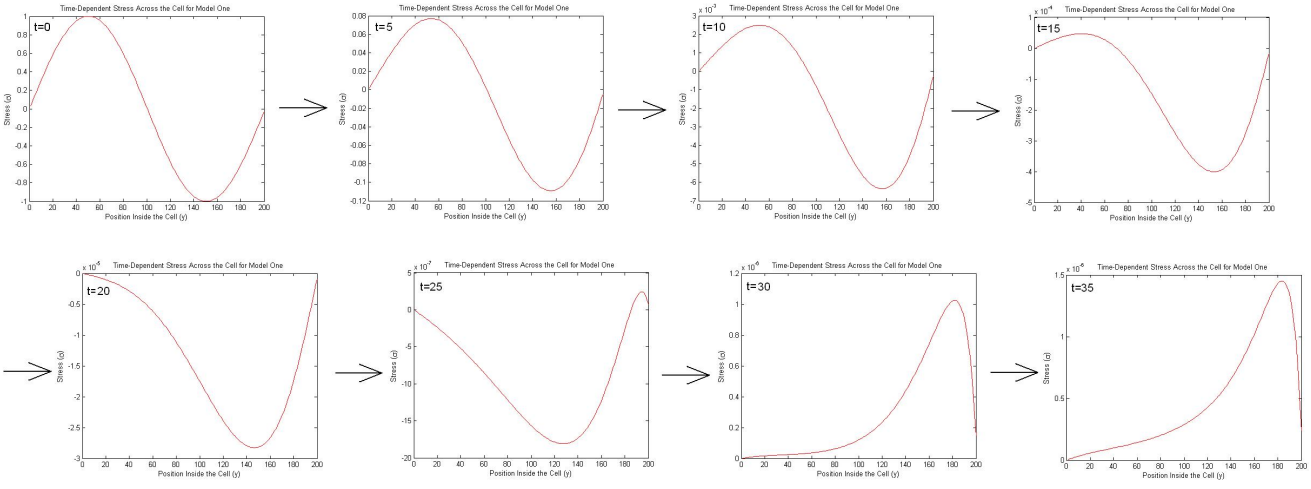

Figure 5.5: Stress (σ) vs. Position (y) for t = 0 to 35 time-steps shown every 5 steps.

Note, t = 35 is the steady-state solution. After trying multiple possible initial conditions, we

concluded our steady-state to be accurate. Varying the Δt and initial condition slightly made little difference for either accuracy or speed of convergence.

Here is a graph of the cell's displacement ($u(x,t)$) versus time (t). Notice the cell reaches a steady-state in reasonable time, and from this we can find the cell's steady-state velocity and length.

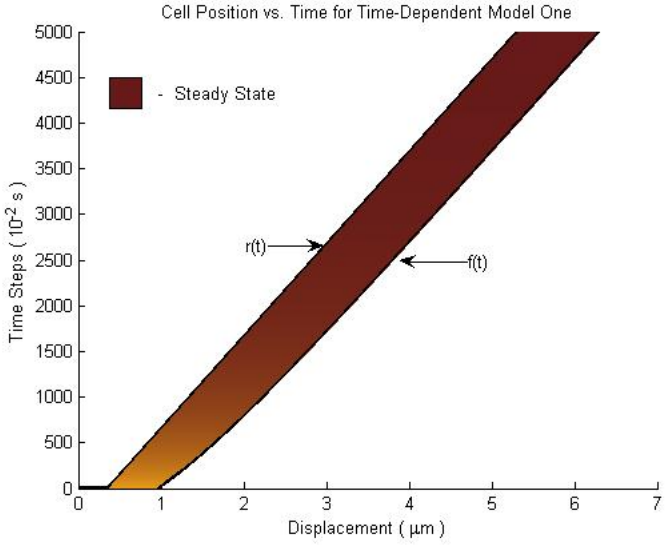

Figure 5.6: Cell Displacement over time.

5.3 Traveling Waves

Our numerical results led us to believe that our MBP could have traveling wave solutions. In order to verify this, we solve a traveling wave problem assuming cell length and speed, denoted L and k, to be constant.

Theoretically, this represents the cell when it has reached a correct steady state solution, because by definition these two values should remain constant. After our traveling wave problem is solved we can then compare the k, L, and σ patterns we get with the values we obtained from our previous time-dependent model.

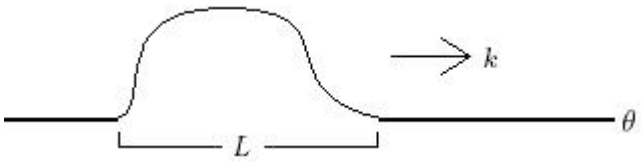

Figure 5.7: A graphical depiction of a traveling wave with constant length (L) and velocity (k).

Due to these assumptions, the paths of the front and rear of the cell become parallel to each other in time, and the stress will remain constant for all time since it is a steady state equation. This makes x and t no longer independent and we can change the coordinate system to one variable, θ, simplifying our model down from a partial to an ordinary differential equation.

Note that in order to have a traveling wave solution we must assume E, β, τ to be constant functions. Let $\theta = x - kt$ and $\sigma_w(\theta) = \sigma(x - kt)$. We also assume that $r(t) = kt$ and $f(t) = kt + L$. Thus, r, L, and σ_w are unknowns in the problem. By the chain rule we then obtain

$$\begin{cases} \dfrac{\partial \sigma}{\partial x} &= \dfrac{d\sigma_w}{d\theta} = \sigma_w'(\theta) \\ \dfrac{\partial \sigma}{\partial t} &= -k\dfrac{d\sigma_w}{d\theta} = -k\sigma_w'(\theta) \end{cases} \tag{5.17}$$

We can now obtain a second order, linear ODE with constant coefficients. By substituting (5.17) into Equation (5.1) we obtain

$$\begin{cases} \sigma_w'' &= \beta\left(\dfrac{-k}{E}\sigma_w' + \dfrac{\sigma_w - \tau}{\mu}\right) \\ k &= V_d + \bar{\beta}\sigma_w'(0) \\ k &= \dfrac{V_0 L_0}{L} + \bar{\beta}\sigma_w'(L) \\ \sigma_w(0) = 0 \quad , & \quad \sigma_w(L) = 0 \end{cases} \tag{5.18}$$

Scaling
We can now scale Equation (5.1) by defining the following:

$$\begin{aligned} &\bar{\theta} = \beta\theta, \quad \bar{\sigma} = \frac{\sigma}{E}, \quad \bar{\tau} = \frac{\tau}{E}, \quad \bar{k} = \frac{k}{E}, \quad \bar{L} = \beta L \\ &\bar{\mu} = \beta\mu, \quad \bar{L_0} = \frac{V_0 \beta L_0}{E}, \quad \bar{V_d} = \frac{V_d}{E} \end{aligned} \tag{5.19}$$

This makes Equation (5.18) become

$$\begin{cases} \bar{\sigma}_w'' &= -\bar{k}\bar{\sigma}_w' + \dfrac{\bar{\sigma}_w - \bar{\tau}}{\bar{\mu}} \\ \bar{k} &= \bar{V_d} + \bar{\sigma}_w'(0) \\ \bar{k} &= \dfrac{\bar{L_0}}{\bar{L}} + \bar{\sigma}_w'(\bar{L}) \\ \bar{\sigma}_w(0) = 0 \quad , & \quad \bar{\sigma}_w(L) = 0 \end{cases} \tag{5.20}$$

In order to solve for $\bar{k}$ and $\bar{L}$ we can use the fact that now $\frac{dr}{dt} = \frac{df}{dt} = k$. We can then use the boundary equations (5.1b,c) from our time-dependent model to generate explicit equations for $\bar{k}$ and $\bar{L}$. The equation for the rear boundary gives us

$$\bar{k} = \bar{V}_d + \bar{\sigma}'_w(0) \tag{5.21}$$

where all variables are now known except $\bar{k}$, so we have an explicit equation for the cell's speed. Next, we look at the equation for the front of the cell, which yields

$$\bar{k} = \frac{\bar{L}_0}{\bar{L}} + \bar{\sigma}'_w(\bar{L})\,. \tag{5.22}$$

Substituting this into equation (5.21), we can solve for $\bar{L}$ to obtain an explicit equation for the cell's length.

$$\bar{L} = \bar{L}_0 \left[\left(\bar{V}_d + \bar{\sigma}'_w(0)\right) - \bar{\sigma}'_w(\bar{L})\right]^{-1} \tag{5.23}$$

Numerical Method

We can now solve this traveling wave problem using the same implicit method used for our time-dependent model. We approximate the first and second derivatives in the same way and we use the same process as before to obtain an equation in implicit form. Our model can thus be approximated by the following equation:

$$A_{i,i-1}\bar{\sigma}_w(\bar{\theta} - \Delta\theta) + A_{i,i}\bar{\sigma}_w(\bar{\theta}) + A_{i,i+1}\bar{\sigma}_w(\bar{\theta} + \Delta\theta) = b_i \tag{5.24}$$

where

$$\begin{aligned} A_{i,i-1} &= \frac{1}{\Delta\theta^2} - \frac{\bar{k}}{2a\Delta\theta} \\ A_{i,i} &= -\frac{2}{\Delta\theta^2} - \frac{1}{\bar{\mu}} \\ A_{i,i+1} &= \frac{1}{\Delta\theta^2} + \frac{\bar{k}}{2a\Delta\theta} \\ b_i &= -\frac{\bar{\tau}}{\bar{\mu}} \end{aligned} \tag{5.25}$$

We can then form a tri-diagonal matrix M from the coefficients A and b. We then use Gaussian Elimination to solve for $\bar{\sigma}_w(\theta_i)$ using the equation $M\bar{\sigma}_w = D$. Then we can obtain values for $\bar{k}$ and $\bar{L}$, and finally rescale them back to k and L.

Numerical Results

Our method reached a solution within 10 iterations. Figure 5.7 shows the graph of σ vs. y. We can obtain k, L values from this solution, and unscale them to compare them with the time-dependent steady-state velocity and length. These are the unscaled values:

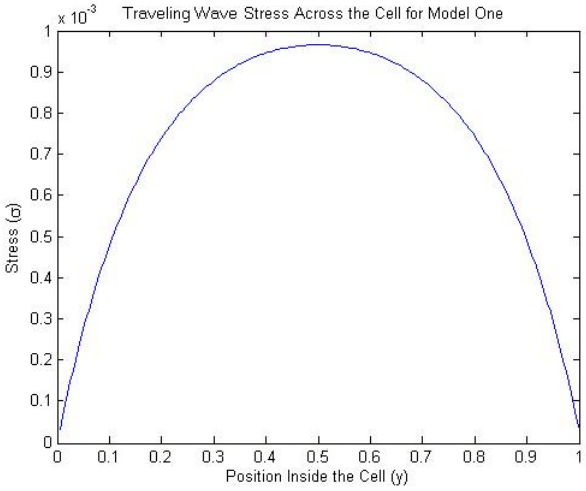

Figure 5.8: Model One Traveling Wave Stress (σ) vs. Position (y)

$$\begin{aligned} k &= 0.146818\ \mu m \cdot s^{-1} \\ L &= 0.518326\ \mu m \end{aligned} \tag{5.26}$$

This matches the length, velocity and stress of the time-dependent model's steady-state when setting the functional coefficients to be constants (see Table 6.2).

5.4 The Existence and Uniqueness of a Traveling Wave

Theoretically, there exists a solution to this Traveling Wave, and the solution is unique. A proof of the existence and uniqueness of this Traveling Wave solution is now presented, developed by us and our advisor, Professor Roger Lui.

We begin with the scaled moving boundary problem (MBP)

$$\begin{cases} \dfrac{d^2\sigma}{d\theta^2} + k\dfrac{d\sigma}{d\theta} - \dfrac{\sigma}{\mu} = -\dfrac{\tau}{\mu} \\ \sigma(0) = 0, \sigma(L) = 0 \\ k = V_d + \dfrac{d\sigma}{d\theta}\Big|_0 \quad , \quad k = \dfrac{L_0}{L} + \dfrac{d\sigma}{d\theta}\Big|_L \end{cases} \tag{5.27}$$

5.4.1 General Solution of MBP

We can now get a general solution to this ODE easily by summing a particular solution and the homogenous solution. We obtain the following general solution to the ODE:

$$\sigma(\theta) = \bar{C}_1 e^{\alpha_1 \theta} + \bar{C}_2 e^{\alpha_2 \theta} + \tau \tag{5.28}$$

where α_1, α_2 are roots of the equation $\xi^2 + k\,\xi - \frac{1}{\mu} = 0$, i.e.

$$\begin{aligned} \alpha_1 &= -\frac{k}{2} + \frac{1}{2}\sqrt{k^2 + \frac{4}{\mu}} \\ \alpha_2 &= -\frac{k}{2} - \frac{1}{2}\sqrt{k^2 + \frac{4}{\mu}}\,. \end{aligned}$$

Now we use the facts that

$$\begin{aligned} \sigma(0) &= \bar{C}_1 + \bar{C}_2 + \tau = 0 \\ \sigma'(\theta) &= \bar{C}_1\alpha_1 e^{\alpha_1\theta} + \bar{C}_2\alpha_2 e^{\alpha_2\theta} \\ \sigma'(0) &= \bar{C}_1\alpha_1 + \bar{C}_2\alpha_2 = k - V_d \end{aligned}$$

to obtain $\begin{pmatrix} 1 & 1 \\ \alpha_1 & \alpha_2 \end{pmatrix} \begin{pmatrix} \bar{C}_1 \\ \bar{C}_2 \end{pmatrix} = \begin{pmatrix} -\tau \\ k - V_d \end{pmatrix}$ which implies that

$$\begin{aligned} \bar{C}_1 &= \frac{\alpha_2\tau + k - V_d}{\alpha_1 - \alpha_2} \\ \bar{C}_2 &= \frac{-\alpha_1\tau - k + V_d}{\alpha_1 - \alpha_2}\,. \end{aligned}$$

Note that $\alpha_1 > 0 > \alpha_2$ for all $k \geq 0$.

5.4.2 Lemmas

Lemma 1

σ does not have a negative minimum on $0 \leq \theta \leq L$.

Proof:
Apply Maximum Principle. Observe that if σ has a negative minimum, then $\frac{d^2\sigma}{d\theta^2} > 0$, $k\frac{d\sigma}{d\theta} = 0$, and $\frac{\sigma}{\mu} > 0$ so their sum must be positive. However, $-\frac{\tau}{\mu} < 0$. This gives a contradiction, and we are done. □

Lemma 2

$$\sigma > 0 \text{ on } (0, L), \;\; \sigma'(0) > 0 \;\; \text{and} \;\; \sigma'(L) < 0.$$

Proof:
This follows from Lemma 1.

Corollary

For Traveling Wave solutions, $k > V_d$.

Proof:
By lemma 2, we have $k - V_d = \sigma'(0) > 0$. Therefore, $k > V_d$. □

Lemma 3

$$\bar{C}_1 < 0.$$

Proof:
If not, $\sigma(\theta) \to \infty$ as $\theta \to \infty$. Since $\sigma(L) = 0$, then it forces a negative minimum. This contradicts Lemma 1 and we are done. □

From the definitions of α_1 and α_2 above, we have

$$\begin{aligned}
\frac{d\alpha_1}{dk} &= -\frac{1}{2} + \frac{1}{2}k\left(k^2 + \frac{4}{\mu}\right)^{-1/2} = \frac{-\alpha_1}{\alpha_1 - \alpha_2} < 0 \\
\frac{d\alpha_2}{dk} &= -\frac{1}{2} - \frac{1}{2}k\left(k^2 + \frac{4}{\mu}\right)^{-1/2} = \frac{\alpha_2}{\alpha_1 - \alpha_2} < 0\,.
\end{aligned}$$

Now using the facts that $\sigma(L) = 0$ and $\sigma'(L) = k - L_0/L$, we have

$$\begin{pmatrix} 1 & 1 \\ \alpha_1 & \alpha_2 \end{pmatrix} \begin{pmatrix} \bar{C}_1 e^{\alpha_1 L} \\ \bar{C}_2 e^{\alpha_2 L} \end{pmatrix} = \begin{pmatrix} -\tau \\ k - \frac{L_0}{L} \end{pmatrix} \Rightarrow \begin{pmatrix} \bar{C}_1 e^{\alpha_1 L} \\ \bar{C}_2 e^{\alpha_2 L} \end{pmatrix} = \frac{1}{\alpha_1 - \alpha_2} \begin{pmatrix} \alpha_2\tau + k - \frac{L_0}{L} \\ -\alpha_1\tau - k + \frac{L_0}{L} \end{pmatrix}$$

Rearranging, we have

$$\begin{cases}
e^{\alpha_1 L} + \dfrac{L_0}{\bar{C}_1(\alpha_1 - \alpha_2)L} = \dfrac{\alpha_2\tau + k}{\bar{C}_1(\alpha_1 - \alpha_2)} \quad \text{——} \quad (1) \\
e^{\alpha_2 L} - \dfrac{L_0}{\bar{C}_2(\alpha_1 - \alpha_2)L} = \dfrac{-\alpha_1\tau - k}{\bar{C}_2(\alpha_1 - \alpha_2)} \quad \text{——} \quad (2)
\end{cases}$$

which we label (1) and (2) respectively.

Lemma 4

Given $k > V_d$, there exists a unique solution to (1) and (2), called $L_1(k)$, $L_2(k)$ respectively.

Proof:
Since $\bar{C}_1 < 0$, $\bar{C}_2 < 0$ and $\alpha_1 > 0 > \alpha_2$, the left hand side of (1) increases from $-\infty$ to ∞ as L increases on $(0, L)$. Similarly, the left hand side of (2) decreases from ∞ to 0 on $(0, \infty)$. Note that the right hand side of (2) is positive. The proof of the lemma is complete. □

Lemma 5

$$\text{Let } \tau \leq 1, \text{ then } \frac{dL_1}{dk} > 0 \text{ and } \frac{dL_2}{dk} < 0.$$

Proof:
Let $\xi = \bar{C}_1(\alpha_1 - \alpha_2) = \alpha_2\tau + (k - V_d) < 0$. Then Equation (1) may be rewritten as

$$\xi e^{\alpha_1 L} + \frac{L_0}{L} = \alpha_2\tau + k = \xi + V_d.$$

Differentiating with respect to k, we have

$$\xi' e^{\alpha_1 L} + \xi(\alpha_1 L)' e^{\alpha_1 L} - \frac{L_0}{L^2} L' = \xi'.$$

Rearranging, we have

$$\xi' + \xi(\alpha_1' L + \alpha_1 L') - \frac{L_0}{L^2} L' = \xi' e^{-\alpha_1 L}.$$

Thus,

$$L' = \frac{\xi'(e^{-\alpha_1 L} - 1) - \xi\alpha_1' L}{\alpha_1 \xi - \frac{L_0}{L^2} e^{-\alpha_1 L}}.$$

Since $\bar{C}_1 < 0$, we have $\xi < 0$ and from above, $\alpha_1' < 0$. Now

$$\xi' = \alpha_2'\tau + 1 = \frac{\alpha_2\tau}{\alpha_1 - \alpha_2} + 1 = \frac{(\tau - 1)\alpha_2 + \alpha_1}{\alpha_1 - \alpha_2} > 0.$$

Therefore, L' is the ratio of two negative quantities and is positive.

To show that $L_2'(k) < 0$, let $\eta = \bar{C}_2(\alpha_1 - \alpha_2) = -\alpha_1\tau - k + V_d$. Then Equation (2) may be rewritten as

$$\eta e^{\alpha_2 L} - \frac{L_0}{L} = -\alpha_1\tau - k = \eta - V_d.$$

Differentiating with respect to k, we have

$$\eta' e^{\alpha_2 L} + \eta(\alpha_2 L)' e^{\alpha_2 L} + \frac{L_0}{L^2} L' = \eta'$$

which as before yields

$$L' = \frac{\eta'(e^{-\alpha_2 L} - 1) - \eta\alpha_2' L}{\eta\alpha_2 + \frac{L_0}{L^2} e^{-\alpha_2 L}} .$$

Now

$$\eta' = -\alpha_1'\tau - 1 = \frac{\alpha_1}{\alpha_1 - \alpha_2}\tau - 1 = \frac{\alpha_1\tau - \alpha_1 + \alpha_2}{\alpha_1 - \alpha_2} < 0$$

if $\tau \leq 1$. Also, since $\eta < 0$, $\alpha_2' < 0$ and $\alpha_2 < 0$, the numerator of L' is negative while its denominator is positive. Therefore, $L_2' < 0$. The proof of the lemma is complete□

Lemma 6

$$L_1(V_d) \ < \ L_2(V_d) .$$

Proof:
Set $k = V_d$ in equations (1) and (2). Then $\bar{C}_1(\alpha_1 - \alpha_2) = \alpha_2\tau$, $\alpha_1 + \alpha_2 = -k = -V_d$, so $\alpha_2 = -(\alpha_1 + V_d)$. Then equations (1) and (2) above become

$$\begin{cases} e^{\alpha_1 L} + \dfrac{1}{\alpha_2\tau}(\dfrac{L_0}{L} - V_d) &= 1 \\ \\ e^{\alpha_2 L} + \dfrac{1}{\alpha_1\tau}(\dfrac{L_0}{L} - V_d) &= 1 . \end{cases}$$

Let $\eta = -\dfrac{\alpha_2}{\alpha_1} = \dfrac{\alpha_1 + V_d}{\alpha_1} > 0$, so $\alpha_2 = -\alpha_1\eta$. Therefore,

$$\begin{cases} e^{\alpha_1 L_1} - \dfrac{1}{\alpha_1\eta\tau}(\dfrac{L_0}{L_1} - V_d) &= 1 \\ \\ e^{-\alpha_1\eta L_2} + \dfrac{1}{\alpha_1\tau}(\dfrac{L_0}{L_2} - V_d) &= 1 . \end{cases}$$

Suppose (i) is not true. Then, $L_2(V_d) \leq L_1(V_d)$, and $1 - e^{-\alpha_1\eta L_2} > \eta(e^{\alpha_1 L_1} - 1)$.
If $L_2 \leq L_1$, we have $1 - e^{-\alpha_1\eta L_2} \leq 1 - e^{\alpha_1\eta L_1}$. Hence, we have $\dfrac{1 - e^{-\alpha_1\eta L_1}}{\eta} > e^{\alpha_1 L_1} - 1$.
Consider two functions

$$f(u) \ = \ \frac{1 - e^{-\eta u}}{\eta}$$

$$g(u) \ = \ e^{u} - 1$$

where $\eta > 0$ and $u \geq 0$. Note that $f(0) = g(0) = 0$. Also

$$f'(u) = e^{-\eta u} < 1$$

$$g'(u) = e^{u} > 1$$

Therefore $f'(u) \leq g'(u)$, and $f(u) < g(u)$ for all $u > 0$, in particular when $u = \alpha_1 L_1$ which is a contradiction. The proof of the lemma is complete. □

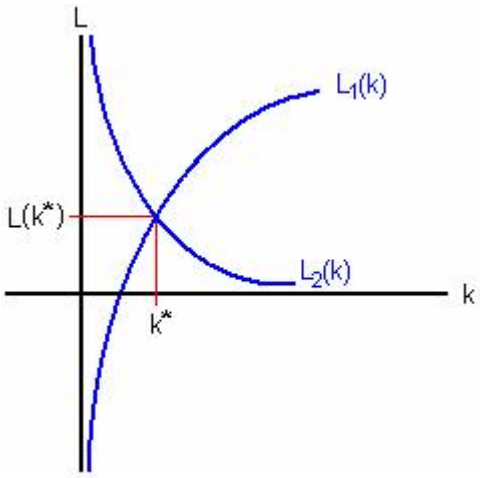

Figure 5.9: Graphical Representation of the unique intersection of L_1 and L_2.

Chapter 6

Model Two

In this model, we improve upon Model One by removing the assumption that actin is a linear function. We replace it by Equation (6.1):

$$\frac{\partial \hat{a}}{\partial t} = -\frac{\partial}{\partial x}(v\hat{a}) - \Gamma\hat{a} \tag{6.1}$$

with boundary condition $a(f(t), t) = a_L$. By definition, $v = \dfrac{\partial u}{\partial t}$. Therefore, we can substitute for v using Equation (4.4), and move all terms to one side to obtain the new equation:

$$\begin{gathered} \frac{\partial \hat{a}}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\hat{a}}{\beta}\frac{\partial \hat{\sigma}}{\partial x}\right) + \Gamma\hat{a} = 0 \\ a(f(t), t) = a_L \end{gathered} \tag{6.2}$$

We change coordinates for (6.2) using (4.10), and defining $a(y, t) = \hat{a}(x, t)$ where

$$\begin{cases} \dfrac{\partial \hat{a}}{\partial x} &= \dfrac{1}{\ell(t)}\dfrac{\partial a}{\partial y} \\ \dfrac{\partial \hat{a}}{\partial t} &= \dfrac{\partial y}{\partial t}\dfrac{\partial a}{\partial y} + \dfrac{\partial a}{\partial t} \end{cases} \tag{6.3}$$

Using the same boundary conditions as for Model One, this gives us Model Two:

$$\begin{cases} \dfrac{\partial \sigma}{\partial t} + \dfrac{\partial y}{\partial t}\dfrac{\partial \sigma}{\partial y} &= E_0 \left[\dfrac{1}{\ell(t)} \dfrac{\partial \bar{\beta}}{\partial y} \dfrac{\partial \sigma}{\partial y} + \dfrac{\bar{\beta}}{\ell^2(t)} \dfrac{\partial^2 \sigma}{\partial y^2} - \dfrac{\sigma}{\mu} + \dfrac{\tau}{\mu} \right] \\ 0 &= \dfrac{\partial a}{\partial t} + \left[\dfrac{\partial y}{\partial t} + \dfrac{\bar{\beta}}{\ell(t)^2} \dfrac{\partial \sigma}{\partial y} \right] \dfrac{\partial a}{\partial y} + \left[\Gamma + \dfrac{\bar{\beta}}{\ell(t)^2} \dfrac{\partial^2 \sigma}{\partial y^2} \right] a \\ \dfrac{dr}{dt} &= \left(V_d + \dfrac{\bar{\beta}}{\ell(t)} \dfrac{\partial \sigma}{\partial y} \right) \Big|_{y=0} \\ \dfrac{df}{dt} &= \left(V_p + \dfrac{\bar{\beta}}{\ell(t)} \dfrac{\partial \sigma}{\partial y} \right) \Big|_{y=1} \\ \sigma(0,t) = 0 \quad , \quad \sigma(1,t) = 0 \quad , \quad a(1,t) = 1 \end{cases} \tag{6.4}$$

Note that we can scale out a_L by first setting $\bar{a} = \dfrac{a}{a_L}$, then setting $\bar{E}_0 = E_0 a_L$. Dropping the bar notation, this will leave us with the same equations for E and a, but with the new boundary condition $a(1,t) = 1$.

Note that while our definition of a has changed, E, β and τ remain as defined in Model One.

6.1 Numerical Method

We are currently developing a numerical method to solve Model Two using the Upwind Method and a similar implicit scheme as the one used for Model One.

6.2 Traveling Wave

We approach this traveling wave the same way we did in Model One (Section (5.3)). Again take $\theta = x - kt$ and $\sigma_w(\theta) = \sigma(x - kt)$, $a_w(\theta) = a(x - kt)$. We let β, τ be constant functions, and we let $E = E_0 a$. Therefore, Equation (6.4) becomes

$$\begin{cases} \sigma_w'' &= \beta \left(\dfrac{\sigma_w - \tau}{\mu} - \dfrac{k\sigma_w'}{E_0 a} \right) \\ 0 &= -k a_w' + \Gamma a_w \\ k &= V_d + \bar{\beta} \sigma_w'(0) \\ k &= \dfrac{V_0 L_0}{L} + \bar{\beta} \sigma_w'(L) \\ \sigma_w(0) = 0 \quad , \quad \sigma_w(L) = 0 \ , \ a_w(L) = 1 \end{cases} \tag{6.5}$$

Now we can scale these equations using Equation (5.19), replacing E with E_0. In addition we must scale our new equation for a, so we define

$$\bar{\Gamma} = \frac{\Gamma}{E_0 \beta}. \tag{6.6}$$

Our equation for a becomes

$$a' - \frac{\bar{\Gamma} + \bar{\sigma}''}{\bar{k} - \bar{\sigma}'} a = 0 \tag{6.7}$$

with boundary condition $a(L) = 1$. Since this equation is a first-order linear ODE we can solve for a explicitly to have

$$a(\theta) = e^{-\int_\theta^L \frac{\bar{\Gamma} + \bar{\sigma}''}{\bar{k} - \bar{\sigma}'} d\theta} = \left[\frac{\bar{L}_0}{\bar{L}(\bar{k} - \bar{\sigma}')} \right] e^{-\bar{\Gamma} \int_\theta^{\bar{L}} \frac{d\theta}{\bar{k} - \bar{\sigma}'}} \tag{6.8}$$

We can now take this and solve the stress-strain equation with a as a known function.

Numerical Method for Finding $\bar{\sigma}$

We can find the values of $\bar{k}$ and $\bar{L}$ the same way as in Model One. We can then use an implicit method to numerically solve for $\bar{\sigma}$. First we must discretize Equation (6.4a).

$$A_{i,i-1}\bar{\sigma}(\theta_{i-1}) + A_{i,i}\bar{\sigma}(\theta_i) + A_{i,i+1}\bar{\sigma}(\theta_{i+1}) = b \tag{6.9}$$

where the coefficients are:

$$\begin{aligned} A_{i,i-1} &= \frac{1}{\Delta\theta^2} - \frac{\bar{k}}{2a\Delta\theta} \\ A_{i,i} &= -\frac{2}{\Delta\theta^2} - \frac{1}{\bar{\mu}} \\ A_{i,i+1} &= \frac{1}{\Delta\theta^2} + \frac{\bar{k}}{2a\Delta\theta} \\ b_i &= -\frac{\bar{\tau}}{\bar{\mu}} \end{aligned} \tag{6.10}$$

Note that in all coefficients, the functions expressed in them are at θ_i regardless of what θ-step the corresponding $\bar{\sigma}$ value is. We then transform these equations into matrix notation so that:

$$\begin{pmatrix} \vdots \\ b_i \\ \vdots \end{pmatrix} = \begin{pmatrix} A_{1,1} & A_{1,2} & 0 & \dots & 0 \\ A_{2,1} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & A_{n-1,n} \\ 0 & \dots & 0 & A_{n,n-1} & A_{n,n} \end{pmatrix} \begin{pmatrix} \vdots \\ \bar{\sigma}(\theta_i) \\ \vdots \end{pmatrix}$$

We can then solve this linear system using Gaussian Elimination to obtain $\bar{\sigma}(\theta)$.

A proof of the existence and uniqueness of this Traveling Wave solution is currently being researched by Professor Roger Lui.

Bibliography

[1] Alberts, Bruce, [et al.]
Essential Cell Biology 2nd edition
Garland Science, 2003.

[2] Berg, Howard C.
Random Walks in Biology Rev. edition
Princeton University Press, 1984.

[3] Dembo, Micah and Yu-Li Wang
Stresses at the Cell-to-Substrate Interface during Locomotion of Fibroblasts,
Biophysical Journal, 1999, 76, pp. 2307-2316.

[4] Dembo, Micah, Chun-Min Lo, Hong-Bei Wang, and Yu-Li Wang
Cell Movement Is Guided by the Rigidity of the Substrate,
Biophysical Journal, 2000, 79, pp. 144-152.

[5] Fung, Y. C.
First Course in Continuum Mechanics 3rd edition,
Prentice Hall, 1994.

[6] Gracheva, Maria and Hans G. Othmer
A Continuum Model of Motility in Ameboid Cells,
Bulletin of Mathematical Biology, 2004, 66, pp. 167-193.

[7] Larippa, Kamila and Alex Mogilner
Transport of a 1D Viscoelastic Actin-Myosin Strip of gel as a Model of a Crawling Cell
2006, Physica A, 372, pp. 113-123.

[8] Palecek, S. P., J. C. Loftus, M. H. Ginsberg, D. A. Lauffenburger and A. F. Horwitz.
Integrin ligand binding properties govern cell migration speed through cellsubstratum adhesiveness.
1997, Nature 385, pp. 537-540.

[9] "Actin." Wikipedia, The Free Encyclopedia.
6 Jan 2007, 01:16 UTC. Wikimedia Foundation, Inc.

<http://en.wikipedia.org/w/index.php?title=Actin&oldid=98774354>.

[10] "Myosin." Wikipedia, The Free Encyclopedia.
22 Nov 2006, 05:24 UTC. Wikimedia Foundation, Inc.

<http://en.wikipedia.org/w/index.php?title=Myosin&oldid=89390077>.

Tables

Below are the constants and parameters used in our project.

Table 6.1: Time-Dependent Model One Parameters

Parameter	Definition	Value
E_0	Cell elastic constant	0.42 Pa
L_0	Initial celllLength	1 μm
M_{Total}	Total Myosin II	20.0 μM
N_{Total}	Total integrin	10^5 $receptors/cell$
V_0	Initial protrusion rate of the front	0.1 $\mu m \cdot s^{-1}$
V_d	Contraction rate of the rear	0.1 $\mu m \cdot s^{-1}$
κ_m^+/κ_m^-	Association/dissociation rate ratio	0.3
$\kappa_{Reg}^+[Reg]_0/\kappa_{Reg}^-$	Association/dissociation rate ratio	0.1
β_0	Drag constant	0.05 $Pa \cdot s \cdot cell/\mu m^2 \cdot receptors$
μ	Cell viscosity	2.0×10^{-3} $Pa \cdot s$
τ_0	Active stress constant	4.2×10^{-8} $Pa \cdot \mu m^3$
ψ_1	Myosin II dissociation assymetry	3.33
ψ_2	Myosin II activation asymmetry	10.0

Table 6.2: Traveling Wave Model One Parameters

Parameter	Definition	Scaled Value	Unscaled Value
E	Elastic Modulus	-	0.21 Pa
V_d	Contraction rate of the rear	0.4762 $Pa/\mu m^2$	0.1 $\mu m \cdot s^{-1}$
β	Viscous drag coefficient	-	0.2588 $Pa \cdot s/\mu m^2$
μ	Cell viscosity	5×10^{-4} $(Pa \cdot s/\mu m)^2$	2.0×10^{-3} $Pa \cdot s$
τ	Contractile stress	5.137×10^{-3}	1.0787×10^{-3} Pa

Time-Dependent Model One

Driver - driverfcn.m

```
tic
% Input Parameters
global ysteps
global tsteps
global sig0
global sig1
global rear0
global front0
global rear1
global front1

SetConstants()

fid1 = fopen('SigmaVals.txt','wt');
fid2 = fopen('Boundaries.txt','wt');

for t = 1:tsteps

%% Printing
    fprintf(fid2,'%12.12f  %12.12f\n',rear0,front0);
    for j = 1:ysteps
            fprintf(fid1,'%12.12f ',sig0(j));
    end
    fprintf(fid1,'\n');

%% Calculation
    %FindSigmaE()
    FindSigmaI()

%% ticker
if and(t~=tsteps,ceil(10*t/tsteps)==10*t/tsteps)
    p = round(t/tsteps *100);
    fprintf('%d %% done, %d / %d steps It time: %d \n',p,t,tsteps,toc)
    tic
end

%% Updating values
    sig0 = sig1;
    rear0 = rear1;
    front0 = front1;
end

fclose(fid1);
fclose(fid2);
```

Setting Function Constants - SetConstants.m

```
function SetConstants = SetConstants()

Constants = textread('constants.txt','%f','commentstyle','matlab');
% constants for beta
global beta0
global N_total
global psi1
global f0
global k_s
beta0 = Constants(1);
k_f = Constants(2);
k_r0 = Constants(3);
n_s = Constants(4);
N_total = Constants(5);
psi1 = Constants(6);
f0 = Constants(7);
k_s = k_f * n_s / k_r0;

% constants for elast
global elast0
elast0 = Constants(8);

% constants for FindSigma
global Mu0
Mu0 = Constants(9);

% constants for tau
global alph
global k_mneg
global k_mpos
global k_regneg
global k_regpos
global M_total
global n_b0
global psi2
global Reg0
global tau0
alph = Constants(10);
k_mneg = Constants(11);
k_mpos = Constants(12);
k_regneg = Constants(13);
k_regpos = Constants(14);
M_total = Constants(15);
n_b0 = Constants(16);
psi2 = Constants(17);
Reg0 = Constants(18);
tau0 = Constants(19);
```

```
% constants for driverfcn
global L0
global ysteps
global tsteps
global dy
global dt
global rear0
global front0
global Vd
global V0
global y
global sig0
L0 = 1;
ysteps = 200;
tsteps = 5000;
dy = 1/(ysteps + 1);
dt = dy;  %Implicit CFL condition
%dt = .5*dy^2; %Explicit Condition
rear0 = 0;
front0 = L0;
Vd = 0.1;
V0 = 0.1;

% Defining Initial Conditions
i = 1:ysteps;
y = i*dy;

for i = 1:ysteps
    % initial stress is parabola concave down, vanishes at 0 and L0.
    %sig0(i) = ((L0/2)^2 - (y(i) - L0/2)^2)/100;
    sig0(i) = sin(pi*y(i)/(.5*L0));
end
```

Text file of Constants - constants.txt

```
0.05          % beta0  (dyn*s/um^3)
0.0001        % k_f  **with n_s, k_r0 (ks)
1             % k_r0  **with n_s, k_f
1             % n_s  **with k_f, k_r0
100000        % N_total //
3.33          % psi1  (unitless)
.75           % f0

0.42          % elast0  (dyn/um)

0.002         % Mu0  (dyn*s/um)
```

```
2              % alph
1              % k_mneg  **with k_mpos
0.3            % k_mpos  **with k_mneg
1              % k_regneg  **with kreg_pos, Reg
1              % k_regpos  **with kreg_neg, Reg
50000          % n_b0 (not from papers)
20.0           % M_total  (uM) //input
10.0           % psi2  (unitless)
0.1            % Reg0  (unitless) **with k_regneg, k_regpos
0.000000042    % tau0  (dyn/um)
```

Explicit Method - FindSigmaE.m

```
global dy
global dt
global ysteps
global sig0
global sig1
global front0
global rear0
global front1
global rear1
global Mu0
global y
global beta0
global N_total
global k_s
global psi1

sigiAtCB = interpsig();
L = front0 - rear0;
Vp = V0*L0/L;
dR = dt*(Vd + betastar(0,0)*sigiAtCB(1)/(L*dy));
dF = dt*(Vp - betastar(1,0)*sigiAtCB(2)/(L*dy));

g = -dR/L - (dF - dR)*y/L;
dbetastardy = (1-psi1)/(k_s*N_total*beta0*L);
rear1 = rear0 + dR;
front1 = front0 + dF;

for i = 1:ysteps
    if i == 1
        dsigmahatdy = sig0(1)*dy;
        d2sigmahatdy2 = sig0(2) - 2*sig0(1);
    end
    if and(i >= 2, i <= ysteps-1)
        dsigmahatdy = .5*dy*(sig0(i+1) - sig0(i-1));
        d2sigmahatdy2 = sig0(i+1) - 2*sig0(i) + sig0(i-1);
```

```
    end
    if i == ysteps
        dsigmahatdy = -sig0(ysteps)*dy;
        d2sigmahatdy2 = -2*sig0(ysteps) + sig0(ysteps-1);
    end
    tmp1 = elasthat(y(i),0)*dt*dbetastardy/L - g(i)*dt;
    tmp2 = elasthat(y(i),0)*dt*betastar(y(i),0)/L^2;
    tmp3 = 1 - elasthat(y(i),0)*dt/Mu0;
    tmp4 = -elasthat(y(i),0)*dt*tauhat(y(i),0)/Mu0;

    sig1(i) = tmp1*dsigmahatdy + tmp2*d2sigmahatdy2 + tmp3*sig0(i) + tmp4;
end
```

Implicit Method - FindSigmaI.m

```
global ysteps
global L0
global dy
global dt
global sig0
global sig1
global rear0
global front0
global rear1
global front1
global V0
global Vd
global Mu0
global y
global beta0
global N_total
global k_s
global psi1
global L

L = front0 - rear0;
Vp = V0*L0/L;
dR = dt*(Vd + sig0(1)/(L*dy*beta(0)));
dF = dt*(Vp - sig0(ysteps)/(L*dy*beta(L)));

rear1 = rear0 + dR;
front1 = front0 + dF;
L = front1 - rear1;

Vp1 = V0*L0/L;
dF = dF - dt*(Vp - Vp1);

g = -(dR + (dF - dR)*y)/L; %g = dy/dt = [(x-r)/L]'
```

```
dB1dy = (1-psi1)/(k_s*N_total*beta0*L);

A = zeros(ysteps);

for i=1:ysteps
    %% Find A
    E = elast(y(i)*L);
    B = 1/beta(y(i)*L);
    T = tau(y(i)*L);
    if i~=1
        A(i,i-1) =  B/(L^2*dy) + g(i)/(2*E) - dB1dy/(2*L);
    end

    A(i,i) = -2*B/(L^2*dy) - dy/Mu0 - dy/(E*dt);

    if i~=ysteps
        A(i,i+1) = B/(L^2*dy) - g(i)/(2*E) + dB1dy/(2*L);
    end
    b(i) = (T/Mu0 - sig0(i)/(E*dt))*dy;
end

%% Solve A*sig1 = b for sig1
sig1 = (A \ b')';
```

Actin Concentration - act.m

```
function act = act(o)

act = o;
```

Elastic Modulus - elast.m

```
function elast = elast(o)

global elast0

elast = elast0*act(o);
```

Viscous Drag Coefficient - beta.m

```
function beta = beta(o)

global beta0
global k_s
global n_f
global psi1
global N_total
global L
```

```
f1 = psi1 + (1 - psi1)*o/L;
n_b = k_s * N_total / (f1 + k_s);

beta = beta0 * n_b;
```

Contractile Stress - tau.m

```
function tau = tau(o)

global tau0
global k_mpos
global k_mneg
global k_regpos
global k_regneg
global M_total
global Reg0
global psi2
global psi1
global N_total
global alph
global n_b0
global k_s
global L

f1 = psi1 + ( 1 - psi1 )*o/L;
n_b = k_s * N_total / ( f1 + k_s );
Reg = Reg0*n_b^alph/( n_b0^alph + n_b^alph );
f2 = ( psi2 + ( 1 - psi2 )*o/L )^(-1);
Kr = k_regpos/k_regneg;
Km = k_mpos/k_mneg;

c1 = Kr*Reg*f2;
c2 = 1/( Km*act(o) ); %inverse of a(x)
m_bpos = M_total*c1/( c1 + 1 + c2 );
tau = -tau0*m_bpos;
```

Steady-State Detection - FindSS.m

```
function SS = FindSS(t0,t1)

b = load('Boundaries.txt');

dt = 1/201; % for implicit with ysteps = 200, dt = dy, L0 = 1

i = t0:t1;
kr = polyfit(i',b(i,1),1);
kf = polyfit(i',b(i,2),1);

fprintf('f(t) = %d * x + %d \nr(t) = %d * x + %d \n',kf(1)/dt,kf(2),kr(1)/dt,kr(2))
```

```
Ls0 = b(t0,2) - b(t0,1);
Ls1 = b(t1,2) - b(t1,1);

fprintf('L(t0) = %d , L(t1) = %d \n \n',Ls0,Ls1)
```

Traveling Wave Model One

Driver - driverfcn.m

```
% Input Parameters (set in SetConstants())
global osteps
global sigw0
global sigw1
global k
global L
global L0
global do
global tsteps

SetConstants()

L = L0;
do = L/(osteps + 1);
fid1 = fopen('SigmawVals.txt','wt');
fid2 = fopen('ss.txt','wt');

for t = 1:tsteps

    % Printing
    fprintf(fid2,'%12.12f  %12.12f\n',k,L);
    for j = 1:osteps
            fprintf(fid1,'%12.12f ',sigw0(j));
    end
    fprintf(fid1,'\n');

    %Solve traveling wave
    FindSigmaw()

    % Updating values
    sigw0 = sigw1;
end
k_scaled = k/.21;
L_scaled = L*.2588;

fclose(fid1);
fclose(fid2);
```

Setting Constants - SetConstants.m

```
function Set = SetConstants()

s = load('SigmaVals.txt');

Constants = textread('constants.txt','%f','commentstyle','matlab');
% constants for beta
global beta0
%global k_f
%global k_r0
%global n_s
global N_total
global psi1
global f0
global k_s
beta0 = Constants(1);
k_f = Constants(2);
k_r0 = Constants(3);
n_s = Constants(4);
N_total = Constants(5);
psi1 = Constants(6);
f0 = Constants(7);
k_s = k_f * n_s / k_r0;

% constants for elast
global elast0
elast0 = Constants(8);

% constants for FindSigmaw
global Mu0
Mu0 = Constants(9);

% constants for tau
global alph
global k_mneg
global k_mpos
global k_regneg
global k_regpos
global M_total
global n_b0
global psi2
global Reg0
global tau0
alph = Constants(10);
k_mneg = Constants(11);
k_mpos = Constants(12);
k_regneg = Constants(13);
k_regpos = Constants(14);
```

```
M_total = Constants(15);
n_b0 = Constants(16);
psi2 = Constants(17);
Reg0 = Constants(18);
tau0 = Constants(19);

% constants for driverfcn
global L0
global osteps
global sigw0
global tsteps
global V0
global Vd
L0 = 1;
osteps = 200;
tsteps = 50;
V0 = 0.1;
Vd = 0.1;

for i = 1:osteps
    sigw0(i) = s(5000,i); % Time-dependent Steady State
end
```

Implicit Traveling Wave Method - FindSigmaw.m

```
function sigw = FindSigmaw()

global osteps
global sigw0
global sigw1
global L
global k
global do
global Mu0
global V0
global Vd
global L0

% Preliminary Calculations
k = Vd + sigw0(1)/(do*beta(0));
L = V0*L0/(k + sigw0(osteps)/(beta(L)*do));
do = L/(osteps + 1);

A = zeros(osteps);
for i=1:osteps
    %% Find A
    o = i*do;
    if i~=1
```

```
        A(i,i-1) = 1/do^2 - k*beta(o)/(2*elast(o)*do);
    end

    A(i,i) = -2/do^2 - beta(o)/Mu0;

    if i~=osteps
        A(i,i+1) = 1/do^2 + k*beta(o)/(2*elast(o)*do);
    end
end
%% Find b
for i=1:osteps
    b(i) = beta(o)*tau(o)/Mu0;
end

%% Solve A*sig1 = b for sig1
sigw1 = (A \ b')';
```

Note for the traveling wave, E, β, τ are defined constants.

Traveling Wave Model Two

Driver - driverfcn.m

```
function x = driverfcn(E0i,Gammai)

global actin
global osteps
global sigw0
global sigw1
global k
global L
global L0
global do
global tsteps
global Gamma
global E0
E0 = E0i;
Gamma = Gammai;

SetConstants()
L = L0;
do = L/(osteps + 1);
fid1 = fopen('SigmawVals.txt','wt');
fid2 = fopen('ss.txt','wt');
fid3 = fopen('ActinVals.txt','wt');

for t = 1:tsteps

    % Printing k and L
    fprintf(fid2,'%16.15f  %16.15f\n',k,L);
```

```
    %Find sigw1
    FindSigmaw()

    % Printing Sigw0 and actin
    for j = 1:osteps
        fprintf(fid1,'%16.15f ',sigw0(j));
        fprintf(fid3,'%16.15f ',actin(j));
    end
    fprintf(fid1,'\n');
    fprintf(fid3,'\n');

    % Updating values
    sigw0 = sigw1;
end

fclose(fid1);
fclose(fid2);
fclose(fid3);

x = [k,L];
```

Finding Stress Implicitly - FindSigmaw.m

```
function FindSigmaw()

global osteps
global actin
global sigw0
global sigw1
global L
global k
global do
global Mu0
global tau
global Vd
global L0

%% Preliminary Calculations
k = Vd + (4*sigw0(1) - sigw0(2))/(2*do);
L = L0/(k - (sigw0(osteps-1)-4*sigw0(osteps))/(2*do));
do = L/(osteps + 1);

%% Find actin
FindAct()

%% Find A and b
A = zeros(osteps);
```

```
b = zeros(1,osteps);
for i=1:osteps
    a = 1/actin(i);

    if i~=1
        A(i,i-1) = 1/do^2 - a*k/(2*do);
    end

    A(i,i) = -2/do^2 - 1/Mu0;

    if i~=osteps
        A(i,i+1) = 1/do^2 + a*k/(2*do);
    end

    b(i) = -tau/Mu0;
end

%% Solve A*sig1 = b for sig1
sigw1 = (A \ b')';
```

Finding Actin Concentration - FindAct.m

```
function FindAct()

global actin
global osteps
global k
global do
global L0
global Gamma
global L

I = 0;
tmp = zeros(1,osteps);
for j=0:osteps-1
    i = osteps+1-j;  %count backwards
    F = Gamma/(k - sigp(i));% + sigpp(i)/(k-sigp(i));  %integrand
    I = I + F*do;  %integral
    tmp(i-1) = exp(-I)/abs(k-sigp(i-1));
end

actin = L0*tmp/L;
```