# Cytoskeleton Micromechanics: A Continuum-Microscopic Approach

Jennifer J. Young

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics.

Chapel Hill
2010

Approved by

Advisor: Sorin Mitran
Reader: Timothy Elston
Reader: M. Gregory Forest
Reader: Kenneth Jacobson
Reader: Laura Miller

# ABSTRACT

JENNIFER J. YOUNG: Cytoskeleton Micromechanics: A Continuum-Microscopic Approach

(Under the direction of Sorin Mitran)

Creating accurate, whole-cell scale models of the cytoskeleton is computationally challenging, due to the material's highly heterogeneous microstructure. Continuum-based models, homogenization methods, and coarse grained models are common modeling approaches. These methods utilize constant-in-time, average mechanical properties, whereas continuum-microscopic (CM) models utilize a microscopic model to periodically update local mechanical parameters for a macroscopic model. CM methods have been used for heterogeneous media with unchanging microstructures. This research focuses on extending a basic CM algorithm to model heterogeneous media with time-varying microstructures. Microscopic data is saved over time in the form of probability distribution functions. These PDFs are then extrapolated forward in time to predict what the microstructure will look like in the future. Keeping track of the microstructure over time allows for the accurate computation of the local mechanical parameters used in the continuum-level equations. The model was tested on a rectangular domain, representative of a cytoskeleton. Results showed that the elastic parameters computed with this algorithm are similar to those computed with a fully-microscopic simulation. Errors for continuum level variables (such as stress) in the 10% range are deemed an acceptable trade-off for the $50 - 75\%$ savings in computational expense offered by this algorithm.

# ACKNOWLEDGEMENTS

I would like to thank everyone who has helped and supported me along the way with my thesis. First of all I would like to thank my family, especially my mom and dad, my husband Gavan, my parents in law, and my two nonnas. I would also like to thank Oreo, my furry companion during the thesis writing process.

Secondly, I would like to thank my professors, especially my advisor, Sorin Mitran. I have learned a great deal from Dr. Mitran and I thank him for his patience and support over the past six years. I would also like to thank my other committee members: Greg Forest, Laura Miller, Tim Elston and Ken Jacobson, for their help and encouragement. I would also like to thank the other professors in the department who have taught me a great deal as well.

Third, I would like to thank my friends both here at UNC and back home in Delaware. Thanks to the UNC crew: Joyce, Paul, Roxana, Andy, Martina, Claudio, Nathan and Swathi for making my time here in Chapel Hill truly enjoyable. Thanks to my Delaware girls: Renate, Jacquie and Liz for the long-distance love and support.

Finally, I would like to thank God, la Madonnina and all the saints I havee prayed to for wisdom and guidance, who have given me hope and faith when I needed it most.

# CONTENTS

# LIST OF FIGURES

CHAPTER 1

# Introduction

Many natural phenomena involve multiple length and time scales. These different scales arise as a result of the discrete nature of matter. As an illustrative example of a separation of length scales, consider an ideal gas, which is composed of discrete molecules. The mean free path $\lambda$ is the average distance a molecule travels before colliding with another molecule. Three basic length scales can be defined by introducing a dimensionless value $K$, (known as the Knudsen number) defined as $\lambda/L$ where $L$ is the physical length scale of interest. If $K \ll 1$ then the length scale of interest is much larger than the mean free path. With $K \approx 1$, the length scale of interest is close in magnitude to $\lambda$, and collisions between molecules become observable. Lastly, if $K \gg 1$, then the length scale of interest is small compared to $\lambda$. Separation of scales also exists for time. Using again a gas example, consider a mole sample of gas in an enclosed macroscopic sized container that is in a state of non-equilibrium. Three time scales, $\tau_1, \tau_2, \tau_3$, can be established with $\tau_1$ the average time it takes a molecule to traverse the length of the container, $\tau_2$ the average time between collisions, and $\tau_3$ the average time it takes for a collision to occur [**52**]. The particular phenomenon to be modeled will dictate the scale (or scales) that are important to focus on to capture the desired behavior.

Each scale warrants a different modeling approach. As a general illustration consider the dynamics of fluids. The fluid motion for the $K \ll 1$ case (e.g. river flow) can be well described by the Navier-Stokes equations. At the $K < 1$ level, a Boltzmann-like equation can be employed to model the dynamics of the fluid particle distribution. At $K \approx 1$, Newton's law of molecular dynamics can be applied to model the positions and

velocities of individual fluid molecules. Zooming in to the $K \gg 1$ case the Schrödinger equation of quantum mechanics can be utilized [**25**].

When creating a computational model, the goal is to include as many components as needed to accurately capture a specific behavior. A more detailed model that includes many of the smaller scale components will increase the accuracy of the model, but may significantly slow down the computation. Numerical analysts seek to strike a balance between creating realistic but efficient models. Going from the above scale descriptions, if the goal is to model flow in a river the most logical choice would be to employ the Navier-Stokes equations. Newton's law could also be used to model the individual particles to capture the same overall fluid motion, but due to the large number of molecules in a region of river the molecular dynamics computation would be computationally intractable. (More details on computational difficulties will be presented in chapter 2.) The decision is made to view the river's fluid as a *continuum*, meaning that the heterogeneous, microscopic structure of the material is ignored [**47**] and is treated instead as a continuous, homogeneous fluid.

The choice to model a material as a continuum is motivated by the economy of description. If the variables describing the discrete, microscopic elements can be averaged without the loss of information then a continuum description is an efficient model choice. However, there are some instances where fluctuations and structural heterogeneities at the microscopic scale, do affect behavior at the continuum level. In these cases, a purely continuum-based model would likely fail to capture these microscopic effects. The development of computational models of the dynamics of continuous media which include data from the microscopic scale is the subject of the following thesis work. The goal of this research is to create efficient numerical models that approximate macroscopic constitutive laws which accurately capture the non-equilibrium microscopic behavior that is observable in deforming biological material, such as the cell cytoskeleton. To achieve this goal, concepts from kinetic theory, statistical mechanics, and probability theory are utilized to create a continuum-microscopic model that retains memory of the microscopic

variables without having to carry out a full microscopic simulation. The model is demonstrated for the example case of the cell cytoskeleton, a highly heterogeneous, crosslinked filament network.

The structure of the thesis is as follows. First a brief synopsis of the history of continuum mechanics, its governing equations and basic examples will be presented. Next, biological background on the cell cytoskeleton will be described, followed by a presentation of current research in cytoskeletal modeling. Chapter 3 will be comprised of two motivating examples that demonstrate the need for a more comprehensive, microscopically-informed continuum model of the cytoskeleton. In Chapter 4, general mathematical theory needed to construct the model will be presented. This will be followed by Chapter 5, which describes the development of the main algorithm. Finally, in Chapter 6, a basic application of the algorithm will be demonstrated, and the chapter will conclude with ideas for future work.

## 1.1. The Development of Continuum Mechanics

Describing a material as continuous ignores the media's discrete, molecular structure [**47**] and instead characterizes it as being comprised of homogenized matter that completely fills the space. The study of the mechanical behavior of continua is the subject of continuum mechanics [**48**]. A brief history of continuum mechanics will first be presented. This will be followed by derivations of the elementary notions of stress and strain put forth by scientists such as Hooke, Navier, Cauchy and Poisson. This will be followed by a presentation of the equations of motion for basic continuous, homogeneous bodies such as a string, membrane, beam and plate. Composite materials will then be discussed, as a lead-in to the more complicated modeling of highly heterogeneous materials. Finally, current multiscale and continuum-microscopic modeling techniques will be described.

**1.1.1. Early History.** The experimental work of Robert Hooke led to his 1660 discovery of a linear relationship between stress and strain in spring-like objects. This law was stated in Hooke's *De Potentia Restitutiva* with the Latin phrase "Ut tensio sic vis",

meaning "As the expansion, so the force". After this discovery, the next one hundred fifty years of research included: (1) work related to the theory of beams and thin rods by Leonhard Euler, Jacob Bernoulli and Daniel Bernoulli; (2) Coulomb's discussion of a non-extensional strain known today as shear; (3) the solidification of the idea of an elasticity modulus of a material by Thomas Young; (4) the extension of rod theory to shells and plates by Euler, a younger Jacob Bernoulli, and Sophie Germaine; (5) Newton's concept of media, that a material is composed of small, finite-sized parts that interact through forces. These notions laid the groundwork for the theory and equations of stress and strain put forth by Navier, Cauchy and Poisson in the early 1800's.

Navier took the Newtonian concept of a solid and formulated the first general equations to describe the equilibrium and vibrational states of elastic bodies. He replaced the notion of discrete molecules (that to Newton had finite size) with the idea of an infinitesimal material point. He assumed that the forces between two material points in a body when the distance between them was changed, was proportional to the change increment and to the initial distance between the points. His equations of motion are written in terms of displacements, and they contain a meterial-specific constant (the first notion of a material-specific elasticity modulus). In 1821, Fresnel introduced the novel idea that waves could travel through an elastic medium. This concept attracted both Cauchy and Poisson to the study of elasticity. Cauchy soon worked out many of the fundamental ideas of elasticity such as stress, strain and the principal axes of stress and strain. He also formulated his own equation of motion, similar to Navier's, but it utilized two elastic constants as opposed to one. Concurrently, Poisson developed a similar set of equations and also contributed many interesting applications of the general theory. (All historical information from [**54**].)

**1.1.2. Strain.** Strain is a measure of how much an object has been stretched, compared to its equilibrium state. Begin with an elastic body and a point on the body with position vector $\mathbf{r} = \langle x_1, x_2, x_3 \rangle$ in Cartesian space. (Vectors will always be noted in bold face.) If this body gets deformed, point $\mathbf{r}$ is moved to $\mathbf{r}'$. Let $\mathbf{u} = \mathbf{r}' - \mathbf{r}$ be the displacement

vector, and in component form it can be written as: $u_i = x'_i - x_i$ where $i = 1, 2, 3$. Take two points on the body $\mathbf{r}_1$ and $\mathbf{r}_2$ that are a small distance apart (infinitesimal) and compute the vector between them to be: $\mathbf{dr} = \mathbf{r}_1 - \mathbf{r}_2$. These points are then displaced to new locations $\mathbf{r}'_1$ and $\mathbf{r}'_2$ after deformation has occurred. The relations: $\mathbf{u}_1 = \mathbf{r}'_1 - \mathbf{r}_1$ and $\mathbf{u}_2 = \mathbf{r}'_2 - \mathbf{r}_2$ and also $\mathbf{du} = \mathbf{u}_1 - \mathbf{u}_2$ can be established (See Figure 1.1 for a diagram).

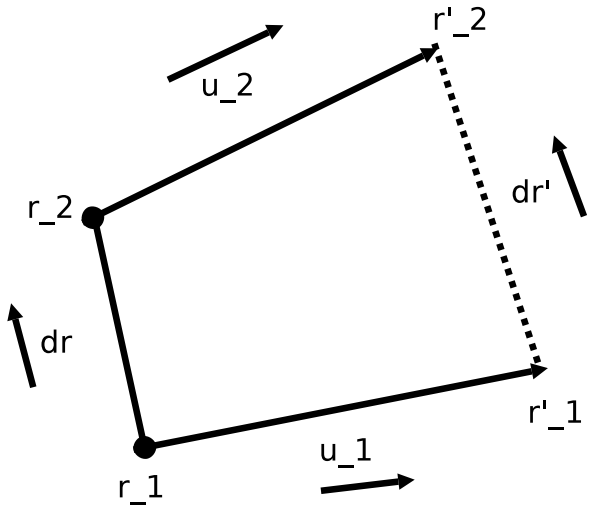


FIGURE 1.1. Diagram of displacement of infinitesimal line segment

The vector between the new points can be written in terms of the initial points and displacements as follows:

$$\begin{aligned}
\mathbf{dr}' &= \mathbf{r}'_1 - \mathbf{r}'_2 \\
\mathbf{dr}' &= (\mathbf{r}_1 + \mathbf{u}_1) - (\mathbf{r}_2 + \mathbf{u}_2) \\
\mathbf{dr}' &= (\mathbf{r}_1 - \mathbf{r}_2) + (\mathbf{u}_1 - \mathbf{u}_2) \\
\mathbf{dr}' &= \mathbf{dr} + \mathbf{du}
\end{aligned}$$

The length of a vector between two points is simply the distance formula:
$|\mathbf{dr}| = \sqrt{dx_1^2 + dx_2^2 + dx_3^2}$. The length between the two deformed points can be expressed as:

$$|\mathbf{dr}'| = |\mathbf{dr} + \mathbf{du}|$$

Call this deformed length $l = |\mathbf{dr}'|$ and $L = |\mathbf{dr}|$ will be the undeformed length. The difference $l - L$ is known as the *stretch*. The expression $l^2 - L^2$ is a measure of deformation, and can be utilized to define strain in terms of displacements. The current squared length can be written as:

$$\begin{aligned} l^2 &= |\mathbf{dr} + \mathbf{du}|^2 \\ l^2 &= (dx_1^2 + dx_2^2 + dx_3^2) + 2dx_1du_1 + 2dx_2du_2 + 2dx_3du_3 + du_1^2 + du_2^2 + du_3^2 \\ l^2 &= L^2 + 2\sum_{i=1}^{3} dx_i du_i + \sum_{i=1}^{3} du_i^2 \end{aligned} \tag{1.1}$$

The differential $du_i$ can be replaced with $\sum_{j=1}^{3} \frac{\partial u_i}{\partial x_j} dx_j$, and Equation 1.1 can be rewritten as:

$$l^2 = L^2 + 2\sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial u_i}{\partial x_j} dx_j dx_i + \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} \frac{\partial u_i}{\partial x_j} dx_j \frac{\partial u_i}{\partial x_k} dx_k \tag{1.2}$$

The second term can be rewritten as:

$$\sum_{i=1}^{3}\sum_{j=1}^{3} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) dx_i dx_j$$

(in a symmetric form), and in the third term the $i$ and $k$ indices can be swapped:

$$\frac{\partial u_k}{\partial x_j} \frac{\partial u_k}{\partial x_i} dx_i dx_j$$

Replacing these two expressions back into 1.2 (and using repeated index summation notation) gives:

$$l^2 = L^2 + \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) dx_i dx_j + \frac{\partial u_k}{\partial x_j} \frac{\partial u_k}{\partial x_i} dx_i dx_j$$

Grouping the last two terms together:

$$l^2 - L^2 = \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_j} \frac{\partial u_k}{\partial x_i} \right) dx_i dx_j \tag{1.3}$$

Equation 1.3 can be written succinctly as:

$$l^2 - L^2 = 2E_{ij}dx_i dx_j$$

where $E_{ij}$ is the Green strain tensor:

$$E_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_j}\frac{\partial u_k}{\partial x_i}\right)$$

Tensor $E_{ij}$ is symmetric and has been derived from a Lagrangian formulation where points in the body are labeled by their initial states. It makes intuitive sense as an expression describing strain because if there is no change in length ($l^2 - L^2 = 0$) then $E_{ij} = 0$. This includes rigid body motion (translations and rotations), whose displacements may be non-zero but $E_{ij}$ still vanishes. This tensor can also be easily related to the notion of strain as the ratio of change in length to equilibrium length:

$$\begin{aligned} l^2 - L^2 &= 2E_{ij}dx_i dx_j \\ (l-L)(l+L) &= 2E_{ij}dx_i dx_j \\ 2L(l-L) &\approx 2E_{ij}dx_i dx_j \\ \frac{l-L}{L} &\approx E_{ij}\hat{dx}_i\hat{dx}_j \end{aligned}$$

where $\hat{dx}_i$ and $\hat{dx}_j$ are simply normalized vector components.

The above strain tensor $\underline{\underline{\mathbf{E}}}$ is valid for both large and small deformation situations and is also commonly known as the finite strain tensor. It can be written in terms of the displacement gradient tensor $\nabla\mathbf{u}$ as:

$$\underline{\underline{\mathbf{E}}} = \frac{1}{2}((\nabla\mathbf{u})^T + \nabla\mathbf{u} + (\nabla\mathbf{u})^T \cdot \nabla\mathbf{u})$$

If deformations are small, meaning $\partial u_i/\partial x_j \ll 1$ then the finite strain tensor can be reduced to the infinitesimal strain tensor:

$$\underline{\underline{\mathbf{E}}} = \frac{1}{2}((\nabla \mathbf{u})^T + \nabla \mathbf{u})$$

This is a result of the fact that if displacement gradients are small, then the product of two such quantities: $\frac{\partial u_k}{\partial x_j}\frac{\partial u_k}{\partial x_i}$, will be several orders of magnitude smaller and can thus be neglected. (Derivation from [**36, 48**]).

Both notions of finite and infinitesimal strains will be used in the following thesis work. The cytoskeleton experiences large strains on the length and time scales of the formation of whole cell protrusions. On the microscopic scale, small displacements occur rapidly due to thermal and biochemical flucations, and at this scale infinitesimal strain theory can be assumed over successive micro-time steps.

**1.1.3. Stress.** Stress is defined as the average force per unit area.

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

Like strain, there are different representations of stress depending on the magnitude of deformation. The Cauchy stress tensor (written in an Eulerian framework) is valid for small deformations (where the deformed and reference states are very close), and is expressed as:

$$\underline{\underline{\sigma}} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

Given an infinitesimal cube upon which surface forces are acting, the tensor can be interpreted as follows: the first index $i$ indicates the direction of the force and the second index $j$ gives the direction of the normal vector to the surface. (See Figure 1.2 for a diagram.) Normal stresses (stresses acting on a surface perpendicular to that surface) are along the diagonal of the stress tensor. Tangential (or shear) stresses appear in the off diagonal positions [**48**]).

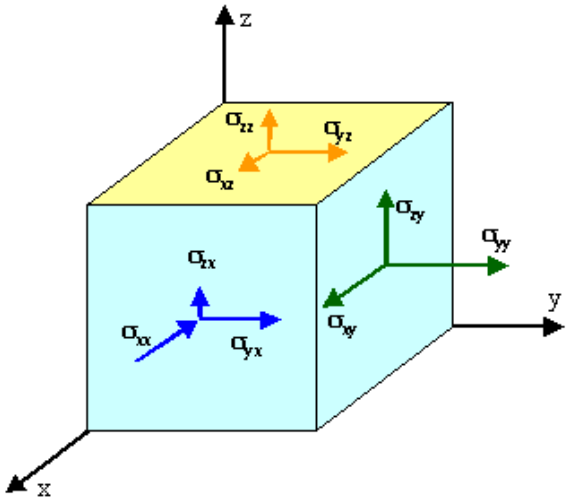


FIGURE 1.2. Components of the Cauchy stress tensor (from www.stiintaazi.ro)

The Cauchy stress tensor can be utilized to find the stress on any surface element $dA$ of an elastic body, as long as the normal vector to that surface is known. This relationship is simply

$$\mathbf{T} = \underline{\underline{\sigma}} \cdot \mathbf{n}$$

where $\mathbf{T}$ is the computed stress vector on the surface of interest, and $\mathbf{n}$ is the normal vector to that surface [**36**].

The Cauchy stress tensor is symmetric. This can be shown by utilizing concepts of angular momentum and moments. The angular momentum vector $\mathbf{L}$ can be defined as $\mathbf{L} = I\boldsymbol{\omega}$ where $I$ is the moment of inertia and $\boldsymbol{\omega}$ is the angular velocity. A moment $\mathbf{M}$ is defined as the cross product of a moment arm vector $\mathbf{r}$ and a force vector $\mathbf{F}$, and describes the rotational forces on an object. The following relationship can be established by Newton's second law:

$$\begin{aligned} \frac{d\mathbf{L}}{dt} &= \sum \mathbf{M} \\ \frac{d(I\boldsymbol{\omega})}{dt} &= \sum \mathbf{r} \times \mathbf{F} \\ I\boldsymbol{\alpha} &= \sum \mathbf{r} \times \mathbf{F} \end{aligned}$$

where $\boldsymbol{\alpha}$ is angular acceleration. Using Figure 1.2, consider the net moment that would cause a rotation of the cube around an axis placed through the centers of the top and bottom faces. The forces acting on the four sides around the cube that could cause rotations are the four shear forces: $\sigma_{xy}dxdz$ and $\sigma_{yx}dydz$ (each repeated twice). Given the forces along with corresponding moment arms, all with their correct signs so as to produce the same direction of rotation, the net moment (in the $z$ direction) can be written as:

$$\begin{aligned}
\sum M_z &= \sigma_{xy}dxdz\frac{dy}{2} - \sigma_{yx}dydz\frac{dx}{2} + \sigma_{xy}dxdz\frac{dy}{2} - \sigma_{yx}dydz\frac{dx}{2} \\
\sum M_z &= \sigma_{xy}dxdydz - \sigma_{yx}dxdydz \\
\sum M_z &= (\sigma_{xy} - \sigma_{yx})dxdydz \qquad (1.4) \\
I\alpha_z &= (\sigma_{xy} - \sigma_{yx})dxdydz \qquad (1.5)
\end{aligned}$$

The moment of inertia of a cube rotating around the axis described above is given by:

$$I = \frac{\rho}{12}dxdydz((dx)^2 + (dy)^2) \qquad (1.6)$$

where $\rho$ is a mass density. Substituting 1.6 into 1.5:

$$\frac{\rho}{12}dxdydz((dx)^2 + (dy)^2)\alpha_z = (\sigma_{xy} - \sigma_{yx})dxdydz$$

This relationship has to hold even as the cube is made arbitrarily small. Cancelling $dxdydz$ from both sides and sending both $dx$ and $dy$ to zero means that the angular acceleration $\boldsymbol{\alpha}$ will tend to infinity in order to equal the quantity: $(\sigma_{xy} - \sigma_{yx})$. An infinite angular acceleration is non-physical, thus $\sigma_{xy} - \sigma_{yx} = 0$ or $\sigma_{xy} = \sigma_{yx}$. Similar results are obtained by constructing equations for the two other axes of rotation. The conclusion is that:

$$\sigma_{ij} = \sigma_{ji}$$

As mentioned earlier, the Cauchy stress tensor is valid for small deformations. For large deformations, the Piola-Kirchhoff stress tensors (first and second) provide Lagrangian descriptions of stress expressing the forces and areas in terms of reference states. To define these two stress tensors, let $\mathbf{X}$ be the initial position vector of a point, and let $\mathbf{x}(\mathbf{X}, t)$ be the current position vector. The displacement vector can be defined as: $\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X}$. The displacement gradient tensor can be expressed as:

$$\nabla_{\mathbf{X}}\mathbf{u} = \nabla_{\mathbf{X}}\mathbf{x} - \underline{\underline{\mathbf{I}}}$$

where $\underline{\underline{\mathbf{I}}}$ is the identity tensor and the gradient was taken with respect to the reference vector $\mathbf{X}$. The quantity $\nabla_{\mathbf{X}}\mathbf{x}$ is often labeled as $\underline{\underline{\mathbf{F}}}$ and is called the deformation gradient tensor. The first Piola-Kirchhoff stress tensor $\underline{\underline{\mathbf{P}}}$ is defined as:

$$\underline{\underline{\mathbf{P}}} = J\underline{\underline{\boldsymbol{\sigma}}} \cdot \underline{\underline{\mathbf{F}}}^{-T}$$

where $\underline{\underline{\boldsymbol{\sigma}}}$ is the Cauchy stress tensor, the $-T$ indicates the inverse transpose, and $J = det(\underline{\underline{\mathbf{F}}})$. This stress tensor relates forces in the current state to areas in the reference state.

The second Piola-Kirchhoff stress tensor $\underline{\underline{\mathbf{S}}}$ expresses both the forces and areas in terms of the reference state. This tensor is defined as:

$$\underline{\underline{\mathbf{S}}} = J\underline{\underline{\mathbf{F}}}^{-1} \cdot \underline{\underline{\boldsymbol{\sigma}}} \cdot \underline{\underline{\mathbf{F}}}^{-T}$$

A linear relationship can be established between stress and strain by generalizing Hooke's law:

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl}$$

where $C_{ijkl}$ is the stiffness tensor. Both $\sigma$ and $\epsilon$ are second order tensors with 9 elements each. The tensor $C_{ijkl}$ is a fourth order tensor, containing 81 elements. In the case of Cauchy stress, the symmetry of $\sigma$ and $\epsilon$, along with a strain energy relationship, allow $C_{ijkl}$ to be reduced to 21 independent entries. If the body is isotropic, meaning that it

has the same material properties independent of direction, then these 21 values can be reduced to two. Extensional behavior and shear behavior will each be described by one material parameter.

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + 2\mu\delta_{ik}\delta_{jl}$$

where $\lambda$ and $\mu$ are the Lamé coefficients [**36**], and the $\delta$'s are Kronecker deltas. The linear stress-strain relationship can be written succinctly as:

$$\sigma_{ij} = \lambda\delta_{ij}\epsilon_{kk} + 2\mu\epsilon_{ij}$$

Stress and strain can be related nonlinearly as well. Determining an accurate stress-strain relationship (denoted as a function of the general form: $\sigma(\epsilon)$) for a particular material is one of the major goals of continuum mechanics research. These constitutive relationships depend on the structure and material properties of the medium as well as the type and magnitude of forces applied. There is also the potential added complexity of time variability. This is particularly true for biological materials (like the cell cytoskeleton), whose material properties change over time due to structural rearrangements induced by biochemical reactions and external forces [**2, 27, 82**]. These structural changes occur at the microscopic level. To accurately model such a material at the continuum-level will require information from the micro-scale. This information must be periodically fed to the continuum level to keep the constitutive law up-to-date.

Developing an accurate continuum-scale, computational model of the deformation of the cell cytoskeleton is the main goal of this thesis. In order to model deformation, the dynamic elasticity equations will be solved. These equations require a relationship between stress and strain in order to close the system. Numerically modeling this relationship at the continuum level, utilizing time-varying microscopic data in an efficient manner is the central focus of this research work.

## 1.2. Simple Continuum Models

The derivation of continuum-level equations of motion for a homogeneous string, membrane, bean and plate will now be presented.

**1.2.1. String.** The equation for longitudinal motion in a uniform string can be derived by starting with a discrete mass-spring system (Figure 1.3).


FIGURE 1.3. A simple mass-spring system

This discrete system is composed of $N$ point masses (each of mass $m$), that in the zero stretch state are spaced a distance $h$ apart from one another. Let $u_j(t)$ be the displacement of mass $j$ at time $t$ from its zero stretch position. Between every two masses, place a spring of stiffness $k$ and equilibrium length $h$. Newton's Law ($F = ma$) can be used to describe the motion of this mass-spring system. Mass $j$ will feel forces from masses $j-1$ and $j+1$. The equation of motion for mass $j$ can be written as:

$$\begin{aligned} m\ddot{u}_j(t) &= k\left[(u_{j+1}(t) - u_j(t)) - (u_j(t) - u_{j-1}(t))\right] \\ m\ddot{u}_j(t) &= k\left[u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)\right] \\ \ddot{u}_j(t) &= \frac{kh^2}{m}\frac{\left[u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)\right]}{h^2} \end{aligned}$$

Taking the limit of $h \to 0$, gives:

$$u_{tt}(x,t) = \frac{E}{\lambda}u_{xx}(x,t)$$

where zero stretch position $x \in \mathbb{R}$ replaces the $j$ integer valued numbering system for the discrete point mass positions. This is the familiar wave equation with $E = kh$ the Young's modulus of the string material and $\lambda = m/h$ a linear mass density. This is a continuum-level constitutive law valid for describing the longitudinal motion of the string whose parameters $(E, \lambda)$ do not depend on the microscopic scale data.

A similar equation can be derived for transverse motion in a uniform string. Let $w(x,t)$ be the transverse displacement of the string above the $x$ axis (See Figure 1.4).

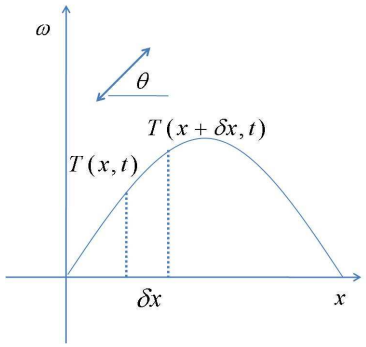


FIGURE 1.4. The transverse motion of an elastic string

When a string is displaced transversely, a tension $T$ is created in the segment. This tension can be represented as a vector $\mathbf{T}$ with $x$ and $y$ components by using the sin and cos of the angle $\theta$ the string makes with the $x$ axis.

$$\mathbf{T} = \left[ \begin{array}{c} T \cdot \cos(\theta) \\ T \cdot \sin(\theta) \end{array} \right]_{x}^{x+\delta x}$$

$\delta x$ is the length of the segment of string under this tension. Assuming the transverse displacements are small, $\sin(\theta)$ can be approximated by $\frac{\delta w}{\delta x} \approx \frac{\partial w}{\partial x}$. Similarly, $\cos(\theta)$ can be assumed to be close to 1. This gives the tension force as:

$$\mathbf{T} = \left[ \begin{array}{c} T \\ T\frac{\partial w}{\partial x} \end{array} \right]_{x}^{x+\delta x}$$

Newton's law can be used to write an equation for the transverse motion of the string. For this example, the transverse direction is the $y$ direction, thus the equation is:

$$\lambda w_{tt}(x,t) = T w_{xx}(x,t)$$

where $\lambda$ is again a linear mass density. The right hand side comes from the second component of the vector $\frac{\partial \mathbf{T}}{\partial x}$ which gives the net force acting on the string in the $y$ direction. This is once again a wave equation like in the longitudinal case [**36**].

**1.2.2. Membrane.** The equation of transverse motion for a thin,flexible membrane is an extension of the elastic string model to two dimensions. Begin with a two-dimensional mass-spring system as seen in Figure 1.5.

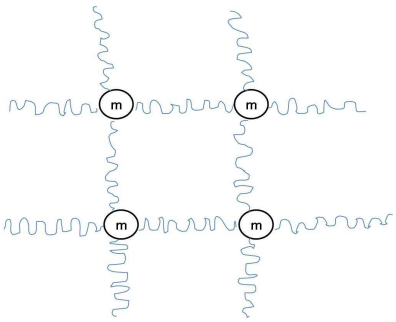


FIGURE 1.5. A two-dimensional mass-spring system

There are $N \times N$ masses in the discrete system, each of mass $m$. A spring of stiffness $k$ and equilibrium length $h$ is placed between each pair of masses, creating a two-dimensional grid (Figure 1.5). Let $w_{i,j}(t)$ be the transverse displacement of mass $i, j$ from its zero stretch state (Figure 1.6). The inertial portion of Newton's law for mass $i, j$ is given by $m\ddot{w}_{i,j}$. The forces acting on that mass stem from the differences in displacement between that point and the four surrounding spring-connected mass points. There are two springs that emanate from the mass in the $x$ direction and two in the $y$ direction. The forces can be calculated form Hooke's law as was done for the string:

$$\begin{aligned} F &= k[(w_{i+1,j}(t) - w_{i,j}(t)) - (w_{i,j}(t) - w_{i-1,j}(t)) \\ &+ (w_{i,j+1}(t) - w_{i,j}(t)) - (w_{i,j}(t) - w_{i,j-1}(t))] \\ F &= k(w_{i+1,j}(t) - 2w_{i,j}(t) + w_{i-1,j}(t) + w_{i,j+1}(t) - 2w_{i,j}(t) + w_{i,j-1}(t)) \\ m\ddot{w}_{i,j}(t) &= kh^2 \frac{w_{i+1,j}(t) - 2w_{i,j}(t) + w_{i-1,j}(t) + w_{i,j+1}(t) - 2w_{i,j}(t) + w_{i,j-1}(t}{h^2} \end{aligned}$$

As in the string case, take the limit as $h \to 0$ to obtain

$$w_{tt}(x, y, t) = \frac{T}{\lambda}(w_{xx}(x, y, t) + w_{yy}(x, y, t))$$

where the positions $x \in \mathbb{R}, y \in \mathbb{R}$ replace the integer indexed mass points, $\lambda = m/h^2$ is a mass per unit area parameter, and $T = k$ is the tension per unit length. This is a two-dimensional wave equation [**60**].

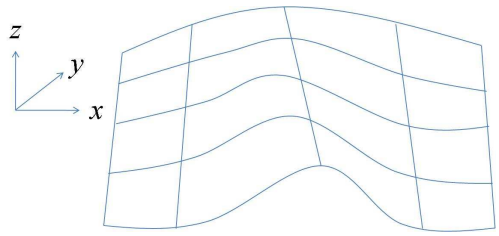


FIGURE 1.6. An elastic membrane undergoing transverse displacement

**1.2.3. Beam.** For the case of a beam, the same longitudinal equation of motion can be derived as in the string case as long as the cross-sectional area $A$ of the beam is uniform. The transverse motion equation will be different because of the added dimension of cross-sectional area which introduces an internal structure to the beam. Besides tension, a beam also experiences shear as its inside layers rub over one another as the beam bends. This shear force will be labeled $N$ and its direction is chosen as the transverse direction (See Figure 1.7 for force diagram).

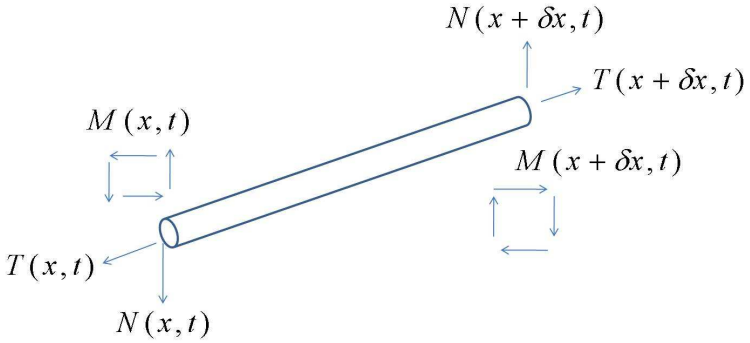


FIGURE 1.7. The forces and moments on a segment of beam

The shear force vector therefore looks like:

$$\mathbf{N} = \begin{bmatrix} 0 \\ N \end{bmatrix}_{x}^{x+\delta x}$$

The transverse displacement equation can be written as:

$$T\frac{\partial^2 w}{\partial x^2} + \frac{\partial N}{\partial x} = \lambda \frac{\partial^2 w}{\partial t^2}$$

The next step is to relate $N$ and $w$. This is done by first relating the shear force with the bending moment $\mathbf{M}$. By Newton's second law:

$$\frac{d\mathbf{L}}{dt} = \sum \mathbf{M}$$

where $\mathbf{L}$ is the angular momentum. When a section of beam, (length $\delta x$) is bent, its rotational forces include the two moments acting on each end, as well as the shear force $N$. This gives:

$$\frac{dL_z}{dt} = -M_z(x+\delta x, t) + M_z(x,t) + \delta x N(x+\delta x, t) \tag{1.7}$$

The subscript $z$ indicates the vector component of interest for this computation, since the rotation is occurring around the $z$ axis. The shear force $N$ is multiplied by the length of its moment arm to compute its moment. By the same argument presented in the symmetry discussion for the Cauchy stress tensor, the only way for Equation 1.7 to be physically correct is if (taking the limit as $\delta x \to 0$):

$$\frac{\partial M_z}{\partial x} = N \tag{1.8}$$

The moment can be written in terms of the transverse displacement $w$ because the moment is proportional to the curvature of the beam. The curvature is given by the second derivative with respect to arc length of $w$. The proportionality constant is $EI$ (the Young's modulus times the moment of inertia for the beam's cross-section):

$$M_z = EI\frac{\partial^2 w}{\partial x^2} \tag{1.9}$$

Utilizing relationships 1.8 and 1.9, the transverse motion of the beam can be written as:

$$EI\frac{\partial^4 w}{\partial x^4} + T\frac{\partial^2 w}{\partial x^2} = \lambda \frac{\partial^2 w}{\partial t^2}$$

This beam was assumed to be uniform, meaning the elastic properties were assumed to be homogeneous for the length of the beam.

**1.2.4. Plate.** As the membrane's equation of motion was the two-dimensional version of the string equation, the plate equation is the extension of the beam equation to three dimensions. Figure 1.8 shows a rectangular section of plate.

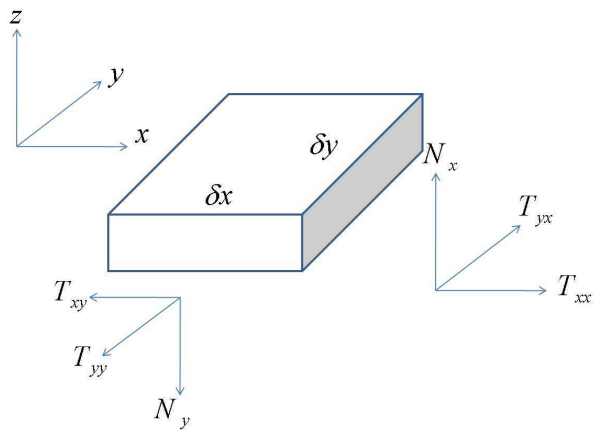


FIGURE 1.8. The forces acting on a section of a plate

An equation for the transverse motion of the plate will be constructed. Looking at Figure 1.8, the transverse direction is the $z$ direction. The $xy$ plane forces can be described by a tensor:

$$\underline{\underline{\mathbf{T}}} = \begin{bmatrix} T_{xx} & T_{xy} \\ T_{yx} & T_{yy} \end{bmatrix}$$

The shear forces in the $z$ direction along the $x$ and $y$ planes will be denoted as $\mathbf{N} = (N_x, N_y)$. The section of plate is assumed to be in equilibrium in the $x$ and $y$ direction, thus by balance of moments, $T_{xy} = T_{yx}$. Let $w(x, y, t)$ be the displacement in the $z$ direction. These displacements are assumed to be small, which will allow the use of approximations (based on trigonometric relationships) to express the tension forces in terms of $w(x, y, t)$. The equation of motion in the $z$ direction is:

$$\frac{\partial}{\partial x}(T_{xx}\frac{\partial w}{\partial x} + T_{xy}\frac{\partial w}{\partial y} + N_x) + \frac{\partial}{\partial y}(T_{yx}\frac{\partial w}{\partial x} + T_{yy}\frac{\partial w}{\partial y} + N_y) = \lambda\frac{\partial^2 w}{\partial t^2}$$

where $\lambda$ is the mass per unit area. The first section on the left hand side represents the net $z$ direction forces over the $x$ variable and the second section is the same for the $y$ variable. The right hand side is the usual inertial component. The shear forces, can be written in terms of moments as:

$$\frac{\partial M_{xx}}{\partial x} + \frac{\partial M_{xy}}{\partial y} + N_y = 0 \quad \frac{\partial M_{yx}}{\partial x} + \frac{\partial M_{yy}}{\partial y} - N_x = 0$$

where the first index on $M$ represents the component of the moment vector, and the second index indicates the direction of the normal vector of the plane. The moments can then be written in terms of the curvature (second derivatives of $w$). The final result is:

$$T_{xx}\frac{\partial^2 w}{\partial x^2} + 2T_{xy}\frac{\partial^2 w}{\partial x \partial y} + T_{yy}\frac{\partial^2 w}{\partial y^2} - D\nabla^4 w = \lambda \frac{\partial^2 w}{\partial t^2}$$

where $D$ is the bending stiffness of the plate (involving the Young's modulus and moment of inertia). In the simple case of an isotropic tension, the equation can be reduced to:

$$T\nabla^2 w - D\nabla^4 w = \lambda w_{tt}$$

Given boundary conditions, this problem can be solved easily for the transverse motion of the entire plate [**36**].

### 1.3. Composite Materials

The materials described thus far have been homogeneous in composition and structure. The mechanical behavior of the string, membrane, beam, and plate is described by constant material properties. Heterogeneous materials, on the other hand, do not have constant material properties. They are either composed of multiple materials, each exhibiting different mechanical behavior, or they have inhomogeneous structures which cause their meterial parameters to vary in space and/or time. Composite materials make up one simple subclass of heterogeneous media. They are composed of two or more materials with different mechanical properties, arranged in alternating layers. Examples of composites include steel, reinforced concrete, fiberglass, and Kevlar. The purpose of

creating such materials is often to capitalize on the combination of beneficial qualities of each component. The alternating layer patterns of composites can be found at a scale well above the molecular level. In comparison, the heterogeneity of biomaterials is much more complex. Their structures can be inhomogeneous down to the molecular scale, and often do not have well-defined patterns.

The inhomogeneities in composites and biomaterials both need to be taken into account to produce accurate models. However, the methods utilized will be different. The periodicity of a composite's heterogeneous structure allows for the successful application of homogenization methods [**53, 56**]. This is not the case with biomaterials. However, understanding the mechanics of a simple composite is a step in the right direction towards the continuum-microscopic approaches that will be utilized in modeling media like the cell cytoskeleton.

As an elementary example, consider a one-dimensional string that is composed of $2N$ alternating equal length segments of two different types of material. The first material has a Young's modulus of $E_1$ while the second has a very different Young's modulus of $E_2$. Let the string be of length $L$. Then the elasticity modulus is given by:

$$E(x) = \begin{cases} E_1 & n\frac{L}{N} < x < (n+\frac{1}{2})\frac{L}{N} \\ E_2 & (n+\frac{1}{2})\frac{L}{N} < x < (n+1)\frac{L}{N} \end{cases}$$

where $n = 0, 1, ...N-1$.

A linear stress-strain relationship for this situation would be:

$$\sigma = E(x)\epsilon = E(x)\frac{\partial u}{\partial x}$$

where $u(x)$ is the longitudinal displacement. Rewriting the stress as a tension force over area gives:

$$\frac{T}{A} = E(x)\frac{\partial u}{\partial x} \tag{1.10}$$

For a homogeneous material, $E$ is a constant, so $u(x)$ can be found by simple integration:

$$u(x) = \frac{T}{AE}x$$

However, if $E$ is spatially varying as in a composite material, then the solution becomes piecewise as:

$$u(x) = \frac{T}{A}\begin{cases} \frac{x}{E_1} + \frac{nL}{2N}(\frac{1}{E_2} - \frac{1}{E_1}) & n\frac{L}{N} < x < (n+\frac{1}{2})\frac{L}{N} \\ \frac{x}{E_2} + \frac{(n+1)L}{2N}(\frac{1}{E_1} - \frac{1}{E_2}) & (n+\frac{1}{2})\frac{L}{N} < x < (n+1)\frac{L}{N} \end{cases}$$

These functions come from the integral of $1/E(x)$ where $E(x)$ looks like Figure 1.9.

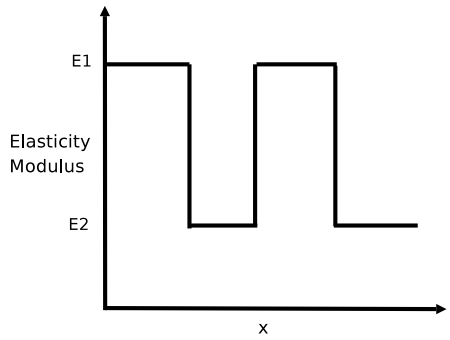


FIGURE 1.9. The graph of the elasticity modulus of a string with alternating segments of two types of materials.

Homogenization is a general technique that can be easily applied to this simple example to average the integral of $\frac{1}{E(x)}$ that arises when solving 1.10.

$$\begin{aligned} \overline{(1/E)} &= \frac{N}{L}\int_0^{L/N} \frac{dx}{E(x)} \\ \overline{(1/E)} &= \frac{N}{L}(\frac{1}{E_1} + \frac{1}{E_2})\frac{L}{2N} \\ \overline{(1/E)} &= \frac{1}{2}(\frac{1}{E_1} + \frac{1}{E_2}) \end{aligned}$$

The function:

$$u(x) = \frac{Tx}{A}\overline{1/E} + c$$

is a good approximation to the exact solution on the scale of the whole string. This basic example was used to introduce one of the fundamental issues in multiscale modeling, namely how to incorporate the varying mechanical properties of an inhomogeneous material into a continuum-level description of the medium. For this simple case, homogenization is sufficient. For materials with more complex space and time-varying properties, a different class of algorithms known as continuum-microscopic methods is often used. An introduction to such methods and a description of how this thesis work contributes to the further development of these algorithms is presented in the forthcoming sections.

## 1.4. Continuum-Microscopic Models

Continuum-microscopic (CM) methods incorporate the modeling of two or more scales (length, time, or both) into one algorithm to model a particular phenomenon. The different scales are usually governed by different physical laws [**25**]. The general idea behind continuum-microscopic modeling is to utilize information obtained from a more detailed description of the material to update or predict information at a less detailed level of description, where presumeably computations can be done more efficiently. CM models have thus far been used under the assumption that the microscopic configuration is known or can be reasonably approximated by a known distribution function (such as a Gaussian or unifrom distribution) [**25, 28, 43**]. However there are many problems for which the microscopic structure is unknown because it is changing over time and highly heterogeneous. This thesis work seeks to extend a basic CM model to address such situations.

CM models have been applied to a wide range of scientific subjects including fluid dynamics [**9, 49, 68**], biology [**12, 26**], chemistry [**8, 31**], and material sciences [**45, 79**]. The main CM methods that have been applied in these various fields include the Heterogeneous Multiscale Method (HMM) [**24**], the Adaptive Mesh and Algorithm Refinement (AMAR) method [**28**] and the Equation-Free Method [**43**]. A brief summary of each method will now be presented.

**1.4.1. The Heterogeneous Multiscale Method.** The Heterogeneous Multiscale Method (HMM) was developed by E and Engquist [**25**]. Usually one is interested in modeling a continuum level situation such as fluid flow or deformation of an object. Sometimes one only has data from the discrete microscopic elements that make up the medium. A model that utilizes this vast quantity of detailed information to evolve the entire macroscopic domain is not computationally feasible [**25**]. The main idea behind HMM is to have two different numerical schemes: one at the continuum level (ex. finite volume with Navier Stokes) and one at the microscopic level (ex. Newton's molecular dynamics). Suppose a grid is laid over the macroscopic domain. To update the continuum variables of each macroscopic grid cell requires the computation of fluxes at each interface between adjacent cells. At these interfaces is where the microscopic scheme is employed to produce more accurate flux information. As an example, let $U$ be a macroscopic variable (such as velocity), and $u$ is the corresponding variable at the microscopic scale. The two variables are related by an averaging operator $Q$ where $Qu = U$. Consider the scalar conservation law that would be used to update the $u$ variable at the microscale:

$$u_t + f(u)_x = 0 \tag{1.11}$$

This equation may be solved using, for example, a Riemann solver. The updated variable $\bar{u}$ is then used to compute the average flux $F$ at the macroscopic scale:

$$F_{j+1/2} = \frac{\int_{t^n}^{t^{n+1}} f_{j+1/2}(\bar{u}, s)ds}{\Delta t}$$

where $j + 1/2$ indexes the macroscopic grid cell interfaces and the values $j$ index the grid cell centers. Then the new macroscopic $U$ variable is found via the following update scheme:

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}(F_{j+1/2} - F_{j-1/2})$$

Due to the smaller spatial scale at the microscopic level, smaller time steps must be used in the solution of 1.11 to get stable results. This is the main reason why full microscopic

simulations are computationally intractable. HMM is often applied to problems where there are two time-scales: A slow continuum level time scale where variables change perhaps on the order of seconds and a fast time scale at the microscopic level where molecules quickly reach a quasi-stationary state, meaning the microscopic variables settle to particular values [**25**]. Therefore the microscopic equations do not need to be solved for the entire macroscopic time step. This separation of scales is what makes HMM possible and useful. The basic steps in HMM can be summarized as:

(1) Create a microscopic instantiation from the macroscopic variable information for the interface regions

(2) Run the microscopic updating scheme (ex. molecular dynamics equations) until the microscopic data has reached a quasi-stationary state

(3) Apply an averaging or compression algorithm to produce macroscopic level flux information for the continuum equation

(4) Update the macroscopic variables

This method works well for systems where thermodynamic equilibrium can be assumed within each macroscopic grid cell. HMM has been applied in many scientific areas such as gas kinetics [**81**], fluids [**68**] and elasticity [**1**].

**1.4.2. Adaptive Mesh and Algorithm Refinement.** The Adaptive Mesh and Algorithm Refinement (AMAR) technique was developed by Garcia et al. [**28**]. This method combines the ideas of grid refinement with the utilization of different equations at the different refinement levels. As a general example, suppose that the goal is to model the flow of cytoplasm in a motile cell. Typically to solve such a problem numerically one would lay a grid over the fluid domain and solve the Navier-Stokes equations for the hydrodynamic variables. Within this fluid grid there may be regions where interesting dynamics is taking place, perhaps near the opening of a protrusion or at the boundary of the cell. In thees regions one may want to use adaptive mesh refinement to obtain a better resolution of the solution. If the refinement changes the spatial scale by several orders of magnitude, the fluid may no longer be viewed as a continuum but rather

as individual fluid particles. The AMAR method proposes employing a different set of equations at this fine grid scale to reflect this new problem description.

The basic idea of adaptive mesh refinement is simply to begin with a coarse grid and divide this grid into smaller grid cells using a specified refinement ratio. The refinement should proceed until the error in the solution has been reduced to below a given threshold. There can be different depths of refinement in different regions of the domain. The variables in the boundary cells that border the interfaces between regions of differing refinement need to match up. This synchronization is done using averaging and interpolation techniques. The fine grid is initialized by interpolating the coarse grid information. Once the fine grid has been advanced, the boundary values are averaged to update the interface boundary with the coarse grid. Also, a flux correction algorithm is applied at the coarse/fine interface. These adjustments help to maintain conservation of variables such as mass, momentum and energy.

AMAR adds another layer of complexity to the normal AMR algorithm. It has to match up variables that come from two different physical laws. One cycle of AMAR begins with the computation of fluxes for all coarse grid cells. These fluxes are used to advance the variables forward in time one continuum step $\Delta t_{cont}$. This is done even for the coarse cells that overlay the refined cells. Once the continuum step has been taken, the particle dynamics equations within the refined region are advanced one small time step $\Delta t_{particle}$ at a time until they have evolved to the same point in time as the coarse grid. In the coarse grid cells directly surrounding the refined region, a "buffer" region of microscopic particles is created. They are instantiated using distribution functions (like Maxwell-Boltzmann) parameterzed by the continuum level variables. These particles are moved with the particles in the refined area during each $\Delta t_{particle}$. If a particle crosses from the refined region to a coarse cell or vice versa, then a contribution to the flux at that interface is recorded. After all microscopic time steps have taken place, the overlying coarse grid is updated using averaged quantities from the refined grid. Also the fluxes

at the interfaces between coarse and refined regions are corrected utilizing the fluxes obtained during the particle level advancement.

In order to maintain numerical stability, the finer grid region may need to take many small time steps to equal one large time step of the coarser grid. In AMAR, the hope is that the region of the domain where particle dynamics needs to be applied is small so that the number of computations can be kept to a minimum.

Garcia et al. demonstrated this technique with several examples such as the movement of a shock wave generated by a piston, the motion of a gas sheared by a moving wall and fluid flow past a sphere [**28**]. In the piston example the region around the shock wave is refined to the gas particle level, and this fine grid moves with the shock wave. The Navier-Stokes equation is used to solve for the overall air flow in the tube, while the Direct Simulation Monte Carlo method is employed for the particles near the shock wave. This method captures the shock wave better then the purely continuum level Navier-Stokes equation version.

**1.4.3. The Equation-Free Method.** The Equation-Free Method (EFM) was introduced by Kevrekidis et al. [**44**] and has been utilized in a wide variety of applications. It is similar to HMM in that its goal is to solve a continuum level problem by using microscopic scale information to improve accuracy. As in HMM, EFM performs a small number of microscopic level time step updates to garner information to be used at the continuum level. The difference between the two techniques is that in EFM the macroscopic level equations are never explicitly advanced as they are in HMM. The microscopic equation solutions are used to predict what the continuum variables will be at the next macroscopic time step. Once the prediction has been made, a new microscopic instantiation is determined from the macroscopic variables and the method is repeated. The transition from the macroscopic level to the microscopic level is done via a "lifting" operator and the opposite transmission of information is done by a "restriction" operator. The lifting operator consists of using the first few statistical moments of the continuum variables

to find a distribution function to instantiate the microscopic variables. The restriction operator is usually an averaging algorithm.

The reasoning behind EFM stems from the difficulty in determining constitutive laws at the macroscopic level. The physical laws for the motion of the microscopic level description of the problem are usually well-known (ex. molecular dynamics or kinetic theory). However, for many practical applications, the interest lies in what is happening at a much coarser level. The microscopic laws are used to try to establish constitutive laws at the macroscopic level, but this task is not always simple. EFM circumvents this issue by not requiring explicit continuum level constitutive relations. The basic steps in EFM are:

(1) Start with initial conditions for the macroscopic variables
(2) Use a lifting operator to instantiate microscopic variables
(3) Run the computation at the microscopic level for a short time period
(4) Use the restriction operator to transfer the variables to the macroscopic scale
(5) Average the information over time to determine an estimate for the future macroscopic variables.

As mentioned previously, EFM has been applied to numerous subjects including population dynamics [**12**], disease evolution [**21**], peptide folding [**38**] and chemical reactions [**55**].

### 1.5. Extending the Continuum-Microscopic Idea

The common drawback among the continuum-microscopic methods presented above is that their utility has thus far been demonstrated for examples where the probability distribution function (PDF) of the microscopic data has an assumed shape. However if a system has a highly heterogeneous micro-structure or experiences many changes and fluctuations over time, its microscopic data is unlikely to consistently conform to one particular shape. The cell cytoskeleton falls into this category due to its complex

structure of interconnected filaments that changes over time in response to mechanical stresses and chemical reactions.

What is needed is a computationally efficient procedure that models continuum-level constitutive relations utilizing microscopic data from configurations that were instantiated based on past microscopic data. This is precisely the goal of the following thesis work. This research intertwines several topics including the investigation of: (1) computationally efficient methods of PDF estimation; (2) the evolution of the microscopic variable distribution functions forward in time in order to perform accurate instantiations at later time steps; (3) the incorporation of continuum level constraints during microscopic instantiation to insure model consistency. These are the research topics to be discussed in this thesis work, with specific application to cytoskeletal mechanics.

CHAPTER 2

# The Cytoskeleton: Biology and Models

In this chapter background information on the biology of the cytoskeleton and its role in cellular protrusive activities will be provided. This will be followed by a quantitative description of the difficulties encountered in trying to model such a complex structure. Next, a review of various computational algorithms that have been utilized to model the cytoskeleton will be presented. The chapter will conclude with an outline of the modeling approach developed in this thesis, and how this new algorithm adds to the current research in the field.

## 2.1. The Structure of the Cytoskeleton

A typical animal cell is approximately 10 $\mu m$ in diameter [**2**], and consists of organelles (such as the nucleus, mitochondria) suspended in a fluid cytosol surrounded by the cytoskeleton. All of these cellular components are encased by a thin plasma membrane (Figure 2.1).

The cytoskeleton of the animal cell is a complex structure that gives the cell mechanical support and integrity [**2, 59**]. This dynamic newtwork of intertwined filaments participates in and orchestrates many cellular activities such as cell migration, mitosis, apoptosis and mechanotransduction [**59**].

The protein polymers that comprise the cytoskeleton include actin filaments, microtubules and intermediate filaments [**2, 10, 59**], and these fibers are crosslinked to one another by proteins such as filamin and $\alpha$-actinin [**2**]. The main types of filaments participating in protrusive activities such as blebbing and lamellipodium formation are actin polymers, and their properties will now be describe in further detail.

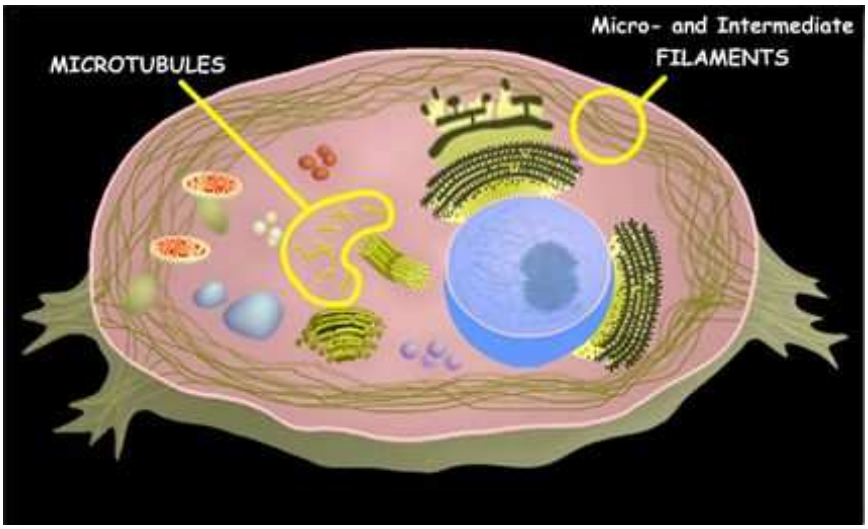


FIGURE 2.1. Cartoon of a typical eukaryotic cell from *www.abcam.com*

**2.1.1. The Actin Filament.** Actin filaments are long polymer chains built from actin protein subunits. These subunits are approximately 5 nm in diameter [**2**]. Free monomers of actin carry a molecule of ATP and are known as G-actin or globular-actin (see Figure 2.2).

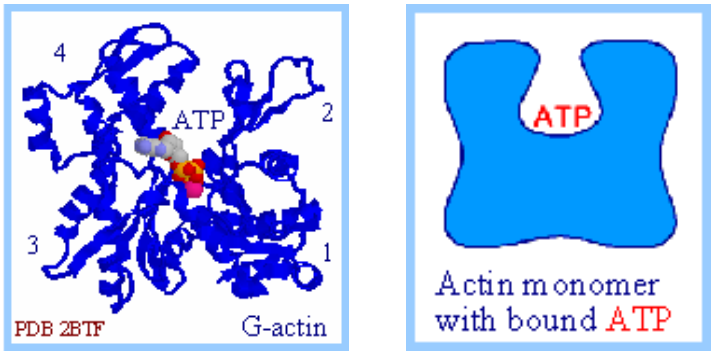


FIGURE 2.2. (A) Ribbon diagram of the actin monomer and (B) cartoon image of the actin monomer with ATP molecule *http* : *//www.rpi.edu/dept/bcbp*

When a G-actin subunit joins a growing polymer chain the ATP molecule is hydrolyzed into ADP and the subunit is attached. The actin protein in filament form is known as F-actin (filamentous-actin). Actin filaments have different rates of growth and shrinkage at their two ends. The "plus" end has a faster rate of elongation and shortening than the "minus" end [**2**].

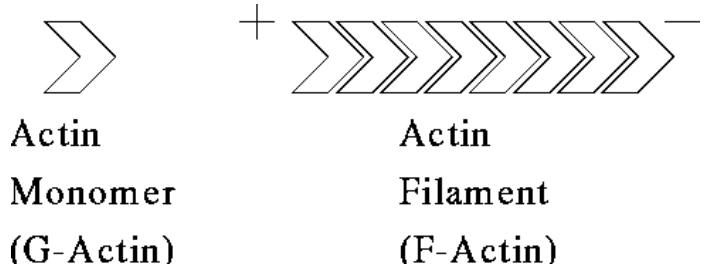


FIGURE 2.3. Actin monomers typically join onto the plus end of the growing chain, *http* : *//www.iam.ubc.ca/ spiros*

The subunits in a filament are held together by weak, noncovalent bonds that can be broken by thermal fluctuations [**2**]. Because of this, it is very easy for monomers to add on and break off at both ends of the chain. This dynamic instability allows the polymers to disassemble easily into monomer form to diffuse across the cell, and reconstruct themselves in a new position [**2**]. However, a chain that breaks easily is not very useful to the cell for doing mechanical work. To circumvent this issue, two actin subunit chains are often bound together in parallel to form a stronger double-stranded helical structure (see Figure 2.4. Subunits can still add on and fall off the ends, but it is much more difficult for a filament to break spontaneously somewhere in the middle with this type of structure [**2**].

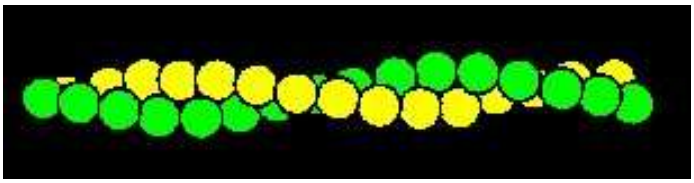

FIGURE 2.4. A doubled stranded, helical actin filament composed of linked G-actin monomers, *http* : *//www.cryst.bbk.ac.uk/PPS*2

Filament length can vary depending on cell type, but they generally are 1-20 $\mu$m long and about 8 nm wide [**10, 41**]. They can be as long as 50-100 $\mu$m in muscle cells [**40**], and as short as 0.2-0.35 $\mu$m in cytoskeleton meshes [**15**]. In either case, they are several orders of magnitude longer than they are wide.

Actin filaments are classified as semi-flexible polymers [**41**]. A single actin filament can withstand an elongation force of about 110-250 pN before breaking, and it only

stretches about 0.2-0.3% under these forces [**41**]. It has a stiffness of approximately 45-65 pN/nm for actin filaments of length 1 $\mu$m [**41**]. On average the Young's modulus of an actin polymer is 0.5 to 2 $\times$ $10^9 N/m^2$ [**10, 41**]. In comparison to stretching, actin filaments bend quite easily. Their flexural rigidity has been found to be on the order of $10^{-26} Nm^2$, based on a persistence length of 10-20 $\mu$m [**10**]. This large difference in magnitude between the stretching and bending properties of actin filaments allows them to be classified as an elastic string for modeling purposes.

**2.1.2. Construction of Filament Networks.** Actin filaments typically group together to form larger structures such as bundles and meshes. Microvilli and filopodia (to be described in depth later) are examples of actin bundle structures (see Figure 2.12). A typical microvillus is composed of 20 to 30 actin filaments tightly bound together in parallel, with a diameter of 0.08 microns and an average length of 1 micron [**2**]. The protein $\alpha$-actinin is one of the main proteins that connects actin filaments together in parallel in these bundled structures [**59**].

The cytoskeleton is an example of a mesh-like actin structure. The actin filaments are attached together in a woven pattern by proteins such as spectrin and filamin. Spectrin is a long 100 nm, flexible protein found close to the cell surface in red blood cells [**2**]. Two molecules link together head to head to create two actin filament binding sites that are spaced approximately 75-200 nm apart depending if the spectrin polymer is in a convoluted position or stretched out straight [**2, 10**]. This distance is quite large compared to the other proteins which bind actin bundles in tight configurations about 14-30 nm apart. There are approximately 1.2 to 2 $\times$ $10^5$ spectrin molecules in one red blood cell [**2, 10**]. Spectrin has an estimated spring constant of 2 $\times$ $10^{-6} J/m^2$ which means spectrin is elastic enough to allow a red blood cell to adjust its shape in order to squeeze through a narrow capillary [**10**]. Filamin is another binding protein found in other types of cells. It crosslinks two filaments together almost at right-angles to one another forming a loose grid of actin polymers [**2, 59**] (Figure 2.5).

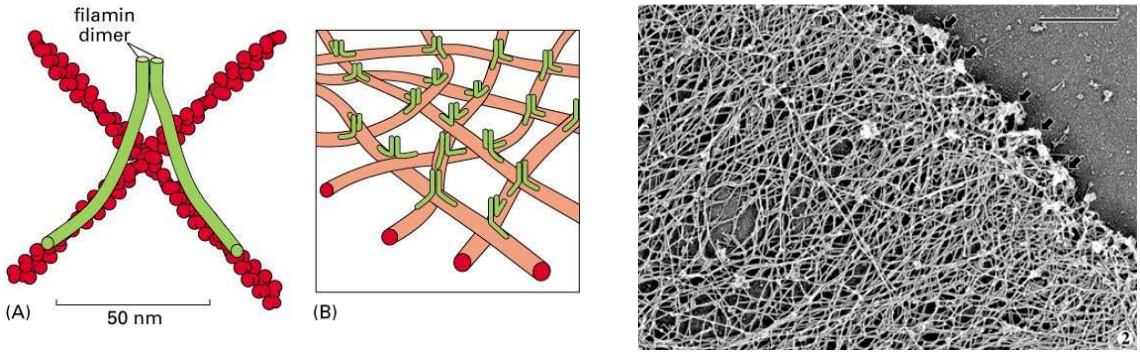


FIGURE 2.5. (A) Cartoon showing how the filamin protein crosslinks two actin filaments, and how it forms an actin mesh, [**2**] and (B) an electron micrograph of a typical actin network near the plasma membrane, *http* : *//scienceblogs.com/*

There are other proteins which keep the actin mesh attached to the plasma membrane [**71**]. In platelets, filamin serves the dual purpose of linking actin filaments to other actin filaments and linking the actin mesh to the plasma membrane. In red blood cells, a protein in the plasma membrane known as band 3 attaches to another protein called ankyrin which in turn attaches to the spectrin proteins on the cytoskeleton [**2**] (see Figure 2.6).

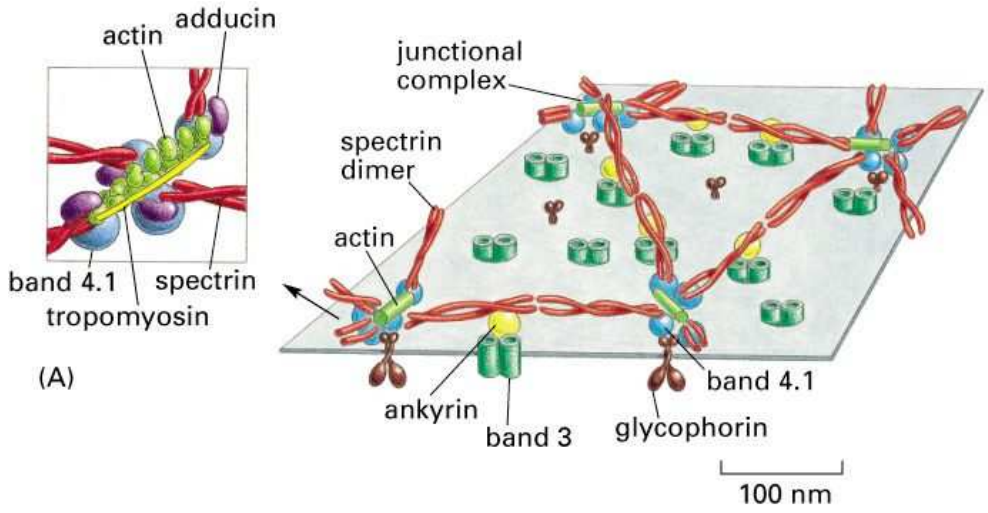


FIGURE 2.6. Diagrams depicting how spectrin crosslinks actin filaments together, and also how it attaches to the plasma membrane

Dystrophin is a protein found in muscle cells that attaches the cytoskeleton to the outer membrane. Other adhesive proteins include ezrin, radixin, and moesin [**2**]. Figure 2.7 shows several examples of membrane-cytoskeleton adhesion via different proteins.

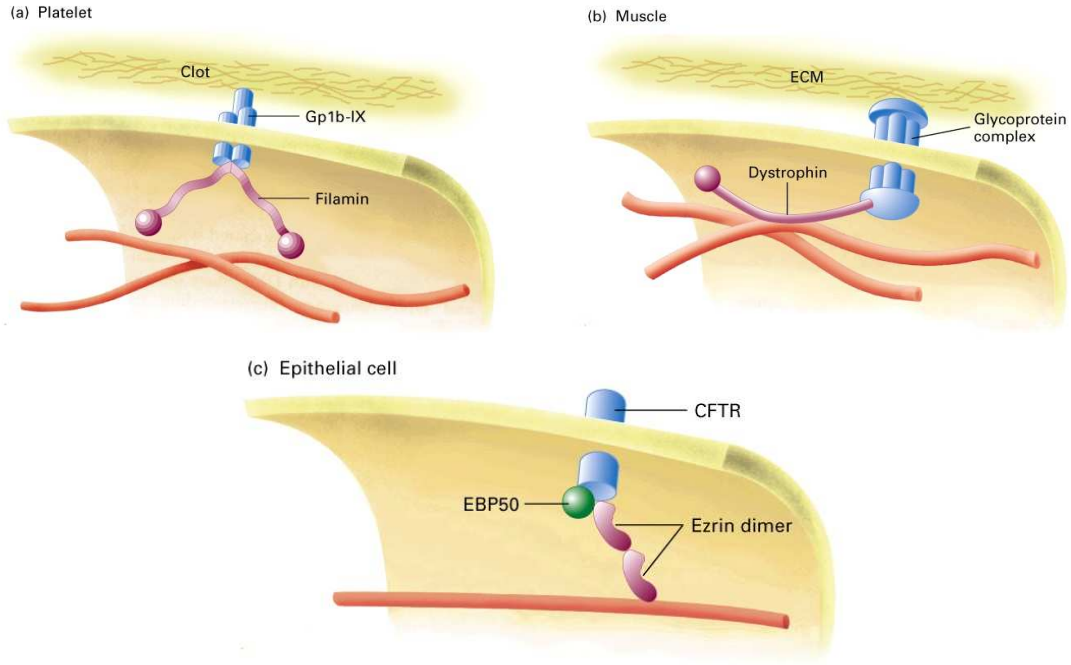


FIGURE 2.7. Diagrams of how various proteins attach the cytoskeleton to the plasma membrane in a (A) platelet, (B) muscle cell and (C) an epithelial cell. Images from $http://bioweb.wku.edu$

The total number of actin filaments within a cell varies by cell type and concentration levels of actin. In red blood cells, actin fibers form a one to two filament thick network of short filaments [**71**]. This amounts to approximately 120,000-300,000 short actin filaments in a red blood cell cytoskeleton. Boal estimates that cells with high actin densities of $5mg/ml$ or more, have approximately $1.9 \times 10^{20} filaments/m^3$ of filaments of length 1 $\mu$m [**10**]. This translates to about 200,000, 1 $\mu m$ filaments in a 10 $\mu m$ diameter animal cell.

**2.1.3. Network Properties and Behavior.** The cytoskeleton is typically between 5 nm - 2 microns thick [**18, 71**]. The size of the gaps in the actin mesh range from 10 nm - 100 nm [**19, 70, 73**], depending on cell type. Individual actin filaments have certain material properties as described above. However, when these filaments become

crosslinked in a gel, these values can change. For instance, the estimated Young's modulus for 1 mg/ml of crosslinked F-actin is $100,000 dyn/cm^2$ which is 10 kPa and the shear modulus is approximated at $1000 dyn/cm^2$ or 100 Pa [**41**]. Charras et al. estimated the elastic modulus of the actin cortex in filamin-depleted melanoma cell line to be 1-3 kPa [**19**]. In general, the Young's modulus for the actin network is lower than the individual actin filaments. This is due to the fact that crosslinking proteins such as spectrin are more elastic than actin, so they make the overall mesh less stiff. In Boal's textbook, he estimates the shear modulus of the actin networks of several types of cells: red blood cell, $6 - 9 \times 10^{-6} J/m^2$ (determined with pipette aspiration); auditory outer hair cells, $1.5 \times 10^{-2} J/m^2$ (pipette aspiration); fibroblasts, $2 - 4 \times 10^{-3} J/m^2$ (calculated with magnetic field applied to magnetic beads attached to the membrane) [**10**].

An interesting feature of actin filaments and consequently of the cytoskeletal network is that it can be contracted by myosin II (a motor protein) to produce a force [**2, 42, 57, 66**]. This force generation plays a central role in the cell's protrusive and locomotive activities. Myosin II, like actin, is found in all eukaryotic cells [**2**]. Myosin II is a long protein composed of two heavy chains and two light chains. Near the end of the two heavy chains is a "head" region from which forces can be generated [**2**]. Myosin II subunits join to form a filament by bundling their tails together. This creates a bipolar filament with myosin heads facing in opposite directions along the fiber (see Figure 2.8). This configuration is ideal for pulling actin filaments together [**2**]. It is often an influx of calcium ions that biochemically triggers myosin to begin the contraction process [**2, 62**].

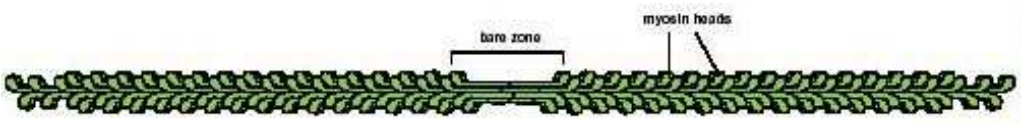


FIGURE 2.8. A myosin thick filament, with the myosin heads facing opposite directions on the fiber, [**2**]

One cycle of force-generation can be described as follows. Each myosin head attaches to the actin filament at a binding site. A molecule of ATP attaches to the myosin head causing the release of the head from the actin filament. The ATP becomes tightly

bound to the head region and this binding causes the 8.5 nm "lever arm" [**2**] to travel toward the plus end of the actin filament and weakly reattach at a new binding site approximately 4-11 nm further up the filament [**75**]. During this traversal, the ATP molecule is hydrolyzed into ADP and an inorganic phosphate. This phosphate is then released, which produces the force-generating "power stroke" that pulls the myosin head and the actin filament back to a normal starting position. During this last stage, the ADP molecule is also released, readying the myosin head for another cycle. Figure 2.9 shows the stages of the actin-myosin force generation cycle. This whole process can be done at rates ranging from 0.2 - 60 $\mu$m/s [**2**]). One myosin head can produce a force ranging from 0.8-8 pN [**10, 19, 75**]. There are several hundred myosin heads on a myosin filament, with approximately 16-20% of the heads working on an actin filament at one time [**75**]. Using this method, myosin filaments can slide actin filaments past each other to produce a contraction of the actin mesh.

## 2.2. Cellular Protrusions

As mentioned at the beginning of this chapter, the cytoskeleton plays a central role in cell migration and cellular protrusive activities. It carries out these tasks by breaking down, rearranging and rebuilding itself as needed. The types of protrusions that can be produced by cytoskeletal reconfigurations include lamellipodia, microvilli and blebs, and they will now be described.

**2.2.1. Lamellipodia.** A lamellipodium is a two-dimensional actin network projection that forms in at the periphery of a motile cell to help the cell crawl over a solid substrate. It is a flat, fan-like protrusion that pushes the cell forward via actin polymerization [**2**]. Lamellipodia are found primarily in epithelial cells and fibroblasts, but also in some neurons. The actin filaments in a lamellipodium are arranged in a very organized way (as opposed to the more randomly entangled cytoskeleton). The filaments form a tree-like structure, with polymers branching off from one another at 70° angles. Growth of this webbed structure needs to happen quickly in order for it to push the cell forward at

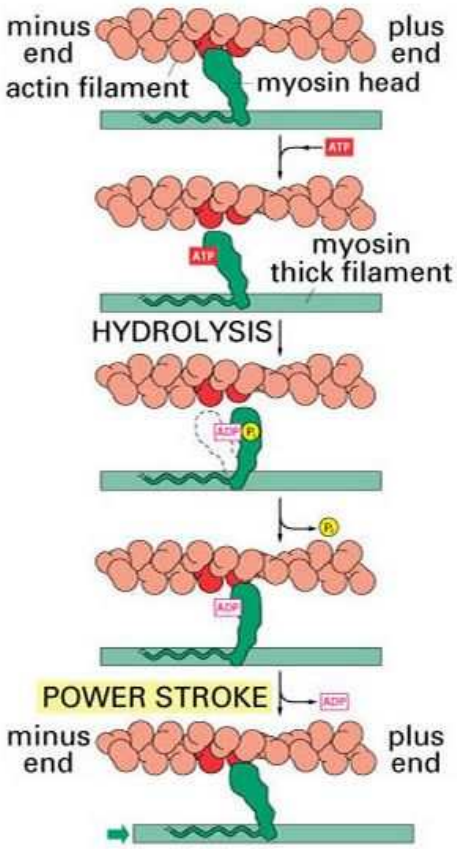


FIGURE 2.9. The actin-myosin force generation cycle [**2**].

a reasonable pace. (For example, keratocytes can crawl at top speeds of 30 $\mu m$/minute [**2**].) To facilitate nucleation (the formation of a new actin filament) and polymerization (elongation of an existing actin filament), the cell utilizes proteins known as actin-related proteins (ARPs). The ARP 2/3 complex attaches to the minus end of an actin subunit, creating a base off of which other actin monomers quickly latch onto forming a growing filament. ARP 2/3 can also attach itself to the middle of existing filaments creating a branch point off of which a new filament segment can grow. (See Figure 2.10 for a diagram).

The thin lamellipodium protrusion needs strong mechanical support behind it in order to push the cell forward. This strength comes from the cell's cytoskeleton whose crosslinked structure provides the necessary mechanical stiffness.

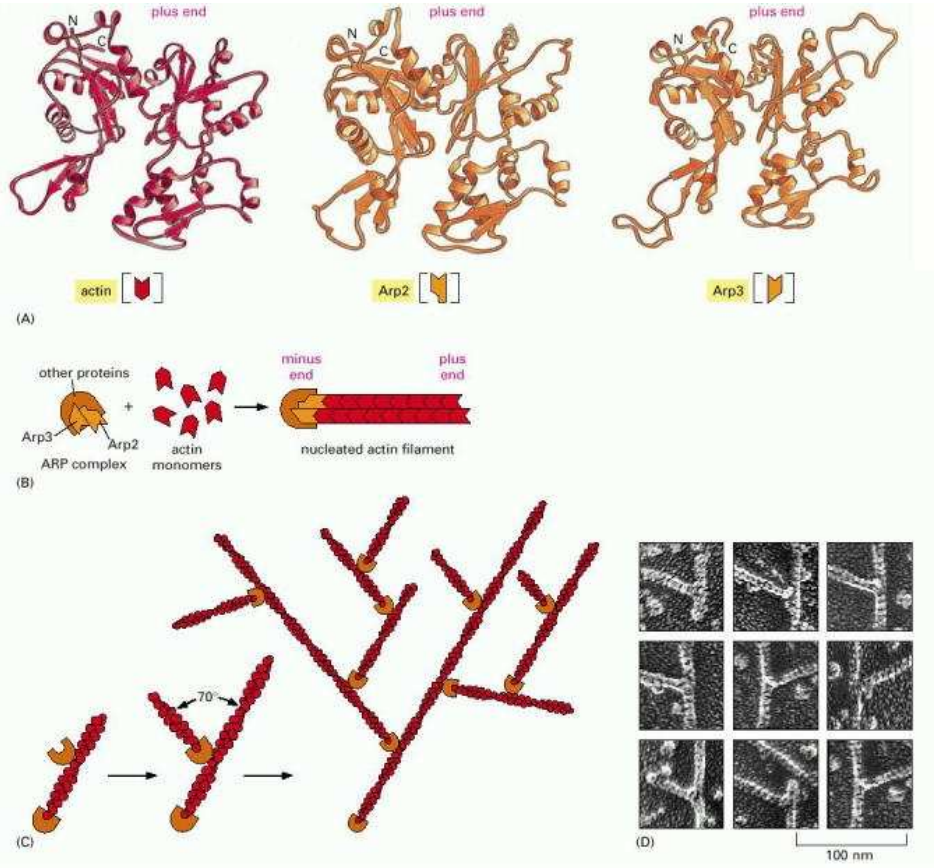


FIGURE 2.10. Image from Alberts', "Molecular Biology of the Cell", depicting (A) the protein structure of actin, ARP 2 and ARP 3, (B) the nucleation process, (C) how a 2D actin web is created, and (D) electron micrographs of the actin branch points [**2**]

How is a cell able to crawl across a solid substrate utilizing a lamellipodium? The whole process is accomplished in several coordinated stages using different actin structures and cell functions. The first stage is the formation of the lamellipodium at the front of the cell (in the direction the cell plans to crawl). The actin filaments in the tree structure are oriented with their plus ends facing forward, and their minus ends facing the interior of the cell. The lamellipodium pushes the cell membrane forward by treadmilling, meaning that actin filaments grow at the front of the protrusion and depolymerize at their other ends. Once the lamellipodium has pushed the membrane forward a short distance, the cell forms focal contacts (attachments) with the solid substrate at the protrusion's leading edge. These attachments provide traction for when the back end of the cell is moved forward to catch up with the front. This task is accomplished by

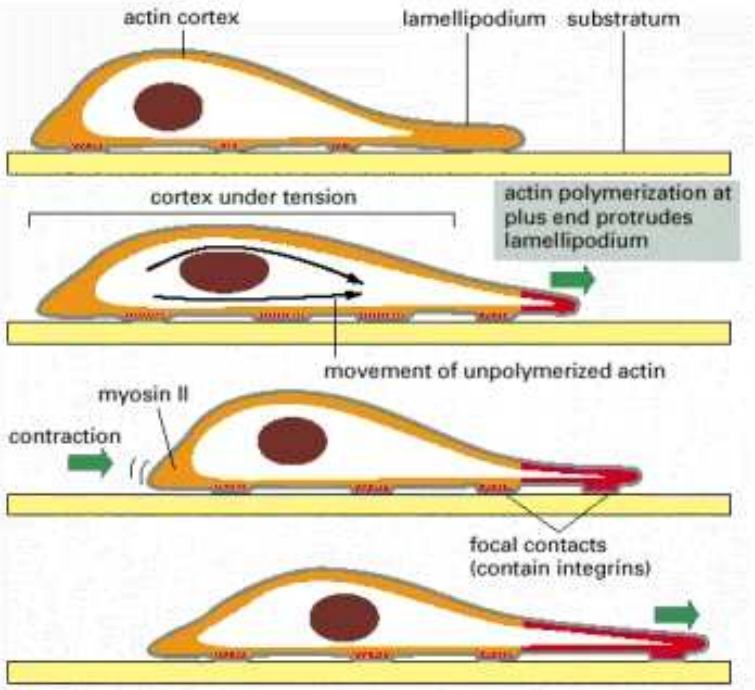


FIGURE 2.11. Image from Alberts', "Molecular Biology of the Cell", showing the stages of one cycle of cell crawling [**2**]

myosin II fibers which contract the actin mesh at the rear of the cell pulling it forward. Figure 2.11 shows the stages in one full crawling step.

**2.2.2. Microvilli.** Microvilli are very different from lamellipodia in structure and purpose. They are thin, finger-like protrusions found most commonly in epithelial cells. A typical intestinal epithelial cell can have thousands of microvilli whose purpose is to increase the surface area of their host cell to facilitate the absorption of nutrients from food passing through the small intestine [**2**]. A microvillus is composed of 20-30 actin filaments bundled together tightly in a parallel structure (see Figure 2.12). They have a diameter of 0.08 microns and an average length of 1 micron [**2**]. The filaments are held together by two crosslinking proteins: villin and fimbrin. These proteins each have two actin filament binding sites that are very close together (8-10 nm apart [**2**]) that facilitate tight bundling. The actin structure is attached to the plasma membrane by adhesive myosin I protein sidearms. Like in lamellipodia, the actin filaments in microvilli are oriented with

their plus ends outward, so polymerization takes places at their protruding end. These actin bundles are anchored in and grow out of the cell's cytoskeleton.

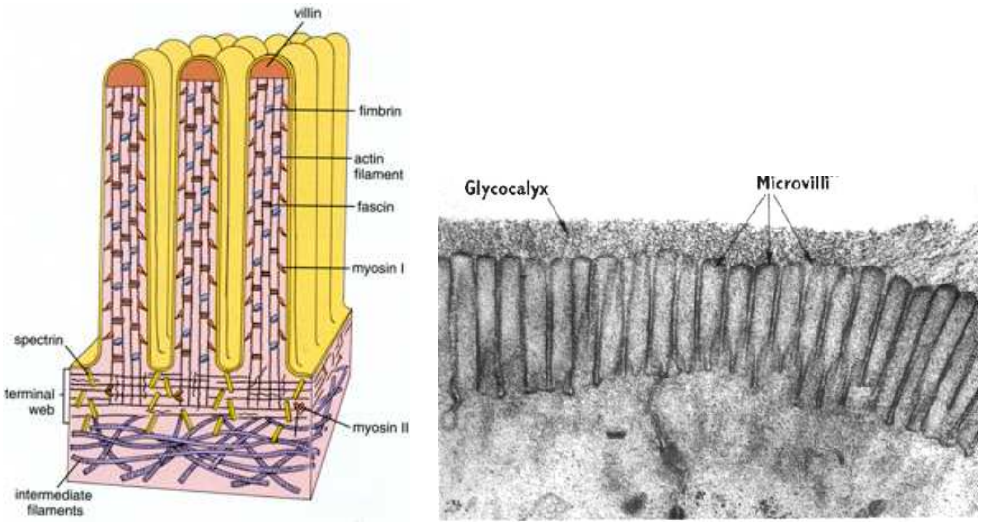


FIGURE 2.12. (A) Cartoon of the structure of microvilli, *http* : *//anatomy.iupui.edu* and (B) Microvilli in an intestinal epithelial cell *http* : *//www.cytochemistry.net/Cell − biology*

**2.2.3. Blebs.** A bleb is a balloon-like, cytosol-filled protrusion of the plasma membrane. Unlike lamellipodia and microvilli, this type of protrusion is not formed by active growth and rearrangement of the cytoskeleton [**19, 22**]. However, the onset of bleb formation is triggered by a contraction of the actin network and the retraction of a bleb back into the cell body is completely driven by cytoskeletal actions [**18**].

The driving force behind bleb formation, is not actin polymerization, but rather fluid pressure [**3, 19, 22**]. The cell's cytosol is typically at a higher pressure (20-300 Pa higher [**17, 67**]) as compared to the external ambient fluid. The membrane is prevented from moving outward in normal circumstances by its connections with the cytoskeleton via adhesive proteins. However, if a section of membrane and cytoskeleton detach, then the fluid pushes on this freed membrane creating a bleb.

Membrane-cytoskeleton detachment is thought to be caused by a contraction of the actin mesh by myosin II fibers that are dispersed in the cytoskeleton [**18, 66**]. The forces created by the contraction are thought to break the bonds between the actin cortex and

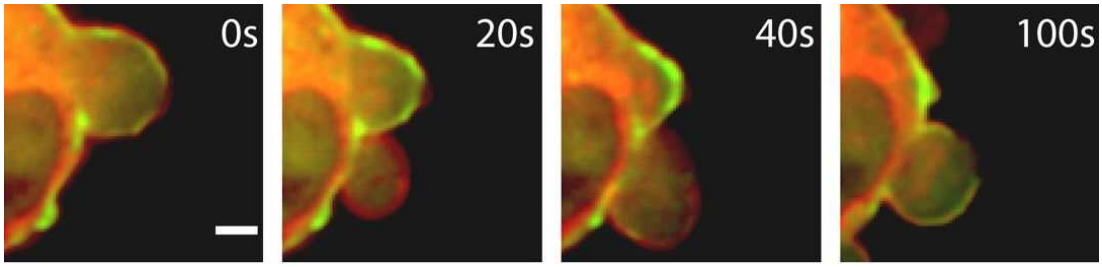


FIGURE 2.13. An expanding and retracting bleb in an M2 cell, where actin is labelled in green. After the bleb has fully inflated (frame 2), actin is recruited into the bleb to build a new cortex in order to retract the bleb [**18**]

the plasma membrane. A gap size of 0.5-1 $\mu$m in the cytoskeleton/membrane connections is enough to initiate bleb formation [**71**].

A bleb takes about 3-7 seconds to fully form. It then stays fully inflated (1-10 $\mu m$ in diameter [**18**]) for about 10-20 seconds. During some cellular activities, the bleb will retract back inward and this is a slower process, taking about 1 minute [**3, 22, 61**]. (See Figure 2.13 for a time sequence of blebbing).

The retraction phase of blebbing is where the cytoskeleton takes on an active role. When the membrane breaks away from the cytoskeleton during the initial detachment, some of the actin-binding proteins such as ankyrin and protein 4.1 remain adhered to the membrane [**18**]. These proteins act as anchors upon which a new cytoskeleton can form. As the cytosol flows into the blebbed region it carries with it actin, myosin and other cytoskeletal protein monomers. These subunits collect at the bleb's perimeter and form a new cytoskeleton within the bleb [**18**]. It takes the cell approximately 30 seconds to reassemble a cortex inside the bleb before retraction can begin [**18**]. The new cortex is built to a thickness of 10-20 nm (3-4 actin filaments thick) with gap sizes of approximately 200 nm [**18**] (see Figure 2.14). Myosin II, present in this new cytoskeleton, creates contractions which pull the blebbed membrane inward to be reattached to the base cytoskeleton [**18, 42**].

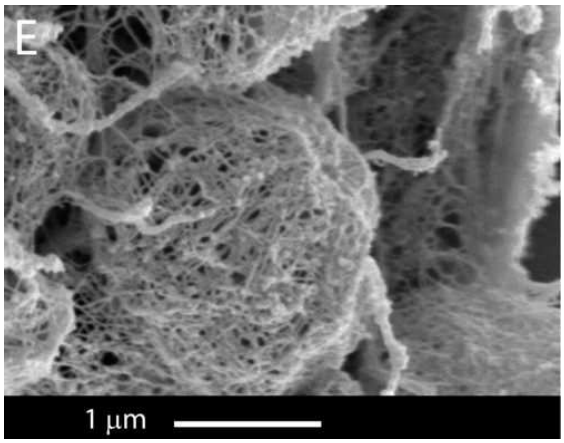


FIGURE 2.14. The newly formed actin cortex inside a bleb of a dividing HeLa cell [**18**]

### 2.3. Computational Modeling Difficulties

The main issue that arises when modeling the cytoskeleton is how to balance representing the details of its complex structure with computational efficiency. To use rough numbers, there are on the order of $10^5$, 1 micron long filaments in a typical cell [**10**]. The majority of filaments are concentrated at the cell periphery as part of the cytoskeleton. These filaments are crosslinked to one another to form the actin network. The typical length of a filament segment between crosslinks is on the order of 100 $nm$ [**78**]. This means there are approximately one million filament segments in the cytoskeleton and roughly 500,000 crosslink protein complexes. Such large numbers of components quickly become difficult to deal with computationally.

For example, suppose one wants to carry out a full simulation of the cytoskeletal network as the cell undergoes a deformation. One approach would be to treat the filament segment endpoints as point masses connected to each other by springs, and write an equation of motion for each point mass using Newton's law. These equations could then be evolved using a numerical scheme. Due to the small distances between neighboring point masses, the time steps taken will have to be small as well to maintain numerical stability. Using the numbers above, the space step based on segment length will be on the order of 100 $nm$. Assuming a wave propagation speed on the order of $10^4$ square nano-seconds (determined by the elasticity modulus and linear mass density of an actin

filament), the time scale would be on the order of 0.01 nano-seconds. Cellular protrusions develop on the order of seconds, thus it would take $10^{11}$ time steps to advance one second of the simulation. During each time step, at least $10^6$ flops (floating point operations) need to take place to update all positions. This is a total of $10^{17}$ flops to advance the system one second. At a flop speed of 100 Giga-flops per second, this would take $10^6$ seconds or about 12 days. To advance the simulation 30 seconds would take about a year. If the algorithm can be parallelized, then the time can be reduced by dividing the time by the number of available processors. However, even with parallelization this is still a large computation.

A different approach might be to replace the solving of the equations of motion with an energy minimization problem. Mechanical theory states that systems will always tend towards states of minimal potential energy. Once the boundaries of the cytoskeletal network have been moved, an energy minimization algorithm could be applied to move all internal filaments to a state of mechanical equilibrium. Larger time steps should be possible in this approach as compared to the first method, although one must still be careful. Any minimization algorithm is only guaranteed to converge if the process starts with an initial guess that is "close" to the solution. The savings in number of time steps is offset by the expense of the energy minimization procedure. A simple linear gradient search algorithm takes on the order of $n$ iterations to converge, where $n$ is the number of elements being moved in the minimization ($10^6$ filament segment endpoints). Each iteration requires the computation of the gradient vector, approximately $100n$ flops. To do one energy minimization would thus take about $10^{14}$ flops, or about 17 minutes on a 100 Giga-flop speed machine. This time can be multiplied by the number of deformation/minimization steps to be taken in one second of simulation (which could vary widely based on the speed and magnitude of deformation) to obtain the total CPU time for one second of simulation.

From these estimations, it should be clear that modeling the cytoskeleton as a network of crosslinked fibers over the time interval of whole cell deformation is very computationally expensive. Because of these difficulties, a variety of methods have been used to create and model simplified versions of the cytoskeleton. There are also efforts being made to thoroughly model a small patch of cytoskeleton to understand how it reacts to various stresses. This research coincides with the work being conducted in this thesis, however the other groups have not yet progressed to the stage of utilizing their detailed models to simulate cell-level deformation. Some examples of different cytoskeletal modeling approaches will be presented next.

## 2.4. Models of the Cytoskeleton

The development of a continuum-microscopic model of the cytoskeleton in this thesis work is timely for this research field, as evidenced by a recent review article by Mofrad [**59**]. In this paper, he presents a summary of the various theoretical and computational techniques currently available for modeling the actin cortex. He mentions that there are a wide range of models "ranging from continuum to discrete descriptions of the cytoskeleton". At the continuum level, the cytoskeleton has been described as an elastic or viscoelastic material, a porous gel, and a soft glassy material. In the discrete catergory, the actin network has been represented via a tensegrity model and also polymer or filament based descriptions. He explains that these different descriptions are a result of the relevant length scales (whole cell vs. small cytoskeletal patch) and the behavior one is trying to capture with a given model.

He acknowledges the limitations of continuum level models that coarse-grain the microstructure, leaving out important details. He states that continuum models "are only as good as the constitutive laws (stress-strain relation) on which they are based", and that continuum descriptions only allow for a "limited number of constants to characterize the cell's behavior". These models do not capture microscopic thermal fluctuations, which via coordination, can affect whole-cell level mechanical responses. Continuum models

also exclude the representation of local biochemical reactions that can collectively trigger cell deformation.

The computational models described in this review article are clearly split into discrete and continuum approaches. The research door is left wide open on the goal of creating microscopically-informed continuum-level models for this biological material. He states that "As a first milestone, continuum models must be able to incorporate an understanding of the cytoskeletal stress field" stemming from the behavior of "cytoskeletal fibers", because "changes in stress-fiber patterns affect cell shape and orientation".

This review article clearly provides motivation for the creation of a continuum-microscopic model of the cytoskeleton, and the hope is that the new model presented in this thesis will be of great utility in this field. Before briefly describing this new model, a summary of other cytoskeletal models will be presented.

**2.4.1. Coarse Graining Models.** Coarse-graining is a method which uses averaging and smoothing tools to create a lower resolution description of the problem to be modeled. Li et al. [**51**] used coarse graining concepts to create a model of the red blood cell (RBC) cytoskeleton in order to study shape configurations when the cell undergoes certain deformations. The RBC has a simple structure composed of a thin cytoskeleton constructed from spectrin protein complexes and short actin filaments. The spectrin tetramers crosslink with the actin filaments to form junctional complexes. These junctions typically bring together six spectrin tetramers, thus creating a hexagonal tessellation pattern in the cytoskeleton. There are approximately $10^5$ spectrin junctions in an RBC. Li et al.'s model of the cytoskeleton consisted of a sample number of these junction complexes (20,000 out of 100,000). The spectrin filaments between these junctions are not explicitly modeled, but instead are represented by potential forces. This model of the RBC was then used for numerical optical tweezer experiments.

Pivkin et al. [**65**] developed another coarse-grained model of the red blood cell for the intended purpose of modeling the RBC in blood flow. They employed the model of Li et al. [**51**] as a starting point for their model, and coarse-grained it further. They

ran their simulations with 5000, 500, 100 and 50 junctional complexes, utilizing concepts from mean field theory to adjust parameter values. They did similar numerical optical tweezer experiments and concluded that a minimum of 100 complex points is required to get an accurate rendering of the deformed RBC. However they do not quantify their meaning of the term "accuracy". They then took the coarse-grained RBC and placed it in a simulated flow tube to mimic blood flow in a capillary to study how blood flow deforms the RBC. Their simulations were able to produce the expected "parachute" shape observed for RBCs in normal blood flow.

**2.4.2. Network Symmetry Models.** Another way of simplifying the representation of a system is to simplify the geometry by using regular patterns. In a paper by Boey et al. [**11**] they created a model to study large deformations of a patch of red blood cell cytoskeleton. The RBC cytoskeleton does have general 6-fold symmetry, creating a triangular pattern. This model by Boey et al. creates a 6-fold, two-dimensional network to test how the network behaves under large deformations. The spectrin proteins are represented as polymer chains with $n$ segments each. In their simulations they used 16 junctional complexes and either 12 or 26 segments per spectrin . They found that their networks behaved like Hookean spring networks up to moderate deformations. At large deformations (more than 50% of its equilibrium dimensions), they found non-Hookean regimes for the elastic moduli of the network.

A coarse-grained version of this microscopic model was utilized by Discher et al. [**23**] in a whole-cell model of the RBC undergoing micropipette aspiration. The spectrin polymer chains composed of multiple segments were replaced by two force potentials (like in the Li paper [**51**]). Utilizing mean field theory, the elastic moduli found in computational experiments of [**11**] are used in this whole cell model of the RBC. This model includes approximately 6000 junctional complexes with 6-fold, fixed connectivity. These complexes are moved on the surface of the RBC during a simulated micropipette aspiration by an energy minimization procedure. One simulation took approximately one week to run on a 200 MHz machine.

Palmer et al. [**63**] used a different type of symmetric network to propose a constitutive model for the stress-strain relationship of actin networks. They employ the 8-chain network model of Arruda and Boyce [**5**] to describe the actin mesh. This three-dimensional model is built from filament "cubes". Each cube has a filament beginning at each of its eight corners that extends inward, where the filaments all meet at a junction. These blocks are averaged, idealized representations of the random actin network configurations found in real cells. The goal of this research was to test this cytoskeleton model's response to shear stress. They found that the 8-chain model did well at mimicking the nonlinear stress-strain relationship observed in the F-actin network shear strain experiments of Gardel et al.[**30**].

**2.4.3. Localized Models.** Many researchers realize that modeling the details of the microscopic structure of the cytoskeleton is important to understanding the material's macroscopic behavior. Several research groups have undertaken the task of thoroughly modeling a small portion of the cytoskeleton to understand its mechanical respone to various stresses. The hope is that the results of these small patch experiments will shed light on the behavior of the whole actin network in a cell.

In the Kwon et al. article [**46**] on modeling actin networks, they modeled a small block of the crosslinked cytoskeleton. The cytoskeleton of a typical cell contains on the order of $10^5$ crosslinked filaments. In this paper, they modeled a $400nm$-length cube that contains approximately 50-100, $350nm$ long actin filaments. The filaments were represented as elastic Euler-Bernoulli beams, and crosslinked to one another via short polymers that have the same material properties as the actin filaments. The purpose of this model was to determine the components of the stiffness tensor $C_{ijkl}$ for the linear stress-strain relationship $\sigma = C\epsilon$ of different actin networks. Their experiments were run for networks with varying filament densities, as well as different distributions of filament orientations. This model performed well for isotropic and nearly isotropic systems, but had large errors when the distribution of filament orientations was far from uniform.

In a similar study, Huisman et al. created an actin network model to examine its mechanical behavior under shear strain [**37**]. They modeled a three-dimensional patch of cytoskeleton (cube of length $= 2\mu m$) containing several hundred actin filaments, connected to each other via rigid crosslinks . Stretching, bending and torsional stiffness values found in the literature were assigned to the filaments, and a large-scale shearing force field was applied to the network. The filaments were found to reorient themselves in the direction of shear, and the computed shear stiffening seems to match the experimental calculations in the literature of similar networks. Their computational experiments were run for varying filament and crosslink concentrations, as well as different filament lengths. They conclude that the response of the network is highly dependent upon the topology of the filament mesh, which backs up the general claim in this thesis that differences in the microscopic structure do affect macroscopic material properties.

Yet another microscopic scale computational investigation was conducted by Head, Levine, and MacKintosh in [**34**]. In this study, they explored the response of two-dimensional model networks to extensional and shear stresses. Their main interest was to look at how strain is distributed in such networks, dependent on crosslink density and filament length. They discovered two distinct regimes, where strains are uniformly and non-uniformly distributed.

Lastly, Buxton et al. [**13**] recently presented a computational model of actin networks built utilizing actin dynamics information. Their initial network begins with 100, 1-micron long actin filaments, that they then allow to polymerize and depolymerize based on various rates and probabilities. The filaments can also undergo capping, severing and crosslinking. The network develops until it reaches a steady state (a state where its average properties remain approximately constant over a specified length of time). The network is then placed under shear stress in order to examine its mechanical response. Different networks were built based on different actin dynamics rates, and the mechanical responses of these networks were compared. The networks upon which these simulations were carried out typically consisted of approximately $10^2 - 10^3$ filaments of lengths $2 -$

$9\mu m$, with about $10^3$ crosslinks connecting them. Each simulation took about 100 hours of CPU time.

**2.4.4. Tensegrity Models.** A tensegrity (tensional integrity) model of the cell was introduced in 1993 by Ingber in [**39**] to model the deformation of cells adhering to a substrate. This model consists of two types of prestressed elements: interconnected tension-bearing elements which represent the actin filaments of the cytoskeleton and compression-bearing elements, which represent microtubules [**73**]. This model assumes that the cell's shape and integrity derives from the cytoskeleton, an active mechanical structure capable of producing tension.

A physical model of Ingber's idea was constructed from 6 wooden dowels (struts) for the microtubules, and 24 elastic strings for the actin filaments. The zero external stress state of this system resting on a surface is a rounded shape [**76**], but when a force is applied to the top of the structure, it flattens out. The model is successful at capturing the strain-hardening observed in cells spreading over a substrate. However, it does not address the cytoskeleton's ability to rearrange and remodel itself during deformation [**76**]. Stamenovic et al. [**73**] used this tensegrity model to analytically compute upper and lower bounds for the Young's modulus of cells. They compare their results against experimental data, finding that the empirical moduli in general fall within their theoretically derived bounds.

**2.4.5. Continuum Models.** The models discussed so far have been discrete in nature, characterizing the cytoskeleton as a network of crosslinked filaments. There is also a body of research dedicated to the treatment of the cytoskeleton as a continuum.

Alt and Dembo in [**4**] utilized a two-phase fluid description of the cytoplasm in ameboid cells. The cytosol (water-like substance within the cell) is represented as a Newtonian fluid, and the cytoskeleton is represented as a highly viscous, polymeric fluid. This characterization is used under the assumption that the crosslinks in the cytoskeleton are constantly rearranging, allowing the network to adapt and move easily (like a fluid). This model is used to simulate the formation of a lamellipodium during cell migration.

During this phenomenon, the cytoskeleton undergoes many structural rearrangements. The model of Alt and Dembo [**4**] is used for understanding the general stages of this process.

Charras et al [**16, 19**], characterized the cytoskeleton as a solid porous medium. They propose that the actin network with interspersed cytoplasmic fluid should be thought of as a "sponge" with pressure diffusion occurring over time. They demonstrate experimentally that localized contractions of the actin mesh can create local pressure increases that do not instantaneously equilibrate across the cell. They develop a linear constitutive law for the cytoskeleton/cytosol complex using concepts from mixture theory. The stress-strain relationship is of the form: $\sigma = E\epsilon - p$, with $\sigma$ the stress, $E$ the bulk elasticity modulus, $\epsilon$ the strain, and $p$ the fluid pressure. Darcy's law for flow through porous media is used to update the fluid pressure term. Their theoretical model was developed to explain the cellular phenomenon of bleb formation.

### 2.5. New Cytoskeletal Model

As evidenced by the previous section, many different models have been developed and implemented to simulate various features and scales of cytoskeletal deformation. The goal of this thesis work is to model whole-cell deformation of the cytoskeleton. This deformation is orchestrated by changes in the microstructure brought about by chemical reactions and mechanical stresses. Models of such phenomena must be whole-cell scale, but they must also include detailed microscopic information. The models presented in Section 2.4 each fall short in meeting these needs in some way.

The coarse graining models of Li et al. and Pivkin et al. can be reasonably utilized for simulations of the RBC due to this cell's nearly homogeneous structure. For other types of cells with very complex, entangled cytoskeletons, these methods would not be able to create accurate structural representations. The patterned models of Boey et al. and Palmer et al. suffer from the same problems. Again for the RBC, these models may be sufficient due to the cell's dominant 6-fold symmetry. However, other cytoskeletal

networks do not follow patterns. Idealizing the actin cortex with the 8-chain model of [**5**] will likely lead to the oversight of microscopic inhomogeneities that create interesting local dynamics in the deforming network.

The tensegrity model of Ingber [**39**] offers an approach to understanding the general structural respone of the cell to external forces, but does not address microscopic issues such as cytoskeletal remodeling. The simplicity of the structure in this model also does not address spatially-varying mechanical properties of a typically heterogeneous cytoskeleton. The continuum models of Alt et al. and Charras et al. also disregard the inherent microstructure of the polymer network. Utilizing a continuum description of the cytoskeleton to model deformation requires a constitutive law to close the system. This constitutive law is complex and time-varying, and dependent on the microstructure of the medium. Without representing this microstructure in some way, the models in [**4**] and [**16**] do not reflect the changes and rearrangements occurring in the cytoskeleton that lead to varying mechanical properties.

The vast array of localized models of the cytoskeleton provide a great deal of insight into the mechanical response of small patches of actin networks. These models utilize detailed descriptions of the microstructure for computational experiments. Although none of the papers mentioned above makes this claim, it would be difficult to justify the idea that the behavior of small regions of cytoskeleton encapsulate the behavior of the entire cellular network. The anisotropy of the cytoskeleton from one patch to the next prevents one from drawing such a conclusion. In [**37**] the author specifically states that the stiffness response of the network is highly dependent on the concentrations of the different proteins in the cytoskeleton. These protein levels can certainly vary in different parts of the cortex as the cell undergoes locomotion and shape change. Currently, the level of detail introduced by these models cannot be extended to a whole-cell model of the cytoskeleton due to computational limitations.

What is needed is a combined approach that utilizes time and space varying microscopic information in an efficient continuum-level model of the medium. This can

be accomplished by extending a basic continuum-microscopic method. The details of this new approach will be presented in the forthcoming chapters, but a brief outline of the algorithm will now be described, highlighting the new features which overcome the difficulties of other models.

To review, the basic steps of a continuum-microscopic approach are:

(1) Create a microscopic instantiation of the system.

(2) Advance the microscopic system a short number of time steps until the data values reach a quasi-state of equilibrium, or until enough data has been collected to discern a pattern of behavior.

(3) Average the microscopic data and use it to update the macroscopic constitutive laws

(4) Advance the continuum level system one large time step

(5) Repeast steps 1-4

A schematic of this process is depicted in Figure 2.15.

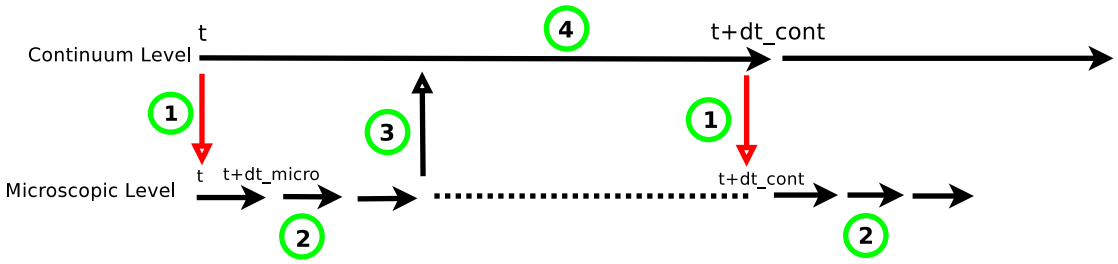


FIGURE 2.15. A visual depiction of the steps in a continuum-microscopic algorithm

The method begins with a detailed microscopic description of the entire cytoskeleton, similar to the three-dimensional networks created in [**37**]. In response to some imposed stresses, the whole system is advanced a short number of microscopic time steps (size $\Delta t_{micro}$ to maintain numerical stability). After each $\Delta t_{micro}$ step, microscopic data on filament variables (such as orientation angles and strain) is collected and saved in the form of distribution functions $f(x, t_s)$, where $x$ denotes the variable and $t_s$ denotes the current time step. After the micro-steps are complete, space-averaged mechanical properties are computed and passed to the continuum level in Step 3 of the general algorithm. Step

4 is then carried out to complete one cycle, using a much larger time step $\Delta t_{cont} \gg \Delta t_{micro}$ (permitted by the continuum-level numerical scheme). Instead of using assumed distribution functions (e.g. Gaussian, uniform) to carry out Step 1 in the next cycle, the $f(x, t_s)$ functions saved during the previous micro-steps are utilized to predict a set of distribution functions for the microscopic variables at this new time step. This prediction is done utilizing approximation and extrapolation methods. These predicted distributions are then used to instantiate a new microstructure and the cycle continues on.

The benefits of this new algorithm are: (1) it provides a possible solution to the problem of microscopic reinstantiation for systems with time-varying microstructures in the continuum-microscopic approach, (2) this model incorporates heterogeneous microscopic data into the computation of the whole-cell cytoskeleton's mechanical properties, (3) it is a computationally efficient model in that only a small fraction of the full time interval of simulation is spent doing microscopic advancements.

Concepts from probability theory and statistical mechanics, relevant to this research effort, will be presented in the next chapter in preparation for the derivation and justification of this new algorithm.

CHAPTER 3

# Mathematical Theory

In this chapter the basic elements of probability theory, probability distribution function estimation, thermodynamics and statistical mechanics will be presented.

## 3.1. Probability Theory

Concepts in probability theory that pertain to the new cytoskeletal model developed in this thesis, include probability distribution functions (PDFs) and statistical convergence.

**3.1.1. Basic Definitions.** A random variable $X$ is defined by a set of of possible states (called a sample space $\Omega$) and a probability distribution function $f(X)$. This function $f(X)$ is non-negative, and is defined so that it normalizes as follows:

$$\int_{-\infty}^{\infty} f(X)dX = 1$$

The probability that $X$ takes on a value between $x$ and $x + dx$ is given by $f(x)dx$ [**77**]. The cumulative distribution function $F(X)$ is defined to be the integral of the PDF, $f(x)$ from negative infinity to $x$:

$$F(x) = \int_{-\infty}^{x} f(X)dX$$

In words, the cumulative distribution function (CDF) gives the probability that the random variable $X$ is found in the range $-\infty$ to $x$.

A PDF can be described by several different sets of statistics: moments, centered moments, and cumulants. The first moment, (also known as the mean, average, or

expected value) is defined as:

$$\langle X \rangle = \int_{\Omega} x f(x) dx$$

This expression is typically denoted as $\mu_1$. Higher order moments are defined as:

$$\mu_m = \langle X^m \rangle = \int_{\Omega} x^m f(x) dx$$

Moments can be defined in a different way by introducing the characteristic function. The characteristic function $G(k)$ of a PDF $f(X)$ is defined as:

$$G(k) = \langle e^{ikX} \rangle = \int_{\Omega} e^{ikx} f(x) dx \tag{3.1}$$

for $k$ any real number. This function is also referred to as the moment-generating function because if a Taylor expansion is done about $k = 0$ the coefficients of the expansion are the moments $\mu_m$ of the distribution [**77**]. In expansion form, $G(k)$ looks like:

$$G(k) = \sum_{m=0}^{\infty} \frac{(ik)^m}{m!} \mu_m \tag{3.2}$$

Centered moments are defined as:

$$\mu_m^c = \langle (X - \mu_1)^m \rangle = \int_{\Omega} (x - \mu_1)^m f(x) dx$$

The first centered moment $\mu_1^c$ is zero. The second centered moment $\mu_2^c = \langle (X - \langle X \rangle)^2 \rangle$ is the variance (or spread) of the data.

Cumulants ($\kappa_m$) are yet another set of statistics that can be used to describe a PDF. They are defined as the coefficients of the Taylor expansion of the natural logarithm of the characteristic function:

$$\log(G(k)) = \sum_{m=1}^{\infty} \frac{(ik)^m}{m!} \kappa_m$$

Each of these three sets of statistics can be written in terms of the other sets by algebraic expressions.

Problems often involve more than one variable. A joint probability distribution function $f(X_1, X_2, ...X_N)$ for the variables $X_1, X_2, .., X_N$ is the analogue to the single variable case. The probability that $X_1 \in (x_1, x_1 + dx_1)$, $X_2 \in (x_2, x_2 + dx_2)$, etc is given by:

$$f(x_1, x_2, .., x_N)dx_1dx_2..dx_N$$

Moments can be defined for these multivariate distributions as:

$$\langle X_1^{m_1} X_2^{m_2} .... X_N^{m_N} \rangle = \int x_1^{m_1} x_2^{m_2} ... x_N^{m_N} f(x_1, x_2, ...x_N)dx_1dx_2...dx_N$$

denoted as $\mu_{m_1,m_2,...m_N}$.

The joint probability distribution function can be written as a product of single variable PDFs if the variables are statistically independent. The statistical independence of random variables $X$ and $Y$ can be established by computing their correlation coefficient. The correlation coefficient $r$ is computed by finding the individual variances of $X$ and $Y$ and their covariance:

$$Var_X = \sum_{i=1}^{n} (X_i - \langle X \rangle)^2$$

$$Var_Y = \sum_{i=1}^{n} (Y_i - \langle Y \rangle)^2$$

$$Cov(X, Y) = \sum_{i=1}^{n} (X_i - \langle X \rangle)(Y_i - \langle Y \rangle)$$

$$r = \frac{Cov(X, Y)}{Var_X \cdot Var_Y}$$

If $r = 0$ then $X$ and $Y$ are uncorrelated and statistically independent. The closer $r$ is to 1, the stronger the correlation is between $X$ and $Y$.

Two other types of probability distribution functions that often arise in multivariate problems are the conditional and marginal distributions. The conditional probability $f(X_1, X_2...X_r|X_{r+1}...X_n)$ is defined to be the probability of $X_1, X_2, ...X_r$ being in states

$x_1, x_2, ...x_r$ given that $X_{r+1}..X_n$ are known to be in states $x_{r+1}..x_n$. The marginal distribution is the probability that $X_1, X_2...X_r$ are in states $x_1, x_2, ..x_r$ no matter what states $X_{r+1}..X_n$ are in. It is defined as:

$$f_r(X_1, X_2, ...X_r) = \int f(X_1, X_2, ..X_r, X_{r+1}, ..X_n) dX_{r+1}..dX_n$$

The full joint probability distribution function $f(X_1, X_2, ...X_n)$ can be written in terms of marginal and conditional distributions as follows:

$$f(X_1, X_2, ...X_n) = f_{n-r}(X_{r+1}, ...X_n) \cdot f(X_1, X_2, ..X_r | X_{r+1}, ..X_n)$$

This relationship is known as Bayes' Rule and is a cornerstone of probability theory.

**3.1.2. Law of Large Numbers.** An important theorem of probability theory is the law of large numbers. This theorem will be useful for verifying statistical convergence of portions of the new algorithm.

**Law of Large Numbers**: Let $X_1, X_2...X_n$ be a sequence of independent random variables each with the same finite mean $\mu$ and finite variance $\sigma^2$. Then as $n \to \infty$, the average of the first $n$ of the $X$'s:

$$\frac{1}{n} S_n = \frac{1}{n}(X_1 + X_2 + ... + X_n) \to \mu$$

In plain language, as the number of samples generated from a random process is increased, the average of these samples will converge to the process' underlying mean [**32**].

**Proof**: To prove this statement, simply look at the expected value of the quantity: $\frac{1}{n}S_n$:

$$\begin{aligned} E\left(\frac{1}{n}S_n\right) &= E\left(\frac{1}{n}(X_1 + X_2 + ...X_n)\right) \\ &= \frac{1}{n}E(X_1 + X_2 + ... + X_n) \\ &= \frac{1}{n}\int (X_1 + X_2 + ..X_n)f(X)dX \\ &= \frac{1}{n}\left[\int X_1 f(X)dX + \int X_2 f(X)dX + ... + \int X_n f(X)dX\right] \\ &= \frac{1}{n}(n\mu) \\ &= \mu \end{aligned}$$

**Part II of the Law of Large Numbers**: The error between the sequence's average and the true mean will converge to zero in mean-square, meaning:

$$E\left(\left(\frac{1}{n}S_n - \mu\right)^2\right) \to 0 \quad \text{as } n \to \infty$$

**Proof**: This statement can also be proved by simply computing the expected value

written above:

$$\begin{aligned}
E\left(\left(\frac{1}{n}S_n - \mu\right)^2\right) &= E\left(\left(\frac{1}{n}S_n - E\left(\frac{1}{n}S_n\right)\right)^2\right) \\
&= Var\left(\frac{1}{n}S_n\right) \\
&= \frac{1}{n^2}Var(S_n) \\
&= \frac{1}{n^2}Var(X_1 + X_2 + ... + X_n) \\
&= \frac{1}{n^2}(Var(X_1) + Var(X_2) + ... + Var(X_n)) \\
&( \quad \text{By independence of the } X_i's) \\
&= \frac{1}{n^2}n\sigma^2 \\
&= \frac{\sigma^2}{n} \to 0 \quad as \ n \to \infty
\end{aligned}$$

Since the error squared converges to zero like $\sigma^2/n$, the error should converge to zero like $\sigma/\sqrt{n}$. Since $\sigma$ is a constant, the rate of convergence is given by $\frac{1}{\sqrt{n}}$. Portions of the new model that involve the use of probability distribution functions will be validated utilizing this theorem. The expected convergence behavior will be $O(n^{-1/2})$.

### 3.2. The PDF Estimation Problem

As described in Chapters 1 and 2, a continuum-microscopic model of the cytoskeleton is the subject of this research work. Step 1 of the general algorithm outlined in Section 2.5 constructs a new microscopic configuration at continuum time step $t_{n+1}$. In the method proposed in this thesis, this reinstantiation will be done using probability distribution functions predicted from past microscopic data distributions at microscopic time steps $t_{n_0}, t_{n_1}, ..., t_{n_r}$ where $t_{n_0}$ equals continuum time $t_n$, and $t_{n_r} \ll t_{n+1}$. These past distribution functions must each be constructed from the microscopic data sets at time $t_{n_i}, i = 0..r$. The process of constructing a PDF from a given data set is known as the probability distribution function estimation problem.

PDF estimation is a classic problem of probability theory. A variety of methods have been developed and tested in pursuit of its solution [**72**]. First a distinction should be made between the two main branches of this field: parametric and non-parametric PDF estimation. In parametric PDF estimation, the data is assumed to come from a particular parametric family of distributions. For example, a normal distribution is determined by two parameters: the mean $\mu$ and the variance $\sigma^2$. If a data set is assumed to come from a normal distribution, an estimate of $\mu$ and $\sigma^2$ is all that is required to construct a PDF. However, if a data set is not known to have been generated from a specific class of distributions, the PDF estimation problem becomes non-parametric. In this case, the data set is the only information available for PDF estimation. The microscopic data on filament orientations and strains collected from the heterogeneous structure of the cytoskeletal network is not known to conform to a known parametric family of distributions, thus non-parametric PDF estimation will be required.

Common non-parametric techniques include kernel estimation, histogram interpolation, and series expansion estimators (such as the characteristic function and the Hermite polynomials). Each of these methods was explored as a potential candidate for use in the algorithm described in the next chapter. A description of each method will be presented. Each method will then be given the same task of estimating the PDF of a data set generated from a known normal distribution ($\mu = 0$ and $\sigma^2 = 1$). The results will then be compared. The data set (in histogram form) and its underlying distribution function are shown in Figure 3.1.

**3.2.1. Characteristic Function.** As stated previously, the characteristic function is defined as:

$$G(k) = \langle e^{ikX} \rangle = \int e^{ikx} f(x) dx \tag{3.3}$$

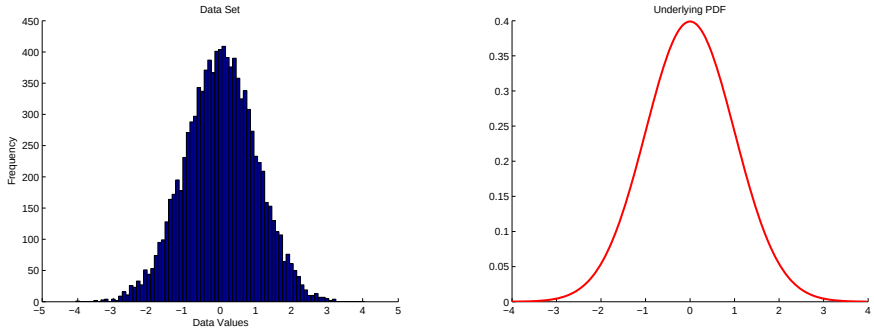


FIGURE 3.1. Data set and its underlying PDF to be used for testing various PDF estimation methods

Looking at Equation 3.3 one can see that the characteristic function and the probability distribution function $f(x)$ form a Fourier transform pair. Thus,

$$f(X) = \frac{1}{2\pi} \int G(k) e^{-ikX} dk \tag{3.4}$$

If an expression for $G(k)$ can be found, it can be used to define $f(X)$. The expansion form of the characteristic equation is given by 3.2. The coefficients of this expansion (the moments) can be computed by simple averaging of the given data set:

$$\mu_j = \frac{\sum_{i=1}^{N} x_i^j}{N} \tag{3.5}$$

where $N$ is the number of data values. The expansion in 3.2 is an infinite series. An approximation to $G(k)$ can be found by truncating 3.2 to a finite number of terms $m$:

$$G(k) = 1 + ik\mu_1 - \frac{k^2}{2}\mu_2 - \frac{ik^3}{6}\mu_3 + ... \frac{(ik)^m}{m!}\mu_m$$

This truncated version of $G(k)$ will be used in 3.4 to estimate $f(X)$. From the given data set, the coefficients $\mu_j$ are computed using 3.5. The values of the first ten moments of the data are displayed in Table 1 along with the first ten moments of the data's known normal distribution.

Once $G(k)$ has been approximated, a Fourier transform is performed to find the estimated probability distribution function $\hat{f}(X)$. Estimated PDFs using $G(k)$ truncated

| Data Moments | Moments of N(0,1) |
|---|---|
| 0.0148 | 0 |
| 0.9882 | 1 |
| -0.0004 | 0 |
| 2.8888 | 3 |
| -0.2717 | 0 |
| 13.6961 | 15 |
| -4.0086 | 0 |
| 88.1064 | 105 |
| -55.0340 | 0 |
| 709.5603 | 945 |

TABLE 1. First ten moments computed from the data set, and the first ten moments of the true PDF

| Number of Expansion Terms | Error $E((f(x) - \hat{f}(x))^2)$ |
|---|---|
| 10 | 0.03177 |
| 20 | 0.01398 |
| 40 | 0.00567 |
| 60 | 0.00414 |
| 100 | 0.00487 |

TABLE 2. Mean Square Errors between the true PDF and the estimated PDFs from characteristic function expansions with moment coefficients

expansions with 10, 20, 40, 60, and 100 terms are shown in Figure 3.2. The red curve on each graph is the true PDF of the data.

An estimate of the error between the estimated PDF and the true PDF can be computed using the mean square error (MSE):

$$Error = E((f(x) - \hat{f}(x))^2)$$

where $f(x)$ is the true PDF, $\hat{f}(x)$ is the estimated PDF, and $E$ indicates an expection or average value. The MSEs for the five examples shown in Figure 3.2 are given in Table 2.

At first, the error decreases as the number of terms increase. However at 100 terms, the error is larger than in the 60 term case. Results of this nature could be attributed to the fact that the basis functions of this series expansion are the monomials. This basis

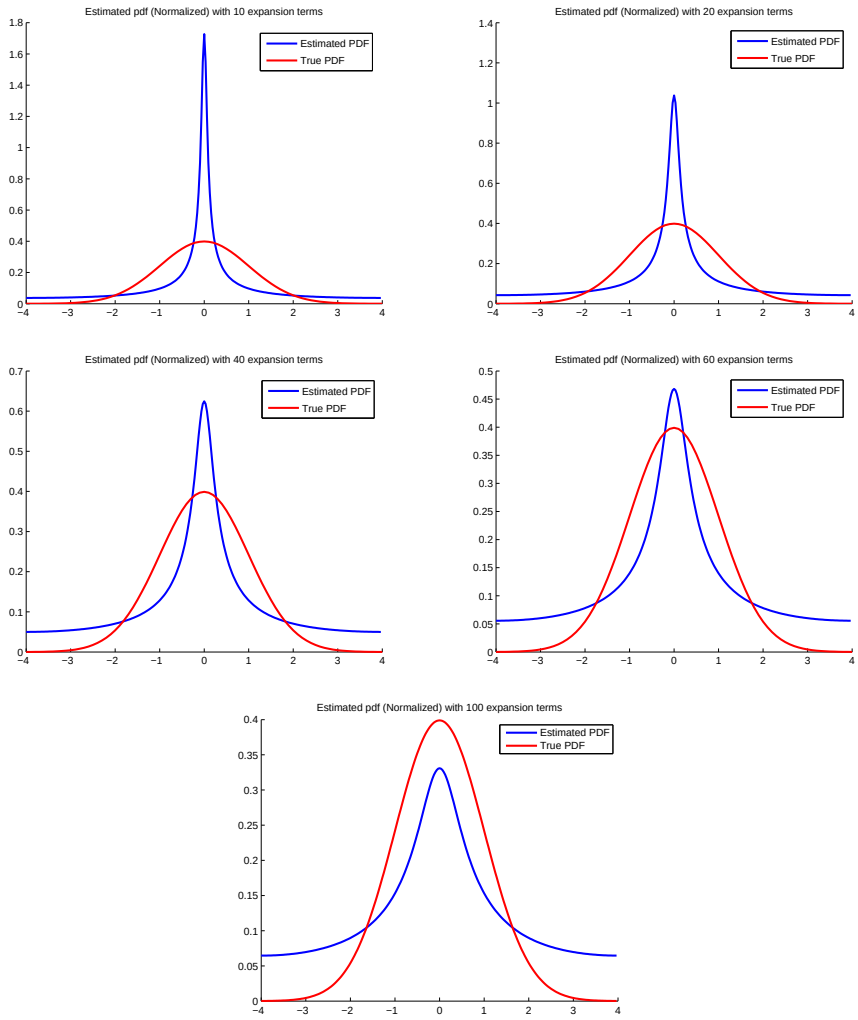


FIGURE 3.2. Estimated PDFs generated using the Fourier transform of a truncated characteristic function expansion, with 10, 20, 40, 60, and 100 expansion terms

set is known to be unstable for approximating functions, due to the fact that the monomials are very similar to one another (nearly linearly dependent) [**7**]. The coefficients $\mu_j$ computed from the data set get larger in magnitude as $j$ increases. Approximations constructed with monomials are very sensitive to these coefficients, so large values will likely cause poor approximation results.

Another option is to write the characteristic function expansion in terms of cumulants. The cumulants can be computed from the moments via the following algebraic

| Number of Expansion Terms | Error $E((f(x) - \hat{f}(x))^2)$ |
|---|---|
| 6 | 0.00708 |
| 10 | 0.00633 |
| 14 | 0.00612 |
| 34 | 0.00531 |
| 54 | 0.00591 |

TABLE 3. Mean Square Errors between the true PDF and the estimated PDFs from characteristic function expansions with cumulant coefficients

relationship:

$$\kappa_n = \mu_n - \sum_{k=1}^{n-1} \binom{n-1}{k-1} \kappa_k \mu_{n-k} \tag{3.6}$$

The characteristic equation can be written as:

$$G(k) = e^{log(G(k)} = e^{\sum_{m=1}^{\infty} \frac{(ik)^m}{m!} \kappa_m}$$

Estimated PDFs using $G(k)$ truncated expansions with 6, 10, 14, 34, and 54 terms are shown in Figure 3.3. (Expansions with $4n$ terms ($n$ an integer) result in values too large for the exponential function to compute in MATLAB, hence expansions with 20, 40, 60 and 100 terms could not be calculated. Expansions with $4n - 2$ terms are computable due to the mutual cancellation of large positive and negative terms.) The red curve on each graph is the true PDF of the data. The errors for these examples are given in Table 3.

The errors behave similarly to the moment expansion cases. Utilizing cumulants instead of moments does not solve the monomial basis problem, so these results are likely due to the same instability issues mentioned beforehand. Without the ability to utilize the infinity of terms in either expansion to compute the full characteristic function, this method will likely be unreliable for PDF estimation.

**3.2.2. Hermite Polynomial Expansion.** Another option for series expansion PDF estimation is to utilize a set of orthogonal basis functions. Orthogonal basis functions area linearly independent, which should solve some of the instability issues of the monomial

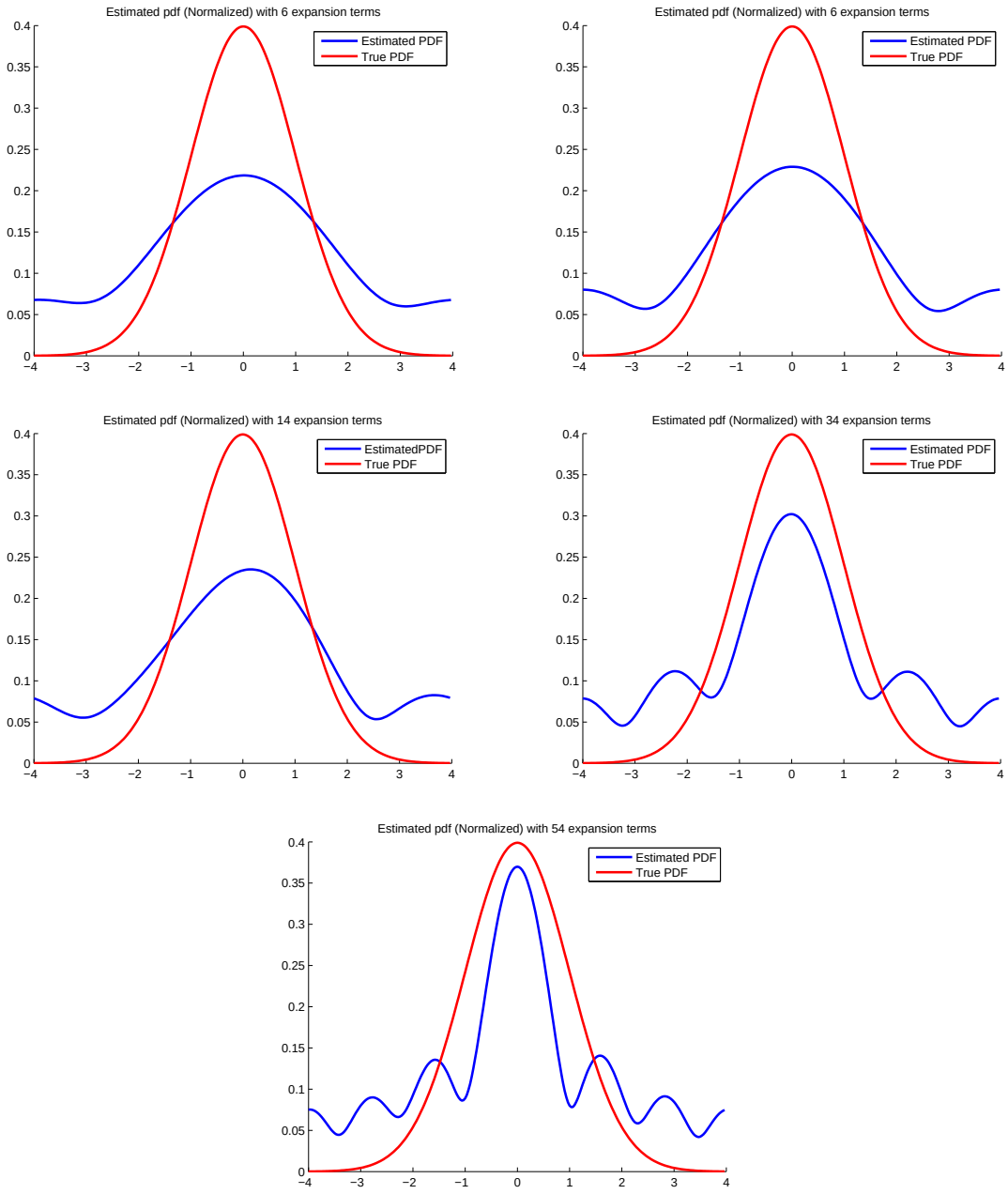


FIGURE 3.3. Estimated PDFs generated using the Fourier transform of a truncated characteristic function expansion containing cumulant coefficients, with 6, 10, 14, 34, and 54 expansion terms

basis. Also, the coefficients of the series expansion with an orthogonal basis are very easy to compute. Out of the classical orthogonal polynomial families (e.g. Lagrange, Laguerre, etc.), the Hermite polynomials are the only set that is orthogonal on the

interval $(-\infty, \infty)$. This property makes them a natural choice for PDF estimation since data could take on any real values.

The goal is to approximate the PDF $f(X)$, with a truncated Hermite polynomial expansion of $N$ terms:

$$\hat{f}(X) = \sum_{n=0}^{N} c_n H_n(x) \tag{3.7}$$

where $H_n(x)$ are the Hermite polynomials and $c_n$ are coefficients. The $H_n(x)$ can be generated from the following definition:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}$$

To establish orthogonality for this basis set, first define the inner product:

$$\int_{-\infty}^{\infty} H_m(x) H_n(x) e^{-x^2} dx \tag{3.8}$$

This integral is equal to zero if $m \neq n$ and equal to $n! 2^n \sqrt{\pi}$ if $m = n$. This property provides a means for finding the expansion coefficients. To see this, take Equation 3.7 and multiply it by $H_m(X)e^{-x^2}$ on both sides:

$$\hat{f}(x) H_m(x) e^{-x^2} = \sum_{n=0}^{N} \left[ c_n H_n(x) \right] H_m(x) e^{-x^2}$$

Integrate both sides:

$$\int_{-\infty}^{\infty} \hat{f}(x) H_m(x) e^{-x^2} dx = \int_{-\infty}^{\infty} \left[ \sum_{i=0}^{\infty} (c_n H_n(x)) H_m(x) e^{-x^2} \right] dx$$

Interchange the integral and sum on the right hand side to give:

$$\int_{-\infty}^{\infty} \hat{f}(x) H_m(x) e^{-x^2} dx = \sum_{i=0}^{\infty} \int_{-\infty}^{\infty} c_n H_n(x) H_m(x) e^{-x^2} dx$$

| Coefficient Number | Coefficient Values |
|---|---|
| 0 | 3.2675e-01 |
| 1 | 3.0605e-03 |
| 2 | -2.7405e-02 |
| 3 | -3.6687e-04 |
| 4 | 1.1450e-03 |
| 5 | 2.4470e-05 |
| 6 | -3.0837e-05 |
| 7 | -1.1394e-06 |
| 8 | 5.2492e-07 |
| 9 | 3.1466e-08 |

TABLE 4. The first ten coefficients for the Hermite polynomial expansion approximating a PDF

The integral on the right hand side is the inner product defined in Equation 3.8. The only non-zero quantity in the sum will be the integral where $m = n$, so:

$$\begin{aligned} \int_{-\infty}^{\infty} \hat{f}(x) H_m(x) e^{-x^2} dx &= \int_{-\infty}^{\infty} c_n H_n(x) H_n(x) e^{-x^2} dx \\ &= c_n n! 2^n \sqrt{\pi} \end{aligned}$$

Therefore the coefficients can be defined as:

$$c_n = \frac{1}{2^n n! \sqrt{\pi}} \int_{-\infty}^{\infty} \hat{f}(x) e^{-x^2} H_n(x) dx \tag{3.9}$$

This expression can be written as an expectation as:

$$c_n = \frac{1}{2^n n! \sqrt{\pi}} E(e^{-x^2} H_n(x))$$

These expectations can be calculated from the data set using:

$$c_n = \frac{1}{2^n n! \sqrt{\pi}} \frac{\sum_{i=1}^{N} e^{-x_i^2} H_n(x_i)}{N}$$

where $N$ is the total number of data points. The first ten coefficients for this data set are shown in Table 4:

Estimated PDFs using $\hat{f}(x)$ truncated expansions with 6, 10, 20, 40, and 60 terms are shown in Figure 3.4. Estimate of the MSEs between the estimated PDFs and the true PDF are given in Table 5.

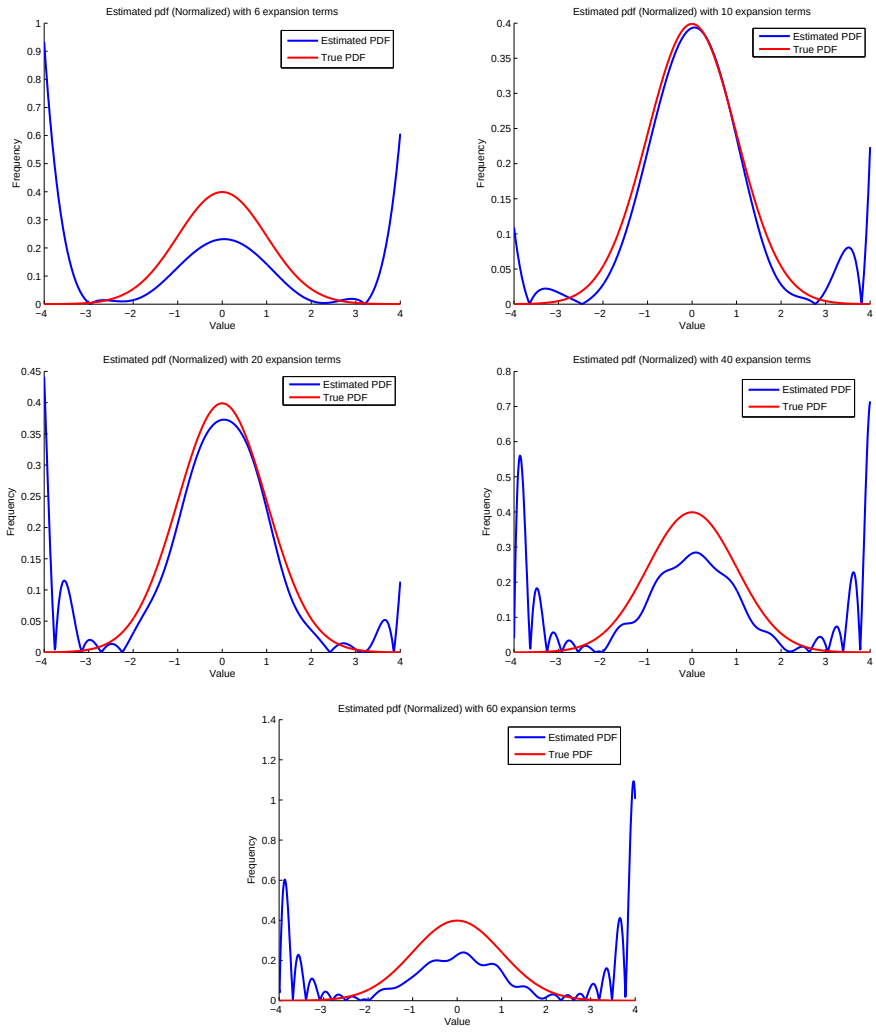


FIGURE 3.4. Estimated PDFs generated using a truncated Hermite polynomial expansion approximation with 6, 10, 20, 40, and 60 terms

As in the characteristic function series expansion case, the errors decrease at first and then increase with the number of terms. Qualitatively, the estimated PDF appears to match the Gaussian well in the middle of the domain, but results at the edges of the domain are poor. Unlike the coefficients in the characteristic function expansion, the $c_n$ values of the Hermite expansion decay as the terms increase. However, the coefficients are not small enough to overcome the values of $H_n(x)$ at the edges of the domain (where

| Number of Expansion Terms | Error $E((f(x) - \hat{f}(x))^2)$ |
|---|---|
| 6 | 0.03374 |
| 10 | 0.00109 |
| 20 | 0.00291 |
| 40 | 0.01914 |
| 60 | 0.03683 |

TABLE 5. Mean Square Errors between the true PDF and the estimated PDFs from Hermite polynomial expansions

$x$ has the highest magnitude). For example, in terms such as:

$$H_8(x) = x^8 - 28x^6 + 210x^4 - 420x^2 + 105$$

the $x^8$ leading term can be quite large for the $|x| > 1$ values. Multiplication by a $c_8$ value on the order of $10^{-8}$ is not small enough to render the $c_8 H_8(x)$ term insignificant. If the full expansion (infinite terms) is computed, there large values would likely even out. However, the truncation leaves behind these extreme tail values.

When using orthogonal basis functions $\phi_n$ to approximate a function $f(x)$, there is a theorem in approximation theory that states that the error: $\|f(x) - \sum_{i=1}^{n} c_i \phi_i\|_2 \to 0$ as $n \to \infty$, so it may appear incorrect that the errors computed in the above example do not converge to zero. However, this theorem assumes that the coefficients $c_i$ have been computed exactly via Equation 3.9. The exact computation is possible if $f(x)$ is known, but in this PDF estimation situation, $f(x)$ is unknown and the only available option is to approximate the $c_n$ values with data averages. Presumably the accuracy of $c_i$ would increase as the size of the data set increases (by the law of large numbers). Due to the overall magnitude of the errors and the poor quality of the estimated PDF at the edges of the domain, this method would likely not be a good candidate for PDF estimation for this research.

**3.2.3. Histogram Interpolation.** One of the oldest and most common PDF estimation techniques is histogram interpolation [**72**]. To construct a histogram, the first step is to establish the bins. Given a starting point $x_0$ and a bin width of $h$, the bins can be

| Bin Width | Error $E((f(x) - \hat{f}(x))^2)$ |
|---|---|
| 0.8 | 3.045e-5 |
| 0.5 | 1.589e-5 |
| 0.2 | 3.753e-5 |
| 0.1 | 8.584e-5 |
| 0.05 | 2.094e-4 |

TABLE 6. Mean Square Errors between the true PDF and the estimated PDFs from interpolated histograms with different bin widths

defined as the following intervals:

$$[x_0 + mh, x_0 + (m+1)h)$$

where $m$ is an integer. As a function, the histogram can then be defined as:

$$\hat{f}(x) = \frac{1}{nh}(\text{No. of data points in same bin as } x) \tag{3.10}$$

This $\hat{f}(x)$ is a piecewise constant approximation to the true underlying PDF function of the data. It is discontinuous, and therefore not differentiable. It can be approximated by a continuous function using interpolation, if a smooth function is needed for further manipulation. For these tests, cubic spline interpolation was used.

The choice of bin width and interval endpoints can produce very different results for $\hat{f}(x)$. Wide bins will tend to blur and smooth out details in the data. Bins of very small widths may not have enough data per bin to get a true picture of the data and the interpolated function will likely have many sharp peaks.

Estimated PDFs using histogram interpolation are shown in Figure 3.5 using various bin widths. The errors between the interpolated histogram and the true PDF are shown in Table 6:

The errors are on the order of $10^{-5}$, two orders of magnitude better than the two series expansion tests. The errors do vary with bin width as expected. The best result was with a bin width of 0.5. Since the true PDF is known in this case, an optimal bin width could be found by solving an error minimization problem. In this research

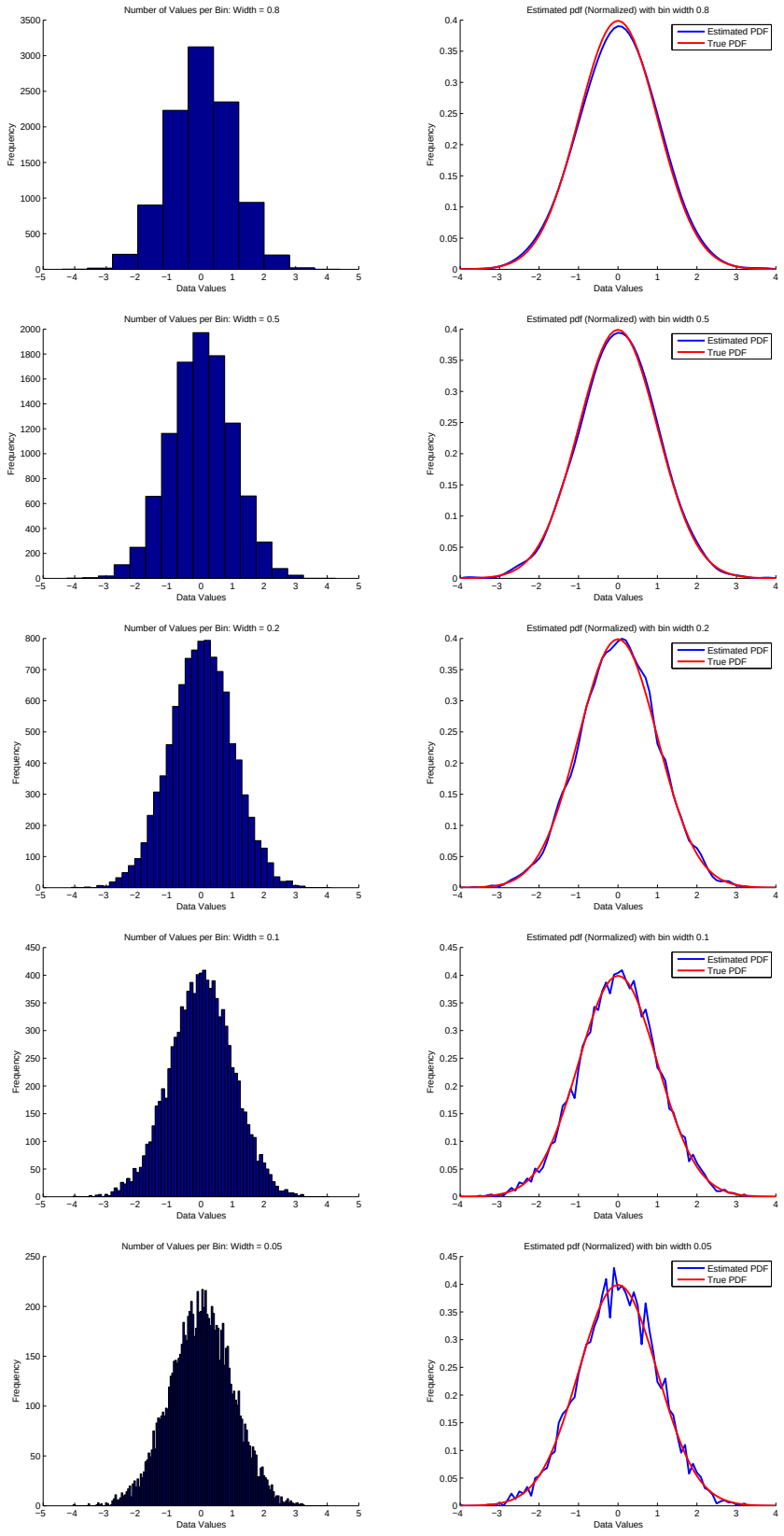


FIGURE 3.5. Binned Data and Estimated PDFs for bin widths of 0.8, 0.5, 0.2, 0.1, and 0.05

work, the true PDF is not known, thus the optimal bin width is not computable in this manner. Typically what is done is that a bin width is estimated based on the number of data points in the data set. For example, Sturges' formula [**74**] says that the optimal number of bins $k$ is given by $k = \lceil \log_2 n + 1 \rceil$, where $n$ is the number of data points. This rule is based on an assumption that the data is fairly normally distributed. For the test case presented above, 10,000 data points were used, thus the optimal number of bins is 15 by this formula. If the bins must cover the range $[-4, 4]$, this results in a bin width of approximately 0.533. The MSE was the smallest for $h = 0.5$, which is close to Sturges' predicted value. Other formulas include one by David Scott in [**69**] and the square root formula. Scott's formula computes the bin width $h$ that minimizes the integrated mean square error as:

$$h = \left( \frac{6}{\int_{-\infty}^{\infty} f'(x)^2 dx} \right)^{1/3} n^{-1/3} \tag{3.11}$$

where $n$ is the number of data points. This of course implies that one knows the PDF $f(x)$, which is typically not the case. A bin width for normally distributed data is given as $h = 3.5\sigma/n^{1/3}$ with $\sigma$ the data's standard deviation. The square root formula is simply $k = \sqrt{n}$ (number of bins equals the square root of the number of data points), and this rule is used in programs such as Excel. The various methods each tend to work better for different types of data sets. With non-parametric PDF estimation, where one does not know if the data conforms to a particular known PDF, the best solution is probably to choose a bin width in the neighborhood of or an average of the suggested bin widths of some of these popular rules.

**3.2.4. Kernel Estimation.** Kernel estimation is another popular PDF estimation method. The definition of the probability distribution function $f(X)$ can be expressed as:

$$f(x) = \lim_{h \to 0} \frac{1}{2h} P(x - h < X < x + h)$$

where $P$ is the probability that the random variable $X$ falls in the range $x-h, x+h$. Kernel estimators try to mimic this idea. The first most intuitive example of a kernel estimator is known as a naive estimator [**72**] and is a discrete version of the above definition.

$$\hat{f}(x) = \frac{1}{2hn}(\text{Number of } X_1, ...X_n \text{ in } (x-h, x+h))$$

where $n$ is the number of data points and $h$ is known as the window width. A small $h$ value is chosen and a simple weight function is defined:

$$w\left(\frac{x-X_i}{h}\right) = \begin{cases} \frac{1}{2} & \text{if } |\frac{x-X_i}{h}| < 1; \\ 0 & \text{otherwise.} \end{cases}$$

The estimated $\hat{f}(x)$ is given by:

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} w\left(\frac{x-X_i}{h}\right)$$

This method eliminates the need to establish pre-determined window (or bin) centers as was the case with histograms. However, the resulting function is still piecewise constant and discontinuous. To alleviate this problem, one can use a different weight function (or kernel) in the summation process:

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)$$

The kernel function $K$ satisfies:

$$\int_{-\infty}^{\infty} K(x)dx = 1$$

and is usually a symmetric probability distribution, such as the normal distribution [**72**]. For the experiment with the given test data, $K$ will be a normal distribution:

$$K\left(\frac{x-X_i}{h}\right) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-X_i}{h}\right)^2}$$

Properties such as continuity and differentiability of the kernel function pass on to $\hat{f}(x)$ so if $K(x)$ is a Gaussian then $\hat{f}(x)$ will be a smooth, continuous and differentiable function. Kernel estimation still has the problem of choice of window width $h$. For $K(x)$ equal to a normal distribution, a small width will produce tall, thin Gaussian curves and will thus create a more spiked PDF that can potentially have spurious data spikes. A wide width creates the opposite problem of over-smoothing and the loss of details. As in the histogram case, there are various methods of choosing a window width and each technique may be optimal for different sets of data. For the case of kernel estimation, assuming that $K$ is symmetric and obeys the following properties:

$$\int K(t)dt = 1 \qquad \int tK(t)dt = 0 \qquad \int t^2K(t)dt = k_2 \neq 0$$

then the window width which minimizes the integrated mean square error can derived as:

$$h = k_2^{-2/5} \left[\int K(t)^2 dt\right]^{1/5} \left[\int f''(x)^2 dt\right]^{-1/5} n^{-1/5}$$

where $n$ is the number of data points and $f''(x)$ is the second derivative of the true PDF [**72**]. As in Equation 3.11, this expression assumes $f(X)$ is known, which in practice is usually not the case. Several techniques have been developed to circumvent this issue. A common choice is to use a standard distribution, (such as a normal distribution) to estimate $\int f''(X)^2$. If a normal distribution is used for $f(X)$ and $K$ is also a normal distribution, the optimal $h$ comes out to approximately:

$$h = 1.06\sigma n^{-1/5}$$

with $\sigma$ the standard deviation of the data [**72**]. Other techniques include least squares cross-validation and likelihood cross-validation [**72**]. Programs such as MATLAB and Python with built-in kernel estimation functions utilize these types of procedures to choose a window width for inputted data sets.

| Window Width | Error $E((f(x) - \hat{f}(x))^2)$ |
|---|---|
| 0.5 | 2.983e-4 |
| 0.2 | 1.896e-5 |
| 0.1 | 2.345e-5 |
| 0.05 | 5.110e-5 |

TABLE 7. Mean Square Errors between the true PDF and the estimated PDFs from kernel estimation with different window widths

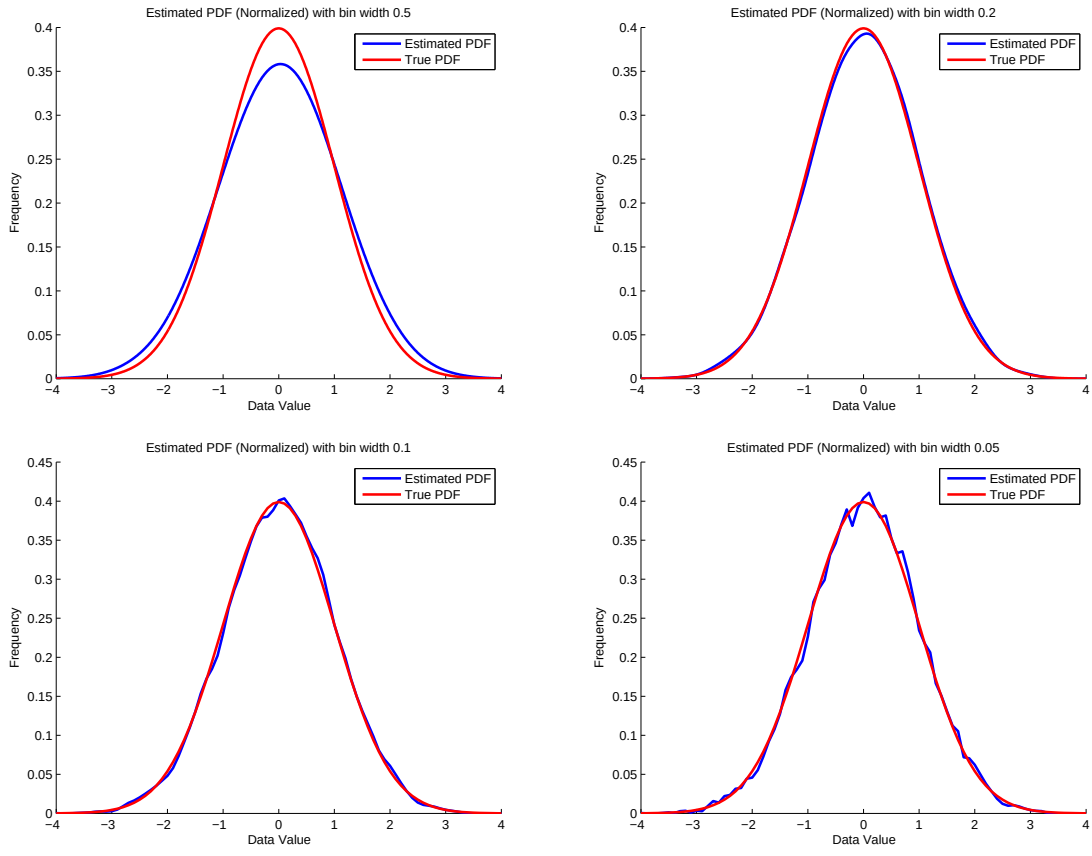


FIGURE 3.6. Estimated PDFs from data set using kernel estimation with window widths of h=0.5, 0.2, 0.1 and 0.05

A kernel estimator based on a Gaussian kernel should perform well at estimating the PDF of data that has been generated from a normal distribution (as in the given data set). Estimated PDFs of varying window widths are shown in Figure 3.6. The errors between the estimated and true PDFs are given in Table 7. As in the histogram PDF estimation case, the errors are on the order of $10^{-5}$.

**3.2.5. Conclusions.** Out of the four methods presented here, the two which perform best (in terms of lowest MSE) were found to be histogram interpolation and kernel estimation. These two methods will be utilized for PDF estimation of microscopic data in the cytoskeletal model.

**3.2.6. Data Regeneration.** Once an approximate PDF has been found, it will be used to generate a new set of data. This is done by creating the cumulative distribution function (CDF), which is simply the integral of the PDF. Since the PDF has been normalized during the estimation process, the CDF values will range from 0 to 1. A uniform random number generator is used to choose a value between 0 and 1 and is designated as the ordinate number in a coordinate pair. The abscissa value that corresponds to the this ordinate value is the data point being sought for the new data set. In other words, let $g(x)$ be the cumulative distribution function. Given a $y$ value where $y = g(x)$, the goal is to find the $x$ value that produces this $y$. This problem is solved via an inverse interpolation scheme. The steps are depicted in Figure 3.7

## 3.3. Thermodynamics and Statistical Mechanics

The theory presented thus far will be utilized for the construction of PDFs and the regeneration of data at the microscopic scale. To create a consistent continuum-microscopic model, the microscopic data must not only conform to the predicted PDFs, but also to macroscopic, continuum level constraints. This is where concepts from statistical mechanics and thermodynamics become useful.

Statistical mechanics is a branch of science that seeks to understand the mechanics of a continuum body by examining the behavior of its microscopic components [**64**]. It is closely related to the field of thermodynamics, which studies macroscopic parameters (e.g. energy, volume, pressure, temperature) and the relationships between these variables, which characterize a system of microscopic elements. An important concept introduced in thermodynamics is that of entropy. In this field, entropy is presented as an abstract, variational function that must be maximized in order for a system to be in thermodynamic

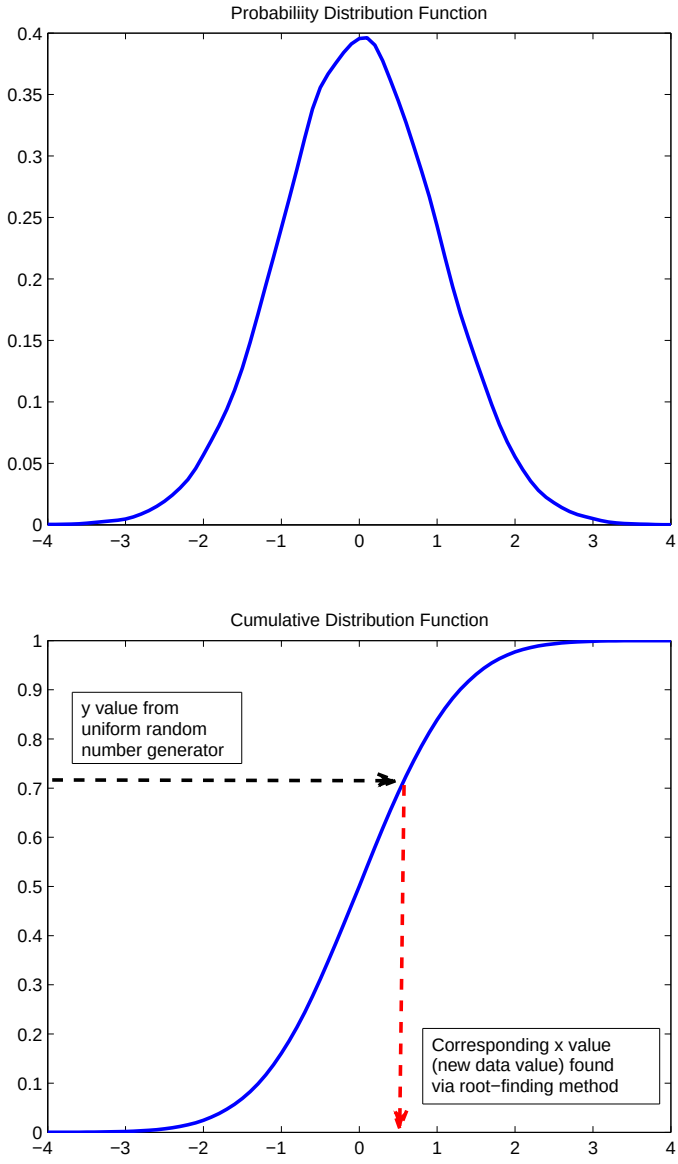


FIGURE 3.7. Estimated PDF for a data set and its corresponding CDF. A new set of data is generated from the PDF by using a random number generator to produce a $y$ value that is then matched to an x value in the CDF via inverse interpolation.

equilibrium. It is statistical mechanics that provides physical meaning to this abstract idea [**14**].

Finding equilibrium states of a system given a set of constraints is one of the main goals of the instantiation procedure in the continuum-microscopic model of the cytoskeleton. The solution of the continuum level equations will provide a macroscopic description

of the microscopic system. Such information will include the total energy of the system, the average strains, the number of microscopic elements, and also the volume of the system. These conditions must be met by the microscopic system in order to maintain model consistency. This requirement creates the need to formulate and solve a fundamental thermodynamics/statistical mechanics problem.

Basic concepts from thermodynamics and statistical mechanics will now be presented as background information that will lay the groundwork for portions of the algorithm developed in this thesis.

**3.3.1. Thermodynamics.** All systems have a tendency to move toward equilibrium states, which are states governed by the internal properties of the system without any dependence on past external influences [**14**]. The first postulate of thermodynamics puts the above statement more formally as:

**Postulate I**: There exists particular states (called equilibrium states) of simple systems that, macroscopically, are characterized completely by the internal energy $U$, the volume $V$, and the mole numbers $N_1, N_2, ....N_r$ of the chemical components.

More constraints can be added to this list if the system must satisfy additional conditions. The determination of these equilibrium states is the central problem of thermodynamics [**14**]. The solution to this problem is presented in the form of an extremum principle, and is where the concept of entropy is introduced:

**Postulate II**: There exists a function (called the entropy $S$) of the extensive parameters of any composite system, defined for all equilibrium states and having the following property: The values assumed by the extensive parameters in the absence of an internal constraint are those that maximize the entropy over the manifold of constrained equilibrium states.

This function $S$ written in terms of the extensive parameters $U$,$V$, and $N$ is called the fundamental relation. If an expression for the entropy $S$ can be found, the state of

maximum entropy can be determined by finding the values for the variables which satisfy $dS = 0$.

There is third postulate regarding the entropy which provides some mathematical rules about the function $S$:

**Postulate III**: The entropy of a composite system is additive over the constituent subsystems. The entropy is continuous and differentiable and is a monotonically increasing function of the energy.

The continuity and differentiability of the entropy function will allow partial derivatives to be taken in order to compote the maximum. The additivity will be important later for the construction of the entropy function in the statistical mechanics framework.

The entropy can be written as a function $S(U, V, N)$, and by postulate III, this fundamental relation can be inverted as $U(S, V, N)$, energy as a function of entropy and the other extensive parameters. Because of this relationship, an equivalent extremum principle can be stated for the energy. In the entropy representation, the system is in its equilibrium state when given total energy $U$, the entropy $S$ is maximized. In the energy formulation, the system is in equilibrium when given entropy $S$, the energy $U$ is minimized. These two principles both arrive at the same resulting equilibrium state. The fact that thermodynamic equilibrium is associated with minimal energy makes intuitive sense. The theory of mechanics states that a system is in a state of stable mechanical equilibrium when its potential energy is minimized. For a system to be in thermodynamic equilibrium, it must be in mechanical equilibrium and also thermal equilibrium. The minimum energy principle presented above is an extension of the mechanical theory of minimal energy that includes thermal effects. Both extremum principles are valid, and the choice to use the entropy or the energy formulation is dependent on the problem at hand.

Given the fundamental relation: $U(S, V, N)$, its differential $dU$ can be written as:

$$dU = \frac{\partial U}{\partial S} dS + \frac{\partial U}{\partial V} dV + \frac{\partial U}{\partial N} dN \tag{3.12}$$

The partial derivatives in 3.12 arise often in the discussion of thermodynamics and are given particular names:

$$\begin{aligned} \frac{\partial U}{\partial S} &= T \quad temperature \\ -\frac{\partial U}{\partial V} &= P \quad pressure \\ \frac{\partial U}{\partial N} &= \mu \quad electrochemical\ potential \end{aligned}$$

Collectively, these variables are known as the intensive parameters. The partial derivatives of the entropy formulation can be written in terms of the intensive parameters as:

$$\frac{\partial S}{\partial U} = \frac{1}{T}, \quad \frac{\partial S}{\partial V} = \frac{-P}{T}, \quad \frac{\partial S}{\partial N} = -\frac{\mu}{T}$$

One way to find the entropy function $S$ is to first determine $dS$ and then integrate. The function $dS$ can be constructed by utilizing known relationships between extensive and intensive parameters for a given system.

**3.3.2. Statistical Mechanics.** Statistical mechanics provides a different interpretation of the entropy concept presented above. This new description of entropy also provides another method for constructing the $S$ function, which is often more convenient then trying to determine the necessary relationships to carry out the method described above.

Suppose that a particular macroscopic system composed of many microscopic elements must have total energy $U$, volume $V$ and total number of microscopic elements $N$. It is reasonable to assume that there will be many possible configurations of the microscopic elements that will fulfill these three constraints. Statistical mechanics assumes that the system experiences rapid transitions between permissible states, induced by the interaction of this system with external random processes. A fundamental notion in statistical mechanics is that the system has equal probability of being in any of its permissible states. The total number of possible states $\Omega$ is the maximum number of states that fulfill the constraints. If a constraint is added or removed, $\Omega$ will adjust to a new maximum value. Entropy was also described as a value that is maximized under

a given set of constraints, thus this suggests that $\Omega$ and $S$ should be related. The entropy is additive and $\Omega$ is multiplicative, thus the unique function that relates these two quantities and fulfills both criteria is:

$$S = k_B \ln \Omega$$

where $k_B$ is a constant for scaling purposes. In order for this definition to coincide with the Kelvin scale of temperature, $k_B$ is chosen to be the Boltzmann constant. This formulation of the entropy in terms of the number of permissible states is known as the microcanonical formalism.

This constitutes a different way of determining the function $S(U, V, N)$. If the number of possible states is countable, then one can find an expression for $\Omega$ in terms of $U, V, N$ and in turn have an expression for $S$. A simple two-state system example of this method will now be described.

Begin with a system of $N$ atoms. Each atom can be either in a ground state of zero energy or an excited state of energy $\epsilon$. The total energy of the system is $U$, therefore $U/\epsilon$ atoms are in the excited state and $N - U/\epsilon$ atoms are in the ground state. How many different ways are there to choose $U/\epsilon$ atoms from the total $N$?

$$\Omega = \frac{N!}{U/\epsilon!(N - U/\epsilon)!}$$

Thus the entropy can be written as:

$$\begin{aligned} S &= k_B \ln \left( \frac{N!}{U/\epsilon!(N - U/\epsilon)!} \right) \\ S &= k_B \ln(N!) - k_B \ln \left( \frac{U}{\epsilon}! \right) - k_B \ln \left[ \left( N - \frac{U}{\epsilon} \right)! \right] \end{aligned}$$

Assuming that the quantities within the logarithms are large, Stirling'sapproximation for the logarithm of a factorial:

$$\ln(M!) \approx M \ln(M) - M$$

can be used to rewrite the entropy function as:

$$S = k_B \left[ N \ln(N) - \frac{U}{\epsilon} \ln\left(\frac{U}{\epsilon}\right) - \left(N - \frac{U}{\epsilon}\right) \ln\left(N - \frac{U}{\epsilon}\right) \right] \tag{3.13}$$

This is the fundamental relation for this particular system, written in terms of its extensive parameters, constructed by counting its total number of permissible states. Expressions for the extensive parameters can be found in terms of the intensive parameters $T$ and $\mu$ by taking the partial derivatives of 3.13 and solving for $U$ and $N$.

In the next chapter, more simple examples of entropy function construction and maximization will be presented within the context of the cytoskeleton. This concludes the theory portion of this thesis. The next chapter will describe the development of a new continuum-microscopic model of the cytoskeleton.

CHAPTER 4

# Motivating Examples

In the previous chapter, concepts from probability theory and statistical mechanics pertaining to the development of the new model were presented. These concepts will now be put to practical use with some examples that demonstrate: (1) the necessity of PDF estimation and extrapolation of the microscopic data, (2) how macroscopic constraints can be incorporated into the microscopic instantiation procedure to maintain model consistency.

## 4.1. Microscopic Distributions of Network Variables

As mentioned in Chapter 1, other continuum-microscopic algorithms perform Step 1 of the general algorithm in Section 2.5 by utilizing known or presumed PDFs. With simpler systems this may be a viable option, but with the cytoskeleton, the distributions of filament angles and strains are unknown and varying with space and time. PDF estimation will be necessary in order to determine an underlying distribution. These distributions will be collected over time to understand how they evolve, in order to make predictions on the shapes of the PDFs at future instantiation steps. As evidence of the complexity of the distributions, a one block example of a cytoskeletal network will now be presented. This example will also be used to demonstrate how changes in the microstructure due to deformation can alter continuum-level quantities such as elasticity moduli.

**4.1.1. Network Construction and Deformation.** Begin with a three-dimensional block with dimensions: $[x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$. At the continuum-level,

this cube is viewed as a portion of continuous media, but at the microscopic level this block is a network of crosslinked cytoskeletal filaments (Figure 4.1).

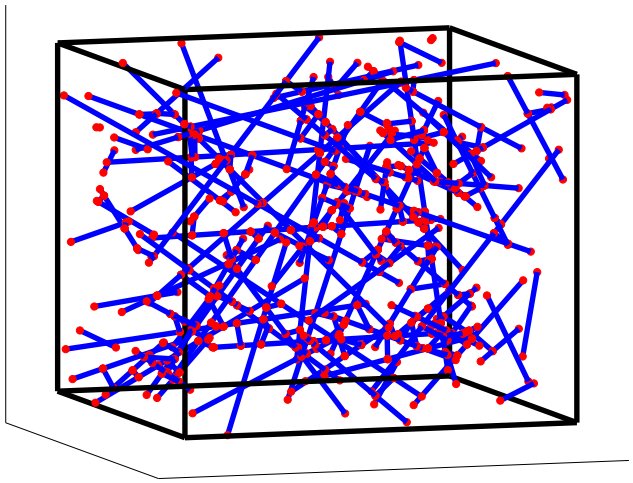

FIGURE 4.1. A three-dimensional microscopic network of crosslinked fibers.

To set up this initial network, $n$ filaments were laid in the block. This was done by placing one endpoint $(x_0, y_0, z_0)$ inside the block: $x_{min} \leq x_0 \leq x_{max}, y_{min} \leq y_0 \leq y_{max}, z_{min} \leq z_0 \leq z_{max}$. The second endpoint $(x_1, y_1, z_1)$ is assigned by choosing two things: (1) a length $t$ for the filament using a Gaussian distribution centered around a mean length $t_0$, (2) a direction vector $\langle x_{dir}, y_{dir}, z_{dir} \rangle$ established by choosing $x_{dir} \in [-1, 1], y_{dir} \in [-1, 1], z_{dir} \in [-1, 1]$ via uniform random number generator, and then normalizing the vector. The second endpoint is thus computed by:

$$x_1 = x_0 + t x_{dir}$$

$$y_1 = y_0 + t y_{dir}$$

$$z_1 = z_0 + t z_{dir}$$

If this endpoint falls outside the block, the endpoint is revised to be the intersection point of that filament with the box wall through which it crosses. This filament is flagged as attached to the wall, which will be important for when the block is strained (to be

explained shortly). The $n$ filaments are assumed to be at their equilibrium lengths in this initial state, however strains could be assigned to each filament if desired.

Crosslinks between filaments must be established next. This is done by taking each filament pair and finding the shortest distance between them. This is done using geometric relationships. Let:

$$\mathbf{p} = \mathbf{p}_0 + t\mathbf{e}_1$$

$$\mathbf{q} = \mathbf{q}_0 + s\mathbf{e}_2$$

be the equations for two filament lines with $\mathbf{p}_0, \mathbf{q}_0$ the starting points on each filament, $\mathbf{e}_1, \mathbf{e}_2$ direction vectors, and $t, s$ lengths along those lines. The distance between any two points on these lines is given by the length of the following vector:

$$\mathbf{r} = (\mathbf{p}_0 - \mathbf{q}_0) + t\mathbf{e}_1 - s\mathbf{e}_2$$

The shortest $\mathbf{r}$ is the one that is perpendicular to both original lines. This gives the following two equations:

$$0 = ((\mathbf{p}_0 - \mathbf{q}_0) \cdot \mathbf{e}_1) + t - s(\mathbf{e}_2 \cdot \mathbf{e}_1)$$

$$0 = ((\mathbf{p}_0 - \mathbf{q}_0) \cdot \mathbf{e}_2) + t(\mathbf{e}_1 \cdot \mathbf{e}_2) - s$$

These two equations can be solved for $t$ and $s$, which will give the points on each line where the shortest distance occurs. The length of this distance can be computed using the distance formula. If this distance is below a certain threshold value, then a crosslink has a probability $P$ of forming. A random number generator is used to pick a value $x \in [0, 1]$. If $x < P$ a crosslink forms and is added to the network as a new short fiber connecting the two original filaments. The two filaments are broken down into four segments (see Figure 4.2).

This completes the initial construction of the network. To demonstrate how the distributions of variables change as a result of deformation, the block will be placed


FIGURE 4.2. Establishing a crosslink between two filaments.

under an extensional strain. During one deformation step, the two walls of the block with equations of plane $x = x_{min}$ and $x = x_{max}$ will be moved outward a distance $\Delta x$ so that these walls will now be defined by the two equations $x = x_{min} - \Delta x$ and $x = x_{max} + \Delta x$. In a typical medium, the other four walls defined as $y = y_{min}$, $y = y_{max}$, $z = z_{min}$ and $z = z_{max}$ would likely move inward (compress) to accommodate the extension in the other direction. For this simple example, the assumption will be made that this does not occur, and the four walls remain defined by the above plane equations. The only thing that changes is that their area increases (see Figure 4.3).

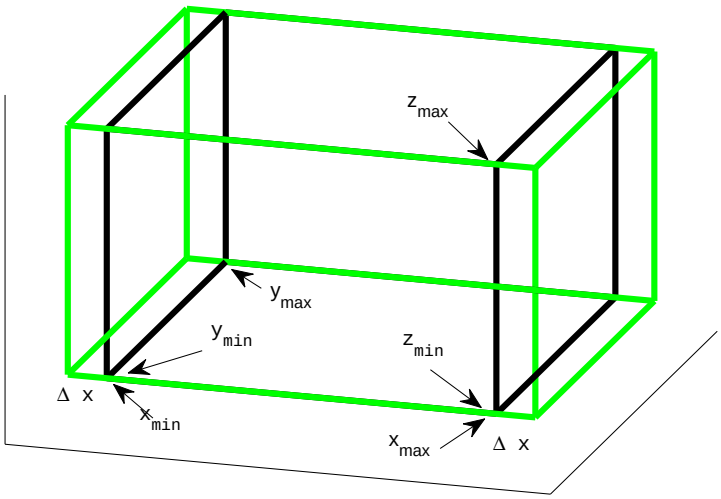


FIGURE 4.3. Original block (in black) under an extensional strain (in green) in the $x$ direction.

Any filaments attached to the six walls move with those walls during the deformation. The filament segments and crosslinks in the box's interior are moved in another manner.

As described in Section 2.3 one way to move all of the interior points is to utilize Newton's Law, $F = ma$ to set up a large system of ordinary differential equations. Maintaining numerical stability in such a system forces one to use very small time steps for an explicit method or to find the solution of a very large linear system in an implicit method. An alternative approach (also mentioned in Section 2.3) is to recast the problem as one of energy minimization. Systems tend to move towards equilibrium states, and as explained in the thermodynamics section in Chapter 3 these equilibrium states occur when the entropy is maximized or equivalently when the energy is minimized. The cytoskeletal network will try to reorient itself into a position of minimal energy, and this is how the internal filaments and crosslinks will be moved in this algorithm. As a starting point, the internal filaments and crosslinks are initially moved via a linear mapping that maps points in the original block to points in the extended block (see Appendix A for mapping details). These assigned positions are likely not a minimal energy configuration, but are simply an initial guess with which to start an energy minimization procedure. In the full algorithm, there will be macroscopic constraints on the system (demonstrated in the next section's examples), but for this case the goal will simply be to rearrange the filaments and crosslinks to minimize the potential energy. Filament segments and crosslinks are modeled as springs, thus the total potential energy is given by:

$$Energy = \sum_{j=1}^{m} \left[ \frac{k_j}{2} (L^j - L_0^j)^2 \right] \tag{4.1}$$

where $k_j$ is the spring constant and $L^j$ and $L_0^j$ are the current and equilibrium lengths respectively of the $j^{th}$ filament or crosslink segment. The equilibrium length $L_0^j$ of each segment was established by computing the length of the segment in its initial state. The current length $L^j$ is found via a distance formula:

$$L^j = \sqrt{(x_1^j - x_0^j)^2 + (y_1^j - y_0^j)^2 + (z_1^j - z_0^j)^2}$$

where the subscripts 1 and 0 represent the two ends of the filament or crosslink segment. The values $x_1^j, y_1^j, z_1^j, x_0^j, y_0^j, z_0^j$ are the variables of the energy function in 5.5. A necessary

condition for a minimum of 5.5 is that the variables satisfy:

$$\partial E/\partial x_1^j = 0 \quad \partial E/\partial x_0^j = 0$$

$$\partial E/\partial y_1^j = 0 \quad \partial E/\partial y_0^j = 0$$

$$\partial E/\partial z_1^j = 0 \quad \partial E/\partial z_0^j = 0$$

for all internal filament and crosslink segments $j$. This is a large nonlinear system of equations. An iterative, gradient search algorithm is employed to find a solution (see Appendix A for algorithm details). Once the minimum has been found, the next deformation step can take place and the process is repeated.

How one computes elasticity moduli of the block that can then be used in the macroscopic equations is not pertinent for demonstrating the complexity of microscopic distributions. However it will be presented here briefly (details in Chapter 6 and Appendix A) for completeness since an analogue to this procedure will be used in the full continuum-microscopic model. In this simple extensional example, the Young's modulus of the block can be computed by using Hooke's Law: $\sigma_{xx} = E\epsilon_{xx}$ where $\sigma_{xx}$ is the stress in the $x$ direction on the block face with normal vector in the $x$ direction, $E$ is the sought after Young's modulus and $\epsilon_{xx}$ is the block's extensional strain in the $x$ direction. The stress $\sigma_{xx}$ can be found by first computing the total force $\mathbf{F}$ in the $x$ direction on the two $x$ walls of the block (walls with original equations $x = x_{min}$ and $x = x_{max}$). The $x$ component of $\mathbf{F}$, denoted $F_x$, can be found by summing the Hookean forces (in the $x$ direction) of all filament segments (total of $m$) that are attached to these two walls.

$$F_x = \sum_{j=1}^{m} -k_j(L^j - L_0^j)x_{dir}^j$$

where $x_{dir}^j$ is the $x$ component of filament segment $j$'s direction vector. $F_x$ can then be recast as $\sigma_{xx}$ by dividing by the area of one of the $x$ walls. The extensional strain $\epsilon_{xx}$ of the block (after the first deformation step) is $\frac{2\Delta x}{x_{max}-x_{min}}$. The Young's modulus $E$ can

then be computed by $E = \sigma_{xx}/\epsilon_{xx}$. For a more detailed presentation of how to find all of the elasticity moduli from the microscopic data see Appendix A.

**4.1.2. Network Behavior.** With the explanation of network construction and deformation complete, attention can now be focused on the network's behavior during the deformation.

The original network was assigned filament orientations from a uniform distribution and zero strains. Does the Young's modulus remain constant over successive deformation steps? If the answer is "No", then this provides evidence for the need for a model of the cytoskeleton that utilizes the response of its microscopic network to update macroscopic parameters. Secondly, do the filament orientations remain uniformly distributed and the filament strains remain zero as the network deforms? If the answer to this question is also "No", this motivates the need to keep track of how these distributions are changing so that Step 1 of the general continuum-microscopic algorithm can be carried out using predicted PDFs as opposed to an assumed family of distributions like those used in the initial construction.

The first task is to track the Young's modulus for this block of filaments as it undergoes several extensional strain steps as outlined above. Intuitively, as the block is stretched further in the $x$ direction, the filaments should tend to align themselves in the direction of strain to minimize their stored energy. This creates a more parallel set of fibers, which should increase the Young's modulus of the block. Thus as strain increases, the expectation is that the Young's modulus will also increase, (a demonstration of strain hardening). Figure 4.4 shows a plot of the calculated Young's modulus of the block versus the strain. These results are corroborated by simulation results from Åström et al (see Figure 1(a) in [**6**]) and also with results from the model of a red blood cell cytoskeleton in Hansen et al. (see Figure 19 in [**33**]). This result has also been shown experimentally by several groups: Chaudhuri et al. in dendritic actin networks (Figure 3(a) [**20**]), Gardel et al. in *in vitro* F-actin networks (Figure 1(b) [**29**]), and by Xu et al. in $\alpha$-actinin crosslinked F-actin networks (Figure 1 [**80**]).

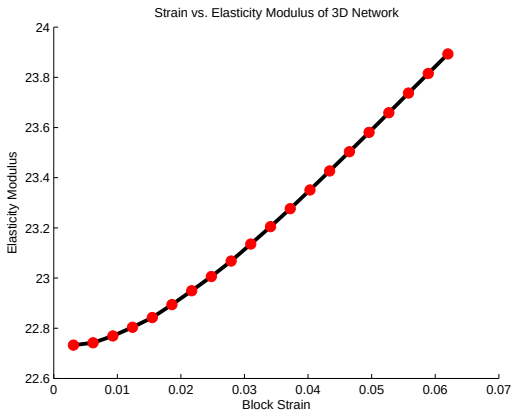


FIGURE 4.4. The Young's modulus vs. strain curve for a box of crosslinked filaments under extensional strain

Data was also collected on the angles each filament makes with the $x$ axis (axis of strain). As the strain grows in the $x$ direction, the average angle decreases as expected (Figure 4.5). Also in Figure 4.5 is a a plot of the average filament strain versus the overall strain. As the block is stretched, the aveage strain of the filaments also increases.

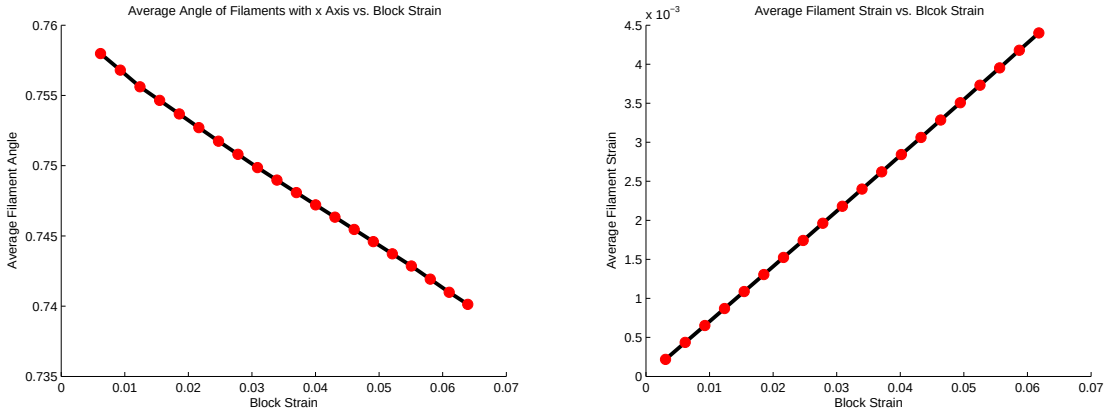


FIGURE 4.5. Average angle of orientation of the filaments with respect to the axis of strain versus the overall block strain. Average filament strain versus the overall block strain

The second question posed at the beginning of this section is in reference to the distribution of microscopic variables, such as filament orientations and strains. To exemplify that the microscopic data does not conform to a known distribution, a histogram of the distribution of orientation angles of the filaments after one deformation step is shown in

Figure 4.6 along with an estimated PDF and a reference uniform PDF. The initial orientations were generated from a uniform distribution. After one deformation step the mean squared error between the estimated PDF and a uniform PDF is found to be $\approx 0.0173$.

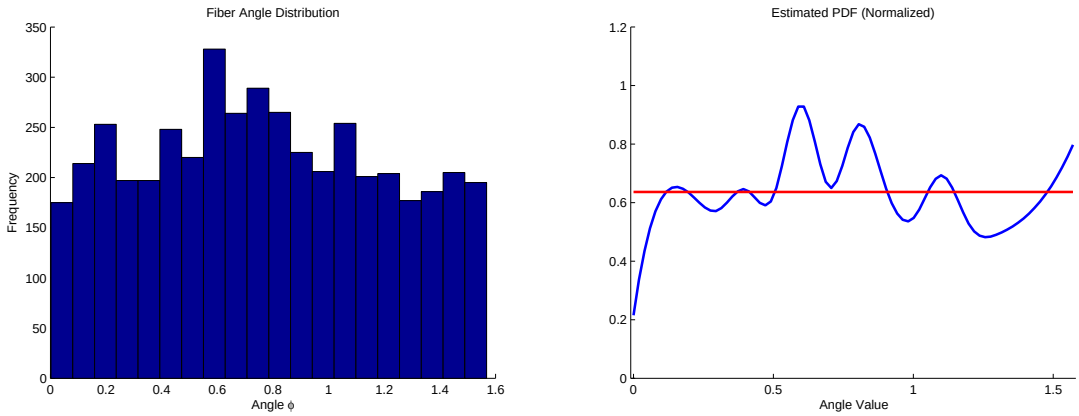


FIGURE 4.6. A histogram of filament orientation angles near the beginning of the simulation, and the data's estimated PDF computed via histogram interpolation.

The strain of each filament segment after the first deformation step was also collected, and this data and estimated PDF are shown in Figure 4.7. The filament strains do not remain zero nor are they a constant value or uniformly distributed.

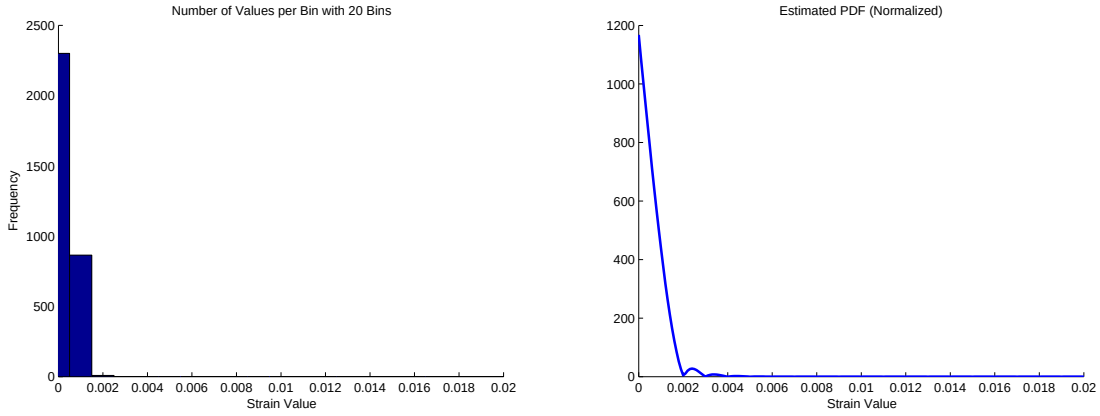


FIGURE 4.7. A histogram of filament orientation angles near the beginning of the simulation, and the data's estimated PDF computed via histogram interpolation.

To demonstrate that these distributions are also time-varying, the orientation angle that each filament makes with the axis of strain and the filament strains were collected

at a later point in time when the extensional strain is much larger. The distributions and estimated PDFs are shown in Figure 4.8. It is clear from the histogram and PDF that the average angle has shifted to a smaller value. As explained before, this is to be expected since the filaments will tend to align themselves in the direction of strain. Also the strains become more varied as the overall strain increases. The estimated angle PDF has an MSE of 0.0198 compared to the uniform distribution, thus the angles are moving further away from a uniform distribution as the extension increases.

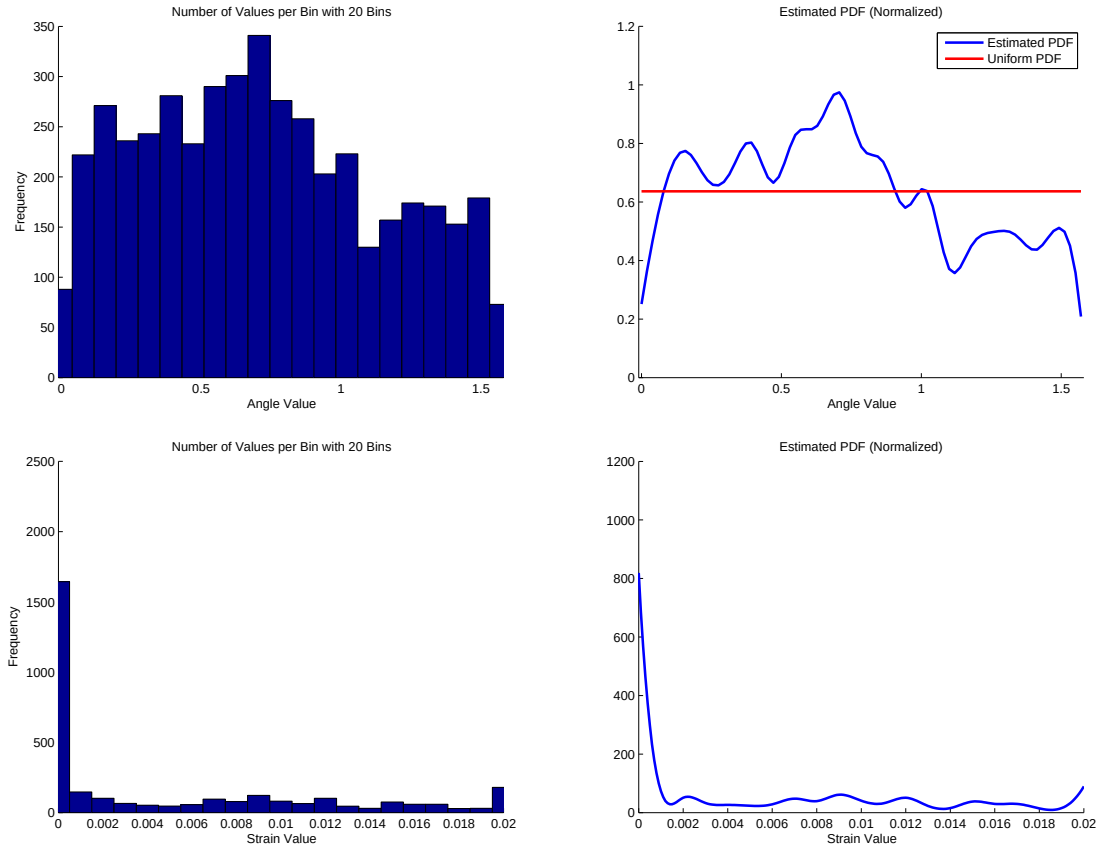


FIGURE 4.8. Top row: the distribution of orientation angles and estimated PDF after a large extensional strain. Bottom row: the distribution of strains and estimated PDF for the same network.

This one block example was presented to solidify the idea that changes occurring at the microscopic level do impact variables (such as elasticity moduli) at the macroscopic scale. This example also demonstrates that the distributions of microscopic variables change over time and do not conform to particular, known families of distributions.

During the microscopic reinstantiation phase of the continuum-microscopic algorithm, in order to produce a network that represents a state into which the original network could have evolved if it had been advanced over the full time interval, the evolution of these distributions must be tracked over time to extrapolate predictions for their shapes at future time steps.

## 4.2. Macroscopic Constraints

The examples presented above deal with the goal of retaining memory of the cytoskeletal microstructure over continuum time steps and utilizing the evolution of the microstructure during micro-time steps to update continuum-level parameters. A second, equally important goal is that the reinstantiated microstructure should fulfill a set of macroscopic parameters and constraints that have been provided by continuum level equations. This will keep the continuum-microscopic model consistent. The microscopic computation finds a new set of elasticity moduli for each continuum grid cell that is then used in the advancement of the continuum level elasticity equations. This advancement produces new values for the positions of each grid cell, as well as new stress, strain, and energy values. The microscopic network created during the next step should embody the same macroscopic parameters as its overlying continuum grid cell. This new system should also be in a state of mechanical equilibrium since natural systems such as the cytoskeleton will tend towards a state of minimal potential energy. These two requirements can both be met by solving the fundamental thermodynamics problem presented in the last chapter. This problem should be solved in the entropy representation since an energy constraint will be provided. To illustrate this approach, several examples will now be presented. The statistical mechanics representation of the entropy will be used to construct the fundamental relations.

**4.2.1. Entropy Formulation. Example 1**: As an elementary example, consider a one-dimensional continuum segment of length $dX$ which represents a segment of cytoskeleton. Microscopically, this segment is composed of bundled actin filaments, with the bundles

connected end to end in this simple case. Suppose in this segment $dX$, there must be $N$ total filaments, and for simplicity these $N$ filaments must be grouped into two bundles. Each bundle is attached with one end to one of the edges of the $dX$ segment, and the other end is attached to the other bundle somewhere in $dX$ (Figure 4.9).

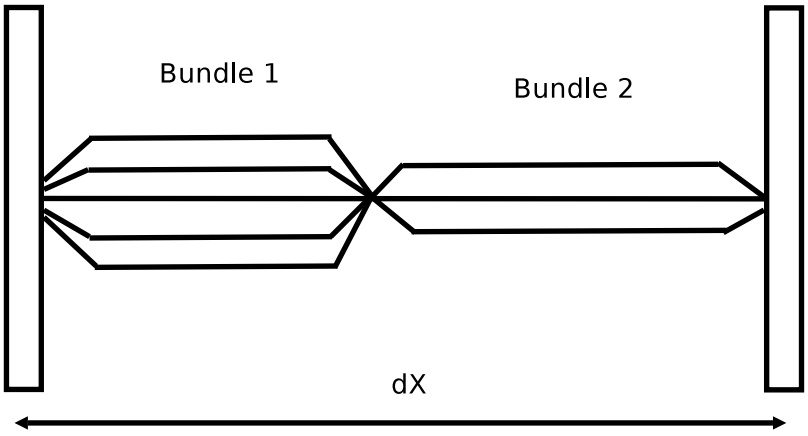


FIGURE 4.9. Segment of continuum cytoskeleton, represented by bundled filaments

A bundle is defined as containing at least one filament (this forces $dX$ to not have any gaps). Suppose again for simplicity that these filaments each have an equilibrium length of $L_0$. The filaments in the first bundle have a current length of $C_1$ and the filaments in second bundle have current lengths of $C_2$ where $C_1 + C_2 = C = dX$. The energy of the system $U$ is also given. This energy can be written as:

$$U = kN_1(C_1 - L_0)^2 + kN_2(C_2 - L_0)^2 \tag{4.2}$$

where $k$ is the spring constant for one filament, and $N_1$ and $N_2$ are the number of filaments in bundle 1 and bundle 2 respectively. Besides relation 4.2, the following must also be true:

$$\begin{aligned} N_1 + N_2 &= N \\ C_1 + C_2 &= C \end{aligned}$$

which allows $U$ to be rewritten as:

$$U = kN_1(C_1 - L_0)^2 + k(N - N_1)(C - C_1 - L_0)^2 \tag{4.3}$$

How many states have the given energy $U$ and have $N$ filaments divided into two bundles whose total length must be $C$? The values $N_1$ and $N_2$ must be integer valued since they denote the number of filaments in each bundle. There are $N - 1$ possible choices for $N_1$. (It is not $N$ because $N_2$ must be at least 1). Once $N_1$ is chosen, $N_2$ is automatically known. Also, once $N_1$ is chosen, $C_1$ is also determined because it must take on a value that satisfies the energy constraint 4.3. With $C_1$ known, $C_2$ is known as well, and thus all variables have been accounted for. The total number of possible states is therefore $N - 1$. The entropy is simply:

$$S = k_B \ln(N - 1) \tag{4.4}$$

The goal is to find an equilibrium state for this system, or a state where the entropy is maximized. The variables to be determined are $N_1, N_2, C_1, C_2$ so 4.4 should be written in terms of these variables. This can be done by utilizing 4.3 to solve for $N$ and replace it in 4.4:

$$S = k_B \ln\left(\frac{U - kN_1(C_1 - L_0)^2 + kN_1(C - C_1 - L_0)^2}{k(C - C_1 - L_0)^2} - 1\right)$$

To find a maximum value for $S$, take its partial derivatives with respect to $N_1$ and $C_1$, set these quantities equal to zero and solve for $N_1$ and $C_1$.

$$\begin{aligned}
\frac{\partial S}{\partial N_1} &= k_B \frac{k(C - C_1 - L_0)^2 - k(C_1 - L_0)^2}{U - kN_1(C_1 - L_0)^2 + kN_1(C - C_1 - L_0)^2} \\
\frac{\partial S}{\partial C_1} &= k_B \left[\frac{-2k(N_1 - 1)(C - C_1 - L_0) - 2kN_1(C_1 - L_0)}{U - kN_1(C_1 - L_0)^2 + kN_1(C - C_1 - L_0)^2} + \frac{2}{(C - C_1 - L_0)}\right]
\end{aligned}$$

Setting these two equations equal to zero and solving for $N_1$ and $C_1$ gives:

$$N_1 = N/2, \qquad C_1 = C/2$$

and means that $N_2 = N_1$ and $C_2 = C_1$. This result matches physical intuition. If both bundles have equilibrium length $L_0$ then the total energy of the system will be minimized if the bundles are the same length and contain the same number of filaments.

**Example 2**: Consider a similar situation as in Example 1, except let there be three filament bundles instad of two. In this case the energy can be written as:

$$U = kN_1(C_1 - L_0)^2 + kN_2(C_2 - L_0)^2 + kN_3(C_3 - L_0)^2$$

The following constraints also apply:

$$N_1 + N_2 + N_3 = N$$

$$C_1 + C_2 + C_3 = C$$

This allows a reduction in the number of variables found in $U$:

$$U = kN_1(C_1 - L_0)^2 + kN_2(C_2 - L_0)^2 + k(N - N_1 - N_2)(C - C_1 - C_2 - L_0)^2 \tag{4.5}$$

The number of possible states that conform to these constraints will be more complicated than in Example 1 since there are more variables. There are $N$ filaments to place in 3 bundles, and the number of ways this can be done is:

$$\frac{(N+2)!}{N!2!} = \frac{(N+2)(N+1)}{2} \tag{4.6}$$

However, this includes cases where some bundles can be empty, which are not permissible states for the system. There are $3(N-1)$ cases with one empty bucket and 3 cases with two empty buckets, so these need to be subtracted off of 4.6:

$$\frac{(N+2)(N+1)}{2} - 3(N-1) - 3$$

$$\frac{(N-1)(N-2)}{2}$$

This is the number of ways the filaments can be arranged into three bundles, and establishes the possibilities for $N_1, N_2, N_3$. The energy constraint in 4.5 still has the two unknowns $C_1, C_2$. Once the number of possibilities for one of these is known, the other will follow automatically from the satisfaction of 4.5. Length is usually a real-valued quantity, but to make things countable here, suppose that $C_1$ can take on only integer values from 1 to $C-2$ (assuming that $C_2$ and $C_3$ must be at least one unit long each, but they can be real-valued). As mentioned in the theory discussion, the number of permissible states $\Omega$ is multiplicative so:

$$\Omega = \frac{(N-1)(N-2)(C-2)}{2}$$

The entropy can then be written as:

$$S = k_B \ln\left(\frac{(N-1)(N-2)(C-2)}{2}\right) \tag{4.7}$$

As in Example 1, the goal is find values for the variables $N_1, N_2, N_3, C_1, C_2, C_3$ that maximize the entropy. Equation 4.5 can be solved for $N$ and substituted into 4.7 to write $S$ in terms of the desired variables. The same process can then be carried out to find what values maximize $S$. This algorithm was carried out using Mathematica and the results were:

$$C_1 = C_2 = C_3 = C/3, \qquad N_1 = N_2 = N_3 = N/3$$

This again matches intuition, as the analogue to the two filament bundle case.

**Example 3** Example 1 can be extended in another manner, by allowing the equilibrium length $L_0$ to vary. Suppose that $L$ is the equilibrium length for segment $dX$ and that $L_0$ can take on two possible values: $L/4$ or $3L/4$. Since the value of the spring constant is based on the spring's equilibrium length, $k$ will now be a function of $L_0$ for each bundle.

The energy for this system can be written as:

$$U = \frac{KL}{L_0} N_1 (C_1 - L_0)^2 + \frac{KL}{L - L_0} (N - N_1)(C - C_1 - L + L_0)^2 \tag{4.8}$$

with $K$ the spring constant of a filament of length $L$. The number of possible states for this system is given by $(N-1)$ (the possibilities for $N_1$) multiplied by 2 (the possibilities for $L_0$). The entropy is given as:

$$S = k_B \ln(2(N-1))$$

Using 4.8 to replace $N$, and taking partial derivatives with respect to $N_1$ and $C_1$ as in Example 1, yields:

$$\begin{aligned}
\frac{\partial S}{\partial N_1} &= k_B \frac{\frac{KL}{L-L_0}(C - C_1 - L + L_0)^2 - \frac{KL}{L_0}(C_1 - L_0)^2}{U - \frac{KL}{L_0} N_1 (C_1 - L_0)^2 + \frac{KL}{L-L_0} N_1 (C - C_1 - L + L_0)^2} \\
\frac{\partial S}{\partial C_1} &= k_B \left[ \frac{-2\frac{KL}{L-L_0}(N_1 - 1)(C - C_1 - L + L_0) - 2\frac{KL}{L_0} N_1 (C_1 - L_0)}{U - \frac{KL}{L_0} N_1 (C_1 - L_0)^2 + \frac{KL}{L-L_0} N_1 (C - C_1 - L + L_0)^2} + \frac{2}{(C - C_1 - L + L_0)} \right]
\end{aligned}$$

Setting the first equation equal to zero, gives:

$$0 = \frac{KL}{L - L_0}(C - C_1 - L + L_0)^2 - \frac{KL}{L_0}(C_1 - L_0)^2$$

If $L_0 = L/4$, then $C_1 = \frac{C - 3L/4 + \sqrt{3}L/4}{\sqrt{3}+1}$, and if $L_0 = 3L/4$ then $C_1 = \frac{\sqrt{3}C + 3L/4 - \sqrt{3}L/4}{\sqrt{3}+1}$. Setting the second equation equal to zero and plugging in each $L_0, C_1$ pair gives the two corresponding $N_1$ values. These expressions for $N_1$ are very complicated, and listing them here would detract from the point of the exercise, which is simply that for this situation there are two possible states that fulfill the given constraints. There is not always just one solution.

The first two examples were presented in order to demonstrate that solving the maximum entropy problem produces physically intuitive results for two systems with unique equilibrium states. Mechanically speaking, the expectation is that a system will tend to settle into a position of minimal energy. From the thermodynamics theory presented, the principle of minimum energy is equivalent to the maximum entropy problem, and will

yield the same results. The entropy maximum was found in each case, and it matched the expected physical results.

The third example was presented to demonstrate that even in a simple one-dimensional case there can be multiple correct solutions to the maximum entropy problem. There can be many equilibrium states which satisfy the given conditions. The question becomes which state to choose. The other goal of the network instantiation procedure to create a network that represents a feasible state the original network could have evolved to, provides some guidance. There are many states that will fulfill the macroscopic constraints, but this field of possibilities can be narrowed by requiring the microscopic elements to be positioned by the extrapolated distribution functions. Once the network has been laid down, it is logical to move the system to the "closest" equilibrium state so as to minimally disturb the configuration predicted by the distribution functions. A nearby equilibrium state can be found by utilizing a gradient search procedure to solve the maximum entropy problem. For the simple cases above, exact solutions were found easily due to the small number of variables involved in each problem. However, the types of networks created in the full three-dimensional model of the cytoskeleton will have a much larger number of variables. This in turn will create a large number of nonlinear equations that will need to be solved to find the equilibrium states. It quickly becomes computationally intractable to solve such a system with direct methods. The approach is instead to use iterative schemes to move the system toward one of its solutions. The typical choice is to use a gradient search algorithm (see Appendix A).

**4.2.2. Equivalent Non-Entropy Formulation.** The entropy formulation is clearly a good approach to finding an equilibrium state for the cytoskeleton network. However as the number of variables increase, and if these variables are allowed to be real-valued, the total number of possible equilibrium states becomes uncountable. It becomes difficult to write down an expression for the entropy as was done so easily in the simple cases presented above. An alternative (yet equivalent) formulation is clearly needed, perhaps relating to the idea of energy minimization even though an energy constraint is given.

This can be formulated as a slightly different minimization problem. Besides requiring $U$ to be a minimum value, the request can be made to also have $(U - U_0)^2$ be minimized, where $U_0$ is the desired energy value. Thus the target function to be minimized will look like:

$$F(\mathbf{x}) = (U(\mathbf{x}) - U_0)^2 + U(\mathbf{x})$$

where $\mathbf{x}$ is a vector containing all variables. The minimization of this target function should be equivalent to the solution of the entropy maximization problem. This equivalence will now be shown by re-doing Examples 1-3 utilizing the above target function, and comparing the results.

**Example 1**: This was the case with two actin filament bundles, each with equilibrium length $L_0$. The target function for this scenario can be written as:

$$F(\mathbf{x}) = (kN_1(C_1-L_0)^2+k(N-N_1)(C-C_1-L_0)^2-U_0)^2+kN_1(C_1-L_0)^2+k(N-N_1)(C-C_1-L_0)^2$$

with $\mathbf{x} = (N_1, C_1)$. Taking partial derivatives with respect to $N_1$ and $C_1$, gives:

$$\begin{aligned}
\frac{\partial F}{\partial N_1} &= 2(kN_1(C_1 - L_0)^2 + k(N - N_1)(C - C_1 - L_0)^2 - U_0)(k(C_1 - L_0)^2 \\
&- k(C - C_1 - L_0)^2) + k(C_1 - L_0)^2 - k(C - C_1 - L_0)^2 \\
\frac{\partial F}{\partial C_1} &= 2(kN_1(C_1 - L_0)^2 + k(N - N_1)(C - C_1 - L_0)^2 - U_0)(2kN_1(C_1 - L_0) \\
&- 2k(N - N_1)(C - C_1 - L_0)) + 2kN_1(C_1 - L_0) - 2k(N - N_1)(C - C_1 - L_0)
\end{aligned}$$

Solving for $N_1$ and $C_1$ yields:

$$N_1 = N/2, \quad C_1 = C/2$$

which leads to $N_2 = N_1$ and $C_2 = C_1$. These are the same results obtained by maximizing the entropy.

**Example 2** In this case, the filaments are grouped into three bundles and all still have an equilibrium length of $L_0$. The target function for this case is:

$$\begin{aligned} F(\mathbf{x}) &= (kN_1(C_1 - L_0)^2 + kN_2(S_2 - L_0)^2 + k(N - N_1 - N_2)(C - C_1 - C_2 - L_0)^2 \\ &- U_0)^2 + kN_1(C_1 - L_0)^2 + kN_2(S_2 - L_0)^2 + k(N - N_1 - N_2)(C - C_1 - C_2 - L_0)^2 \end{aligned}$$

with $\mathbf{x} = (N_1, N_2, C_1, C_2)$. Computing partial derivatives with respect to the four variables will lead to four equations with similar structures to the two bundle case. Solving these four equations for the four unknowns gives:

$$N_1 = N_2 = N_3 = N/3, \quad C_1 = C_2 = C_3 = C/3$$

These results again match the entropy formulation results.

**Example 3** The third example involved two filament bundles. The equilibrium length of the first bundle can either be $L/4$ or $3L/4$. The energy for this system can be written as:

$$U = \frac{KL}{L_0} N_1 (C_1 - L_0)^2 + \frac{KL}{L - L_0}(N - N_1)(C - C_1 - L + L_0)^2$$

and the target function is thus:

$$\begin{aligned} F(\mathbf{x}) &= (\frac{KL}{L_0} N_1 (C_1 - L_0)^2 + \frac{KL}{L - L_0}(N - N_1)(C - C_1 - L + L_0)^2 - U_0)^2 \\ &+ \frac{KL}{L_0} N_1 (C_1 - L_0)^2 + \frac{KL}{L - L_0}(N - N_1)(C - C_1 - L + L_0)^2 \end{aligned}$$

For the case of $L_0 = L/4$, if partial derivatives of the target function are taken with respect to $C_1$ and $N_1$, and are both set equal to zero, $C_1$ is found to be: $\frac{C - 3L/4 + \sqrt{3}L/4}{\sqrt{3}+1}$ as in the entropy formulation. With $L_0 = 3L/4$, the result is $C_1 = \frac{\sqrt{3}C + 3L/4 - \sqrt{3}L/4}{\sqrt{3}+1}$, which also matches the entropy formulation.

These simple examples demonstrate that the minimization of the target function introduced above is equivalent to maximizing the entropy for the same problem. As mentioned previously the entropy becomes more difficult to represent as the problems

become more complex with more variables and dimensions. Beyond these simple examples, it is thus difficult to demonstrate the same equivalence. However, the problems with two or three dimensions and more variables are only extensions of these examples, and the same physical principles apply, thus the minimization of similarly constructed target functions should produce valid results.

**4.2.3. Note on Cytoskeletal Networks.** In the above examples the constraints included restrictions on the number of filaments, filament lengths, and the total energy of the system. Also included were constraints on filament equilibrium length. In these 1D cases, knowing the current and equilibrium lengths of the filament bundles can also be recast as knowing the strain of the system. In the continuum-microscopic model of the cytoskeleton, the continuum level advances the elasticity equations, which will update the velocities, stresses and strains of each three-dimensional grid cell. To have a truly consistent model, the microscopic network within each grid cell should embody the stress and strain state that was found at the continuum level. However, requiring a network to have a particular stress and strain would imply that the two elasticity moduli are known (assuming isotropy). The point of doing the microscopic computation is to determine new elasticity moduli, and thus one does not want to impose both a stress and strain component. Since strain data is being collected anyway, having the average strain state of the network match the strain state of the continuum cell seems to be a logical choice.

However, there is another issue that arises that has to do with the type of microscopic network one is constructing. Utilizing the descriptors of Head, Levine and MacKintosh in [**35**], networks can be affine (have uniform strain distribution) or non-affine (have non-uniform strain distribution). The degree of affinity of a network is correlated to the density of the crosslinks and filament rigidity [**35**]. Typically, a network with high filament stiffness and high crosslink density is affine because the filaments do not have much freedom to move. The strains on these filament tend to be fairly close to the overall imposed strain. In networks with low filament stiffness and low crosslink density, even though a uniform strain may be imposed, the individual filaments may have different

strains due to their ability to move around to situate themselves in a configuration of minimal energy. These networks are non-affine since their strain distribution is non-uniform. Also, the average strain of these filaments may not match the overall imposed strain. This is again due to the filaments' freedom to move to reduce their potential energy. Despite the mismatch, when computing the elasticity moduli of this type of network, the convention is to use the imposed strain state in the calculation as opposed to the network's strain state [**35**]. This is a flaw and its solution is an open research question that will not be addressed in this thesis work, but would be very interesting to delve into in future work.

The non-affine situation is the one presented in this thesis in order to demonstrate the inclusion of an energy minimization procedure. Other types of cytoskeletal networks will be explored in future work. For this case, during network reinstantiation strains are assigned utilizing PDF distributions extrapolated from past strain data (described in the next chapter). These strains may not necessarily average out to the same strain state as the continuum level. However, the distribution of strains will come from the extrapolated microscopic data and therefore should provide an accurate picture of the microstructure's strain state.

CHAPTER 5

# A Continuum-Microscopic Model of Cytoskeletal Mechanics

In this chapter, a continuum-microscopic (CM) model of the cytoskeleton will be presented. The main addition to the basic CM model described in Section 2.5 is a new procedure for carrying out the reinstantiation step (Step 1). The development of this algorithm will be presented first, followed by a description of how this procedure fits into the overall CM framework. As described in Chapter 4, the general idea is to have a continuum description of the cytoskeleton (as a solid elastic material), and a microscopic description of the cytoskeleton (as a crosslinked filament network) and incorporate these two scales into a CM model. The steps of the CM method specific for this cytoskeleton problem will be:

(1) Create a microscopic instantiation of the cytoskeletal network.
(2) Deform the microscopic network a short number of time steps.
(3) Utilize averaging to compute the elasticity moduli for each network block and use them to update the macroscopic constitutive laws
(4) Advance the continuum level equations (elasticity equations) one large time step
(5) Repeast steps 1-4

The main research question faced in this modeling technique is how to accurately reinstantiate the microscopic configuration at a later point in time. In the forthcoming sections, the algorithm that has been developed to accomplish this task will be presented.

As a simple case upon which to build the algorithm, the one block network example presented in the previous chapter will be utilized. Given a particular network, the first goal will be to generate probability distribution functions to represent its data and use these to create a new network that has the same continuum-level properties (such as

elasticity moduli) as the original network. The ultimate goal is to be able to do this reinstantiation procedure at a future point in time, creating a network that represents a configuration the original network could have evolved to.

## 5.1. Network Reinstantiation: Current Time

As stated above, the first goal is to begin with a network block, place it under some strain, collect data from it, and then use this data to construct a new network with very similar average properties and behavior. Data that will be collected includes the orientation of each filament as well as each filament's strain. Two simple cases involving a network of parallel filaments and a network of non-crosslinked filaments will be presented first in order to demonstrate the development of various aspects of the algorithm. This will be followed by a fully crosslinked filament network example.

**5.1.1. Parallel Filaments.** As a simple first case to test that the general PDF construction is working properly, a block of $n$ parallel filaments was created. The block dimensions are $[x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$. Each filament $i$ has a uniformly chosen endpoint on the $x = x_{min}$ wall, meaning a coordinate of $(x_{min}, y_i, z_i)$ where $y_{min} \leq y_i \leq y_{max}, z_{min} \leq z_i \leq z_{max}$. The second endpoint is assigned to be $(x_{max}, y_i, z_i)$, a point on the opposite wall of the block, which creates a filament parallel to the $x$ axis (Figure 5.1). These filaments all have the same angle of orientation and also the same length.

From this initial configuration, the block is strained in the $x$ direction, in the same manner as described in Chapter 4. All $n$ filaments in this case are attached at both ends to walls. No energy minimization is required for this case because there are no internal filament segments and crosslinks to move around. This case is used simply as a test of the probability distribution function generation and network reconstruction basic algorithm. Data collected includes the two angles $\theta_i$ and $\phi_i$ of each filament that define its orientation in three-dimensional space (like in spherical coordinates), as well as the strain $\epsilon_i$ of each filament. For this simple case the strains and angles are the same for

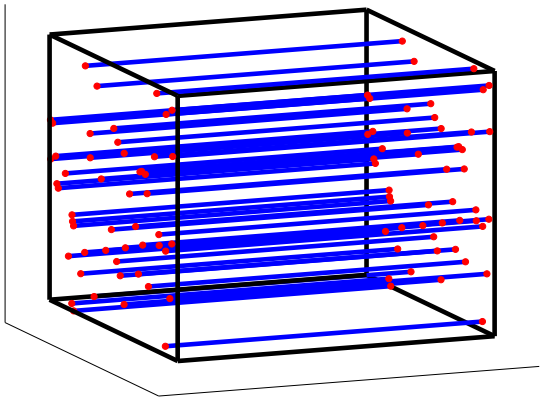

FIGURE 5.1. Network block of parallel filaments.

each filament (see example histogram of $\theta_i$ data in Figure 5.2). Kernel estimation (as described in Chapter 3) using Gaussian kernels and a small window width was used to construct single variable PDFs $f(X)$ for each of the two angles and the strain. These PDFs each have a spiked shape centered over the correct value from the data (Figure 5.2). These PDFs are then used to produce a new set of angles and strains that are assigned to filaments in a new network. In Figure 5.2 one can see that the histogram of the new data is the same as original histogram.

The Young's modulus of the original block is simple to compute. The spring constant $k$ of each filament is the same since they are all the same length. The forces of each filament act only in the $x$ direction since the filaments are all oriented parallel to the $x$ axis. The displacement $\Delta x$ of each filament is the same as the displacement of the block itself. The total force is thus equal to $nk\Delta x$. Assuming the original box was a unit box (all sides length 1), the stress is equal to the force and the strain is simply $\Delta x$. The Young's modulus $E$ is equal to the stress divided by the strain which in this case is: $nk$ (a constant). The newly constructed network should have the same Young's modulus. An example to test this was done with 150 filaments with a spring constant of $k = 0.4$. The modulus of the original network was found to be $E = 60$ as expected. After reinstantiation, the new network's modulus was found to be $E = 60.002$, which is a

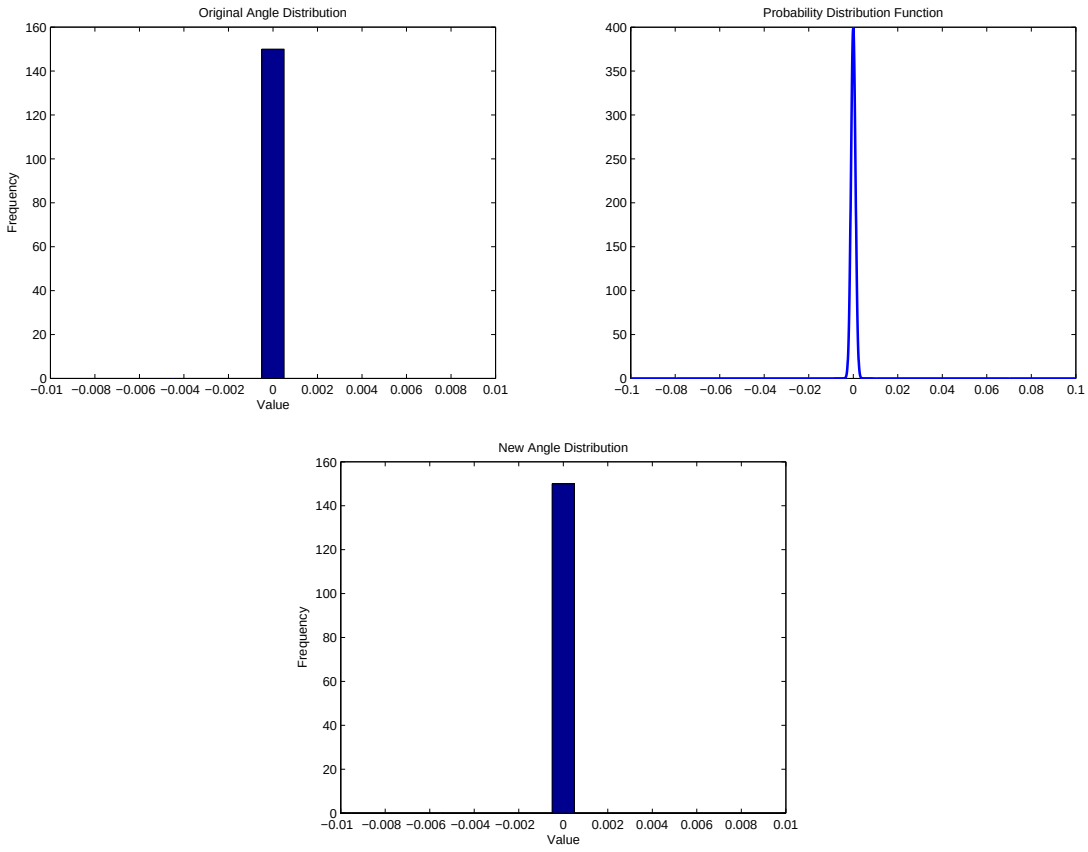


FIGURE 5.2. Each graph shows angle data (one of the two angles): (1) the original network's data, (2) the probability distribution function produced with kernel estimation, (3) the newly generated data for the new network.

relative error of $3.33 \cdot 10^{-5}$. This is to be expected due to the simplicity of this test case, but it is a confirmation that the general code framework successfully renders network reconstructions based on the generated pdfs.

By the law of large numbers, as the number of filaments in the original and reconstructed networks is increased the relative error between the network properties should decrease. To check this, the same test as above was run with different numbers of filaments. Each test with $n$ filaments (where $n = 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000$) was run 1000 times and the found elasticity moduli were averaged. The relative error between this average modulus and the original network's elasticity modulus was computed. Figure 5.3 shows the log-log plot of the error vs. the number of filaments. As the number

of filaments increases the relative error goes down. The slope of the regression line is -0.46, which is close to the convergence rate of -0.5 given by the law of large numbers.

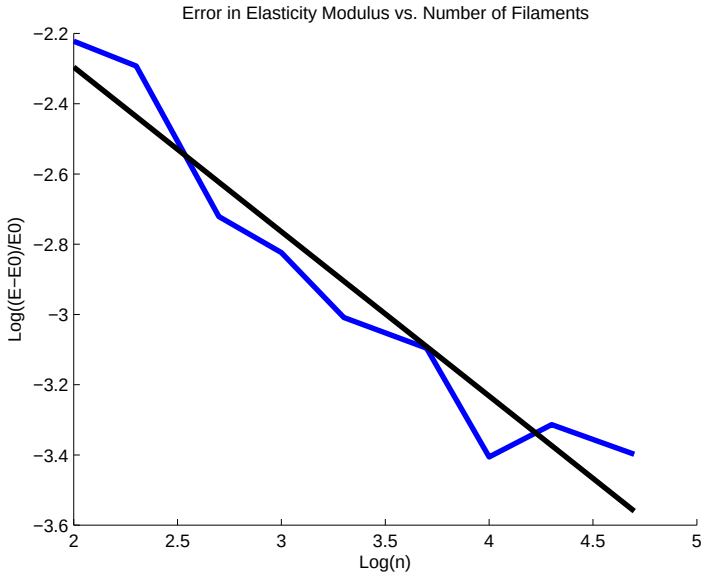


FIGURE 5.3. This graph depicts the log of the relative error between the original elasticity modulus and the ensemble averaged elasticity modulus of the reconstruction versus the log of the number of filaments in the network. The data is shown in blue and the least squares regression line is shown in black (has slope of -0.46).

**5.1.2. Non-Crosslinked Filaments.** In this next case, all filaments begin with one endpoint $(x_0, y_0, z_0)$ on one of the six walls of the three-dimensional block, and can grow in any direction. They are made long enough so that they will hit another wall of the block (Figure 5.4), so their second endpoint $(x_1, y_1, z_1)$ is also attached to a wall. No crosslinks are established for this case, thus like in the first case no energy minimization is needed. However, unlike the parallel case, there are many more possibilities for the two angles and strains of each filament.

Figure 5.5 shows a typical data set for filament strain after the block has been strained in the $x$ direction. This figure also shows the data's estimated PDF and CDF, as well as a new data set generated by this PDF. The new data appears qualitatively to be a good reproduction of the initial data set. As a quantitative measure, the average relative error between the original and new histogram heights was compured and found to be on the

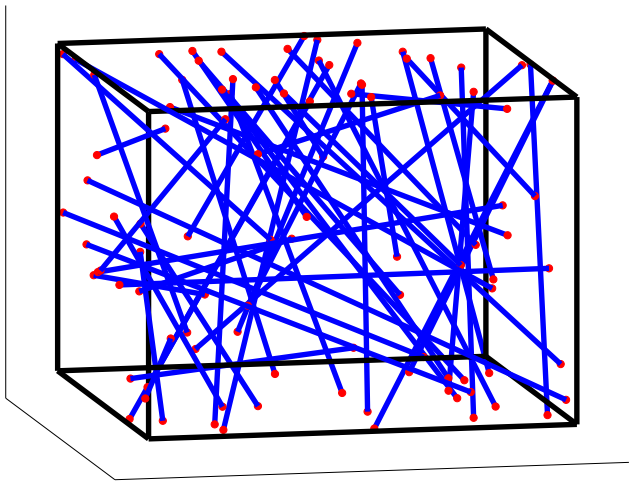

FIGURE 5.4. Network of non-crosslinked filaments.

order of $10^{-2}$. Non-parametric PDF estimation error is likely the dominant error source, with factors such as kernel type and window width playing a role. Without knowing the underlying PDF it is difficult to know how to construct an optimal PDF estimation algorithm for the given data set.

To conduct the same Young's modulus before and after comparison test as was done in the parallel filament test, each filament's two angles ($\theta$ and $\phi$) and strain $\epsilon$ are collected and single variable PDFs were constructed for each data set: $f(\theta), g(\phi), h(\epsilon)$. A new network was then established as follows. The box has the same dimensions as the original block in its strained state. A filament is laid in this box by placing its first endpoint $(x_0, y_0, z_0)$ on one of the six walls as before. A $\theta$ and a $\phi$ value were then generated from $f(\theta)$ and $g(\phi)$ in order to establish a direction vector $(x_{dir}, y_{dir}, z_{dir})$ for the filament. The

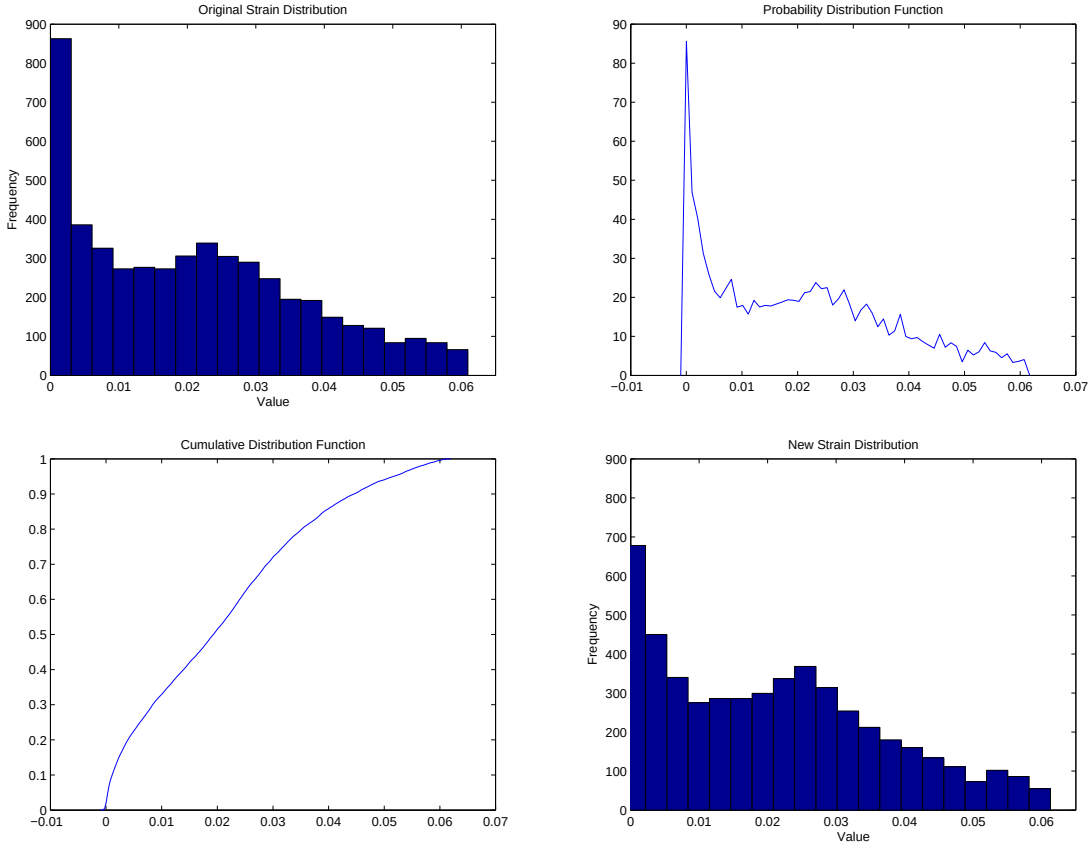


FIGURE 5.5. The original network's strain distribution, the probability distribution function produced with kernel estimation for this data, the cumulative distribution function and the newly generated strain data for the new network.

second endpoint $(x_1, y_1, z_1)$ was found using:

$$x_1 = x_0 + Lx_{dir}$$

$$y_1 = y_0 + Ly_{dir}$$

$$z_1 = z_0 + Lz_{dir}$$

where $L$ is the filament length, long enough to insure that it reaches another wall of the block. The second endpoint is then modified to be the intersection point of the filament with this wall. The filament is then assigned a strain $\epsilon$ utilizing $h(\epsilon)$. The elasticity modulus of this new network was then computed and compared against the original

network's modulus. (The elasticity modulus was computed using the method described in Chapter 4 since this block is again under a simple extensional strain.)

As in the parallel filament case, this test was run for increasing numbers of $n$ filaments, with multiple runs for each $n$ case to produce an average error. The results are shown in Figure 5.6 in table form and also in a log-log plot.

| **Number of Filaments** | **Original $E_0$** | **Average New $E$** | **Relative Error: $(E - E_0)/E_0$** |
|---|---|---|---|
| 500 | 21.529 | 15.933 | 0.3528 |
| 1000 | 40.109 | 25.511 | 0.36539 |
| 2000 | 83.407 | 53.124 | 0.3638 |
| 5000 | 216.06 | 143.334 | 0.3366 |
| 10000 | 437.378 | 275.27 | 0.3708 |
| 20000 | 863.3838 | 552.09 | 0.36055 |
| 50000 | 2191.42 | 1392.21 | 0.364697 |

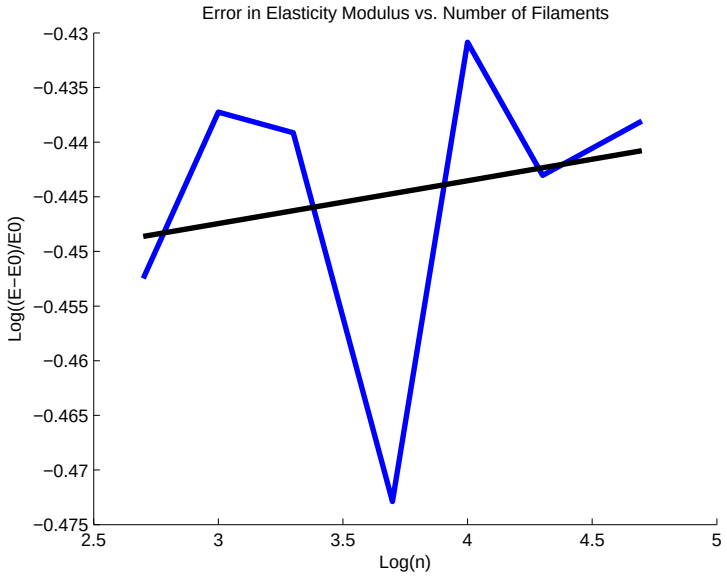


FIGURE 5.6. The table displays the relative errors in elasticity modulus for different numbers of filaments in a network, using three independent pdf reconstructions for the strain and each angle. The graph is a log-log plot the error vs. the number of filaments. The data is in blue and the line of best fit is in black.

The networks constructed with these distributions do not have the same elasticity modulus as the original network. The average elasticity modulus comes out consistently lower then the original with the relative errors on the order of $\approx 0.36$. There is no convergence as the number of filaments is increased. These results can be attributed to

the fact that the new strains are being assigned to the filaments without regard to the orientation of that filament. Intuitively, the angles and strain should be related, because filaments that are more aligned with the axis of strain have larger strains. Filaments which are more perpendicular to the axis of strain should have very low strains. Evidence of a relationship between angles and strains can be shown by computing the correlation coefficients between pairs of the three variables. These correlation coefficients are:

| **Variable 1** | **Variable 2** | **Correlation Coefficient** |
|---|---|---|
| Strain | Angle 1 | 0.51 |
| Strain | Angle 2 | 0.39 |
| Angle 1 | Angle 2 | 0.071 |

It is clear from these correlation coefficients that a relationship does exist between strain and each angle. The two angles are not as highly coorelated with one another. The correlation coefficients between each angle and the strain imply that single variable distributions $f(\theta), g(\phi), h(\epsilon)$ cannot be used to separately assign the two angles and a strain to each new filament. Single variable PDFs can only be used if variables are independent (correlation coefficients equal to zero). What is needed is a joint probability distribution function $F(\theta, \phi, \epsilon)$. This function provides a relationship between the variables, and can be used to find the probability of having $\theta \in [\theta_0, \theta_0 + h], \phi \in [\phi_0, \phi_0 + h], \epsilon \in [\epsilon_0, \epsilon_0 + h]$. The problem with joint PDFs as opposed to single variable PDFs is that the dimension of the phase space increases and the situation becomes more complex. Instead of three, one-dimensional phase spaces for each variable, this joint PDF has one, three-dimensional phase space. In a discrete case, suppose there are $m$ possible states for each variable, then there are $m^3$ possible states for the variable triplet $\theta, \phi, \epsilon$. Suppose a minimum of $p$ filaments is considered a reasonable number of data points for PDF estimation in the single variable case. This means that a minimum of $p^3$ filaments will be needed to get an equivalent amount of data to be able to estimate the joint PDF. Also the PDF estimation process becomes more complex in that one must now use multivariate kernel functions. The data generation process from a multivariate PDF for the new network also becomes

more complex. For the single variable case, one uses the CDF and a basic reverse interpolation procedure to find a data point. In the multivariate case, the CDF (like the PDF) is a surface, and the reverse interpolation procedure is much more complex and computationally taxing.

To circumvent this issue, the goal will be to establish a relationship between the strain and angles in a different manner. One possibility is to do single variable distributions for the angles, and then deterministically assign each filament a strain based on these angles. As a simple first test of this idea, consider a two-dimensional case, where there is only one angle involved. The strain of a filament can be determined by its current angle of orientation and the overall strain of the box in the following way. (This derivation is based on an extensional strain in the principle $x$ direction, but a similar formula could be derived for a more complex strain tensor.) Figure 5.7 will be used for notation where the hypotenuses of the triangles represent a filament before and after stretching. The $l$ is the original length of the box within which this filament is placed and the $\Delta l$ is the displacement of the block in the $x$ direction.

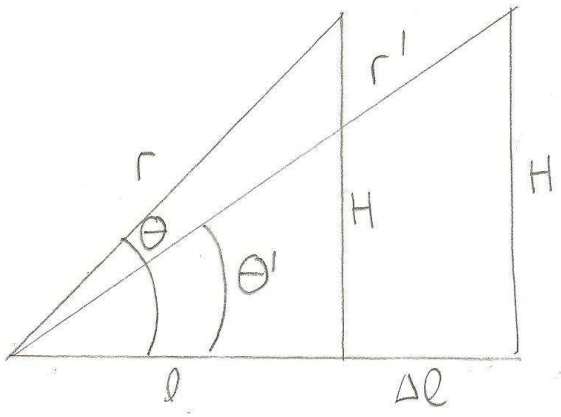


FIGURE 5.7. Diagram of a two-dimensional filament strain calculation

The strain of this filament $\epsilon_{new}$ can be described as:

$$\begin{aligned}
\epsilon_{new} &= \frac{r' - r}{r} \\
&= \frac{\frac{l+\Delta l}{\cos(\theta')}}{\frac{l}{\cos(\theta)}} - 1 \\
&= (1+\epsilon)\frac{\cos(\theta)}{\cos(\theta')} - 1
\end{aligned}$$

where $\epsilon$ is the overall strain of the block. The goal is to find $\epsilon_{new}$ in terms of only its current angle $\theta'$. The dependence on $\theta$ can be removed by creating another relationship using tangents:

$$\begin{aligned}
\frac{\tan(\theta)}{\tan(\theta')} &= \frac{\frac{H}{l}}{\frac{H}{l+\Delta l}} \\
&= (1+\epsilon) \\
\theta &= \arctan((1+\epsilon)\tan(\theta'))
\end{aligned}$$

This last line can be substituted into the filament strain equation to give:

$$\epsilon_{new} = (1+\epsilon)\frac{\cos(\arctan((1+\epsilon)\tan(\theta'))}{\cos(\theta')} - 1$$

Utilizing this relationship to assign a strain to a filament, the following results, (shown in Figure 5.1.2), were obtained for the elasticity modulus test. The relative error between the original and new Young's modulus attenuates to values in the $10^{-3}$ range as the number of filaments is increased. This is a large improvement over the three single variable PDF method.

These results show that in this simple case the strain can be deterministically assigned by knowing the angle. The need to collect strain data has been completely eliminated.

| Number of Filaments | Relative Error $(E - E_0)/E_0$ |
|---|---|
| 100 | 0.0312 |
| 200 | 0.00612 |
| 500 | 0.001 |
| 1000 | 0.0044 |
| 2000 | 0.00233 |
| 5000 | 0.00188 |
| 10000 | 0.00275 |
| 20000 | 0.0046 |
| 50000 | 0.00452 |

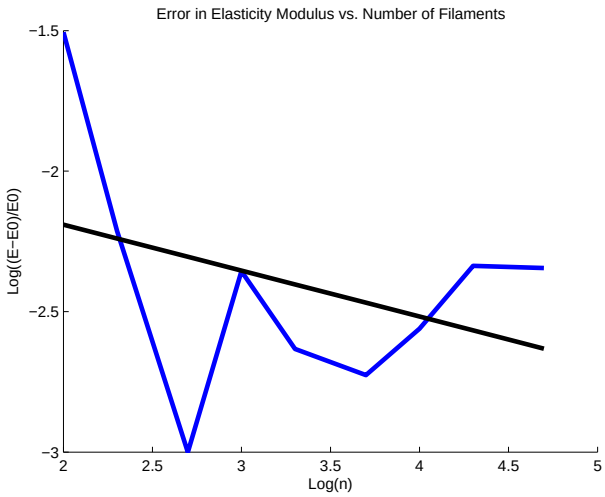


FIGURE 5.8. The relative error in elasticity moduli found for increasing numbers of filaments within the 2D network where strains are deterministically assigned. The blue line is the data and the black is the regression line with slope -0.17

A similar relationship can be established for a three-dimensional case. Here again, the formula will be derived for an extensional strain in the $x$ direction but could be derived for more complex strain fields if needed. Figure 5.9 will be used as a guide for the notation in the derivation of the formula.

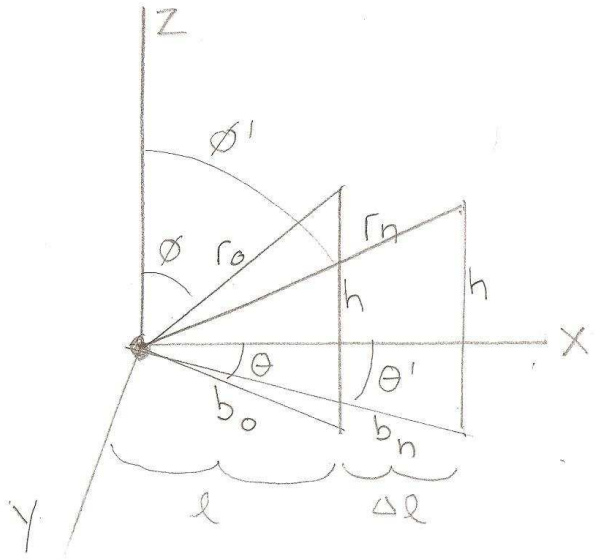


FIGURE 5.9. Diagram of three-dimensionally strained filament

Let $r_0$ and $r_{new}$ represent the equilibrium and strained lengths of the filament. The new strain of the filament is given by:

$$\epsilon_{new} = \frac{r_{new} - r_0}{r_0} = \frac{r_{new}}{r_0} - 1 \tag{5.1}$$

The goal is to replace $r_0$ and $r_{new}$ with expressions involving the overall strain of the block $\epsilon$ in the $x$ direction and the filament's current orientation angles. Due to the three-dimensionality, this is a multi-step process to find the correct relationship. The $\theta$ angles in Figure 5.9 are the angles the filament makes with the $x$-axis if projected into the $x$-$y$ plane, and the $\phi$ angles are the angles made with the $z$-axis if the filament is projected onto the $x$-$z$ plane. Expressions for $r_0$ and $r_{new}$ can be developed as follows:

$$r_0 = \frac{b_0}{\cos(s)} \qquad r_{new} = \frac{b_{new}}{\cos(s')}$$

where $s = \frac{\pi}{2} - \phi$ and $s' = \frac{\pi}{2} - \phi'$. The values $b_0$ and $b_{new}$ can be written in terms of angles and strains. The $b$ variables are the hypotenuses of the triangles representing the projection of the filaments onto the $x$-$y$ plane (Figure 5.10).

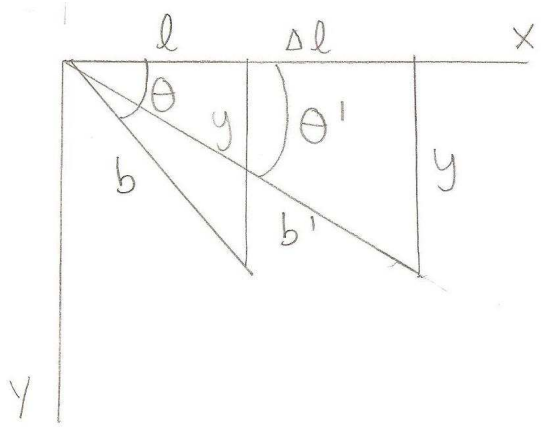


FIGURE 5.10. Diagram of the $x$-$y$ plane with the projections of the equilibrium and strained filaments

The $b$ variables can be written as:

$$b_0 = \frac{l}{\cos(\theta)} \qquad b_{new} = \frac{l + \Delta l}{\cos(\theta')}$$

Substituting these expressions into the expressions for $r_0$ and $r_{new}$, Equation 5.1 looks like

$$\begin{aligned} \epsilon_{new} &= \frac{\frac{l+\Delta l}{\cos(\theta')\cos(s')}}{\frac{l}{\cos(\theta)\cos(s)}} - 1 \\ &= (1+\epsilon)\frac{\cos(\theta)\cos(s)}{\cos(\theta')\cos(s')} - 1 \end{aligned} \tag{5.2}$$

As in the 2D case, the goal is to write $\epsilon_{new}$ in terms of only $\theta'$, $s'$ and $\epsilon$ so $\theta$ and $s$ must be eliminated. This can be done using tangent relationships as was done for the two-dimensional case.

$$\frac{\tan(\theta)}{\tan(\theta')} = \frac{\frac{y}{l}}{\frac{y}{l+\Delta l}}$$

$$\frac{\tan(\theta)}{\tan(\theta')} = (1+\epsilon)$$

$$\theta = \arctan((1+\epsilon)\tan(\theta')) \tag{5.3}$$

A similar expression can be found for $s$:

$$\frac{\tan(s)}{\tan(s')} = \frac{\frac{l+\Delta l}{\cos(\theta')}}{\frac{l}{\cos(\theta)}}$$

$$\frac{\tan(s)}{\tan(s')} = (1+\epsilon)\frac{\cos(\theta)}{\cos(\theta')}$$

$$s = \arctan((1+\epsilon)\frac{\cos(\theta)}{\cos(\theta')}\tan(s')) \tag{5.4}$$

Equations 5.3 and 5.4 can be substituted into 5.2 to find $\epsilon_{new}$ as:

$$\epsilon_{new} = (1+\epsilon)\frac{\cos(\arctan((1+\epsilon)\tan(\theta')))\cos(\arctan((1+\epsilon)\frac{\cos(\arctan((1+\epsilon)\tan(\theta')))}{\cos(\theta')}\tan(s')))}{\cos(\theta')\cos(s')} - 1$$

The results of the elasticity modulus test for a 3D network using this 3D deterministic relationship for assigning the strain are shown in Figure 5.11. These results parallel the results found in the two-dimensional case, although the convergence rate is better ($\approx -0.35$). The relative errors settle out to values in the 0.7% range.

| Number of Filaments | Relative Error $(E - E_0)/E_0$ |
|---|---|
| 100 | 0.0407 |
| 200 | 0.0470 |
| 500 | 0.0198 |
| 1000 | 0.0146 |
| 2000 | 0.0106 |
| 5000 | 0.00721 |
| 10000 | 0.0032 |
| 20000 | 0.0069 |
| 50000 | 0.00871 |

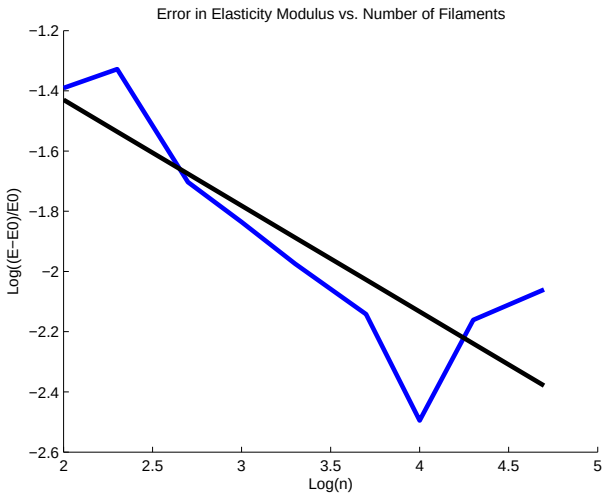


FIGURE 5.11. The relative errors in elasticity moduli found for increasing numbers of filaments within the 3D network when strains are assigned deterministically. The blue line is the data and the black is the regression line with slope -0.35

These strain-angle relationships have worked well for these simple non-crosslinked cases. However the relationships will likely break down for more complex models that include crosslinking and external effects such as fluid forces and biochemical reactions. A relationship that utilizes the real strain data would be a more general approach that should be applicable to a wider range of network systems.

One idea is to replace a global joint probability distribution function $F(\theta, \phi, \epsilon)$ with local, piecewise PDFs. The idea goes as follows. In the three-dimensional case the two angles $\theta$ and $\phi$ determine the filament's orientation. The correlation coefficient between the two angle variables is very low in comparison to the correlation coefficients between the strain and each angle. It is a reasonable assumption that the two angles are still independent and thus single variable PDFs will be constructed for each angle data set and used to assign an orientation to each filament. The strains are correlated to the angles and will be assigned in a different way.

Angle $\theta$ ranges from $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and angle $\phi$ ranges from $[0, \pi]$. Divide the range of $\theta$ angles into $m$ bins of equal length $\frac{\pi}{m}$. These intervals look like $[-\frac{\pi}{2} + \frac{\pi}{m}i, -\frac{\pi}{2} + \frac{\pi}{m}(i+1)]$ with $i = 0..m-1$. The same is done for the $\phi$ angle values, whose intervals look like

$[\frac{\pi}{m}j, \frac{\pi}{m}(j+1)]$ with $j = 0..m-1$. If a filament has orientation angles $\theta', \phi'$, these angles fall into a bin pair: $\theta' \in [-\frac{\pi}{2} + \frac{\pi}{m}i, -\frac{\pi}{2} + \frac{\pi}{m}(i+1)], \phi' \in [\frac{\pi}{m}j, \frac{\pi}{m}(j+1)]$. This bin pair will be indexed as $i, j$. There are $m \times m$ bin pairs. Once it has been determined that filament $k$ falls into bin pair $i, j$, the filament's strain $\epsilon_k$ is recorded in this bin $i, j$. Once all filament strains have been binned, the mean $\mu_{i,j}$ and standard deviation $\sigma_{i,j}$ of strain data in each bin pair $i, j$ are computed and used to construct a local normal distribution $N(\mu_{i,j}, \sigma_{i,j})$. These local distributions for the strain data are put together in a global piecewise distribution for the strain. During reinstantiation, a filament $k$ is assigned two angles $\theta_k, \phi_k$, and its strain is assigned using the following distribution:

$$H(\epsilon) = N(\mu_{i,j}, \sigma_{i,j}) \text{ when } \theta_k \in [-\frac{\pi}{2} + \frac{\pi}{m}i, -\frac{\pi}{2} + \frac{\pi}{m}(i+1)], \phi_k \in [\frac{\pi}{m}j, \frac{\pi}{m}(j+1)]$$

Strain $\epsilon_k$ will be generated from this normal distribution. Figure 5.12 gives a visual representation of what is being done.

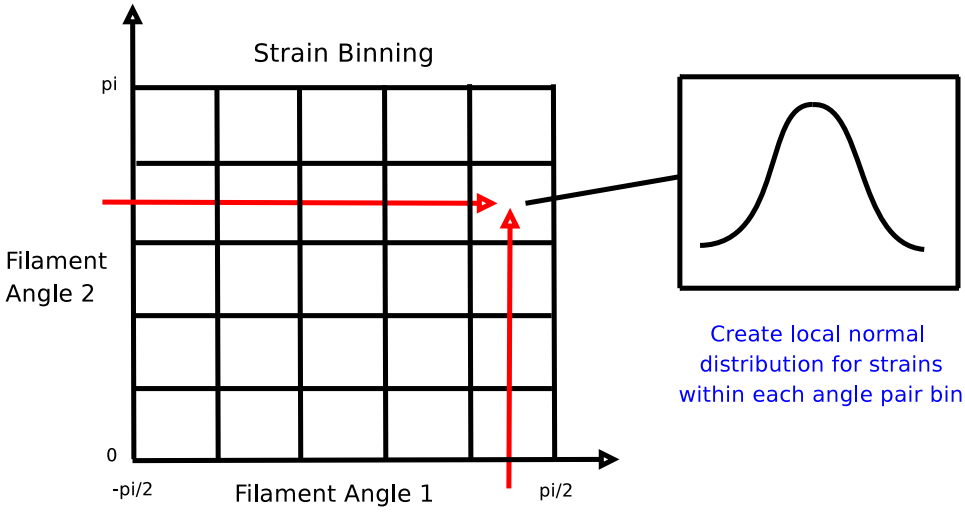


FIGURE 5.12. The strain of a filament is saved into an angle pair bin (based on its two orientation angles). Once all strains have been stored in these bins, a mean and variance is calculated for the data within each bin. These values are then used to create a normal distribution for the strains for each bin.

Results of the elasticity modulus test done using this strain assignment method are shown in Figure 5.13. This method had a convergence rate of -1.36 and relative errors reaching the 0.4% range as the number of filaments is increased.

| Number of Filaments | Relative Error |
|---|---|
| 500 | 0.7414 |
| 1000 | 0.551 |
| 2000 | 0.305 |
| 5000 | 0.0515 |
| 10000 | 0.00687 |
| 20000 | 0.00390 |
| 50000 | 0.00413 |

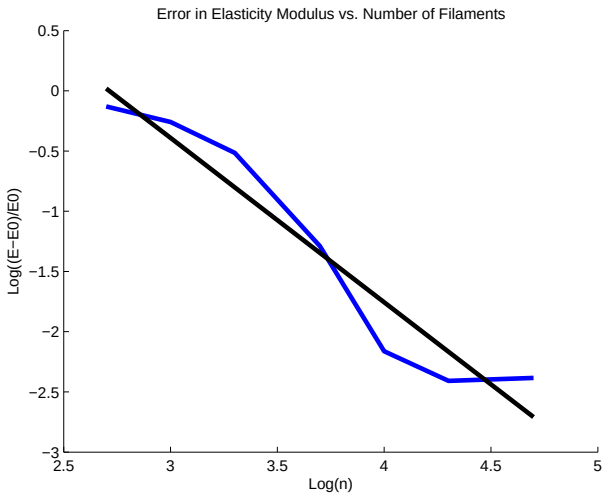


FIGURE 5.13. The relative error in elasticity modulus found for increasing numbers of filaments within the 3D network using the strain binning method. This data has slope -1.36

The large errors for the lower numbers of filaments ($n$) is attributed to the small amount of data. For networks with a small $n$ there may be not be enough data to construct the Gaussians within each bin (some bins may be empty). However, the results quickly improve as the number of filaments are increased. This method still works best with larger amounts of data, but it avoids the numerical complexities of multivariate PDF estimation and data regeneration that would occur with a joint probability distribution function. The conclusion is that this binning algorithm is a valid substitute for a joint probability distribution function.

**5.1.3. Crosslinked Filaments.** The ideas presented above will now be extended to the case of a fully crosslinked network of filaments. For this case, the network will be set up exactly as described in the example in Chapter 4, and a minimization problem will be solved to rearrange the network after each strain step. The function to be minimized will not simply be the energy, but will look instead like the functions presented in the thermodynamics examples of Chapter 4 ($F(X) = (U(X) - U_0)^2 + U(X)$, where $U(X)$ is the energy). However first some examples will be presented where the function to be minimized is only the energy ($G(X) = U(X)$) to illustrate the problems that occur that led to the development of $F(X)$.

The first test will be to try to use the deterministic relationship derived in the previous section to assign strains to new filaments during network reconstruction. The only difference between this case and the previous one is that after each strain step, the internal filament segments and crosslinks are rearranged into a position of minimal energy by the same energy minimization procedure described in the previous chapter. The same elasticity modulus comparison test (original vs. new) was conducted for networks with increasing numbers of filaments. The results of this test are shown in Figure 5.14.

| **Number of Filaments** | **Number of Segments** | **Relative Error** $(E - E_0)/E_0$ |
|---|---|---|
| 100 | 145 | 0.0426 |
| 200 | 401 | 0.005 |
| 500 | 1955 | 0.10 |
| 1000 | 6619 | 0.14 |
| 2000 | 24785 | 0.15 |

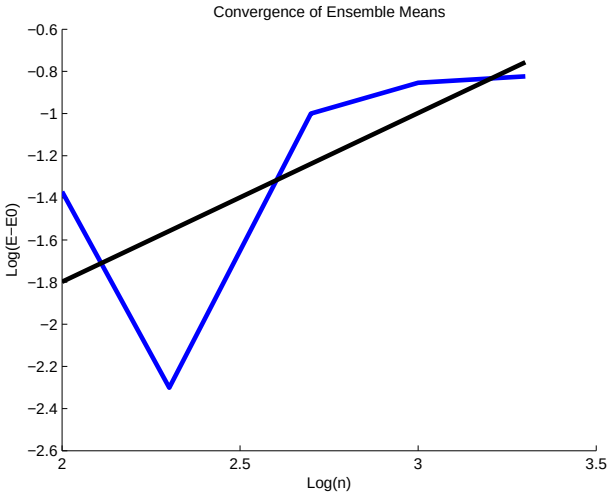


FIGURE 5.14. The relative error in elasticity modulus with increasing numbers of filaments in a 3D network. In this case the strains were assigned using the three-dimensional deterministic relationship. This data has positive slope of 0.8

This method does not converge, and the elasticity moduli produced in the reconstructions tends to run on the order of 10% higher than the original network's modulus. This can be explained by the fact that there are crosslinks present. The energy minimization rearranges the filament segments and crosslinks into positions of minimal overall energy.

Since the filaments have the freedom to move to minimize their energy, the filament strains will tend to decrease. The deterministic relationship does not take this into account and thus in general will assign a higher strain to the filament than it had in the original network.

In a second test, the binning method described in the previous section was used to assign strains. The results of this test are shown in Figure 5.15. In this case, the reconstructed networks have elasticity moduli that are too low compared to the original. However this too has a logical explanation. Data is collected from the original network after energy minimization has taken place. Given the same boundary conditions, a new network is created during reconstruction. This network has different crosslinking connectivity than the original network. The filaments in this new network are assigned strains $\epsilon_i$ via the binning method. Due to the fact that this strain assignment method is still a random process, it is likely that there is a less costly (energetically-speaking) configuration for this new network to be in. Thus when the energy minimization is run the new network settles to a state of lower energy and thus a lower elasticity modulus. Figure 5.16 gives a visual description of what is happening. The red data represents the total energy of the original network. Each group of red data points is a strain step, with each data point representing the energy after another $m$ minimization steps (until convergence was reached). The blue data represents the total energy of two example new networks. The points represent the network's energy after each $m$ energy minimization steps (until convergence is reached). What is important to take note of is that the initial energy of each of the two example new networks is close to that of the original network. However, there is a steep drop in energy when the energy minimization is applied. The network is able to find a new configuration that significantly lowers its total energy.

The solution to this problem is where thermodynamics comes into play. The goal is to create a network that possesses the same stored energy as the original network and whose filaments are arranged in a configuration of minimal energy. Mathematically, the variables $X$ of the network should satisfy both $G(X) = U(X) \rightarrow \nabla G(X) = 0$

| Number of Filaments | Number of Segments | Relative Error |
|---|---|---|
| 100 | 145 | 0.10 |
| 200 | 401 | 0.0557 |
| 500 | 1955 | 0.078 |
| 1000 | 6619 | 0.16 |
| 2000 | 24785 | 0.0706 |

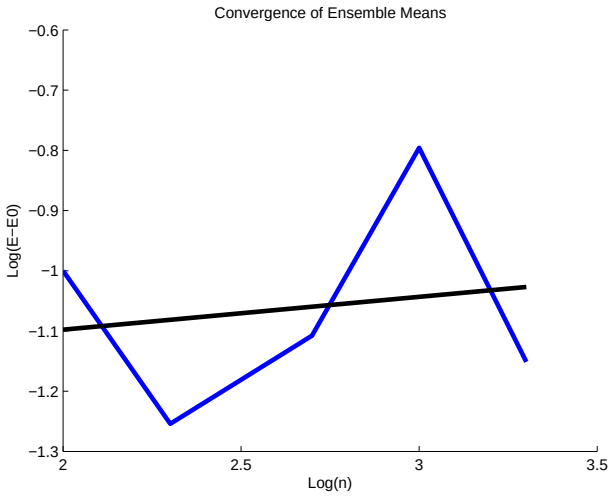


FIGURE 5.15. The relative error in elasticity modulus with increasing numbers of filaments in a 3D network. In this case the strains were assigned using the binning method. This data has positive slope of 0.05

and $U(X) = U_0$, where $U_0$ is the desired stored energy. These two expressions can be combined into one function:

$$F(X) = (U(X) - U_0)^2 + U(X)$$

which is the target function described in the previous chapter. The goal is find a global minimum of this function, where hopefully both parts are simultaneously minimized. For simple cases like those presented in the previous chapter, the solution is easy to find analytically. For the networks in the cytoskeleton simulation an analytical solution will not be possible, thus a numerical solution generated by an iterative procedure is sought instead. The problem one may encounter with such a complex system is that the minimization iterations may lead to a local minimum rather than a global minimum. Also, an issue that is likely to occur is that since the two requirements $U(X) = U_0$ and

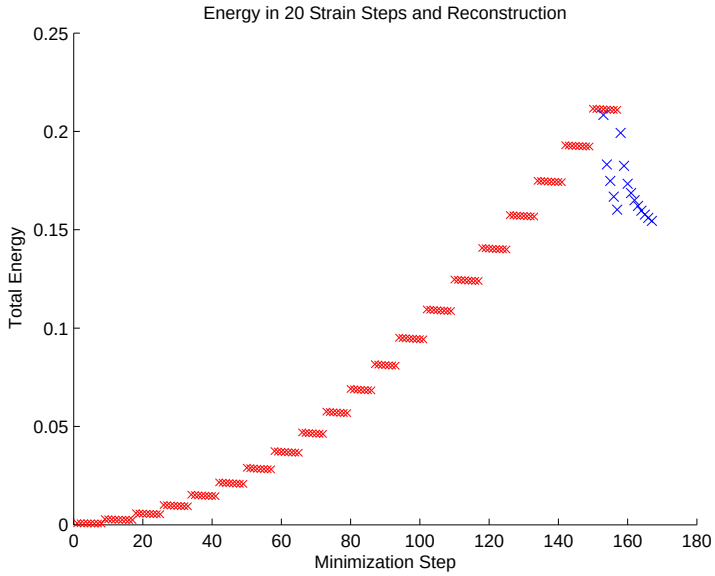


FIGURE 5.16. The red data points represent the total energy within the 20 strain steps which occur in the simulation. Each "step" contains several data points that show the energy decrease that occurs during the application of the energy minimization procedure after each strain. The two sets of blue data represent the total energy in two reconstructed networks that have been run through the energy minimization algorithm. Reconstruction was done using the binning method.

$U(X)$ be a minimum have been combined into one function $F(X)$, there is a possibility that both conditions may not be satisfied.

Because both conditions are important to satisfy, a different approach has been taken to this problem. The idea goes as follows. The total energy of the system is:

$$Energy = \sum_{j=1}^{m} \left[ \frac{k_j}{2} (L^j - L_0^j)^2 \right] \tag{5.5}$$

with $L^j$ and $L_0^j$ the current and equilibrium lengths respectively. When a strain is assigned to a filament during reinstantiation, what is really being done is that the filament is assigned an equilibrium length $L_0^j$ that gives the filament the desired strain. If this system is then modified so as to minimize $U(X)$ the current length $L^j$ of the filament changes, but the equilibrium length remains the value it was assigned. After an application of the minimization of $U(X)$ most filaments end up with a lower strain. The goal is to choose strains for the filaments such that when the energy minimization is applied the

end result is that $U(X)$ of the system is equal to $U_0$. This can be achieved by utilizing a method based on a root-finding method such as bisection. First, a strain $\epsilon_i$ is assigned to each filament via the binning method with distribution $H(\epsilon)$ described previously. Energy minimization is then applied. Typically the resulting total energy $U_1$ is lower than the desired $U_0$ values. Suppose that $U_1 < U_0$, then the $\epsilon_i$ values drawn from the $H(\epsilon)$ distribution are marked as a lower bound for the bisection method. To create an upper energy bound, begin with the same initial filament configuration and strain assignment $\epsilon_i$. However, modify each $\epsilon_i$ as:

$$\epsilon_i^{new} = C\epsilon_i \tag{5.6}$$

where $C$ is a constant and is greater than 1. This will raise all the strains of all the filaments by the same percentage. The energy minimization is then run on this system to find the final energy state $U_2$. If $U_2 > U_0$ then an upper bound has been found. If $U_2 < U_0$ then process 5.6 must be repeated again with a larger $C$ value until a final configuration is found with $U_2 > U_0$.

In some cases the first energy computed $U_1$ may be greater than $U_0$. If so, then this becomes the upper bound case and a lower bound is created by doing 5.6 with a $C < 1$.

Once an upper and lower bound have been found, bisection can be used to zero in on a $C$ value that will result in a network whose minimal energy state has total energy $U_0$. Figure 5.17 shows several steps of the process.

The configuration found via this method certainly is a minimum of the target function $F(X)$. It is by no means the only possible solution. However, as stated in the previous chapter, utilizing the angle and strain distributions should narrow the field of possibilities to a set of states that the original system could have evolved to. Since thermal fluctuations exist in this system, it is feasible to conclude that these states are equally valid, and thus choosing one as a configuration for the reinstantiated network is acceptable.

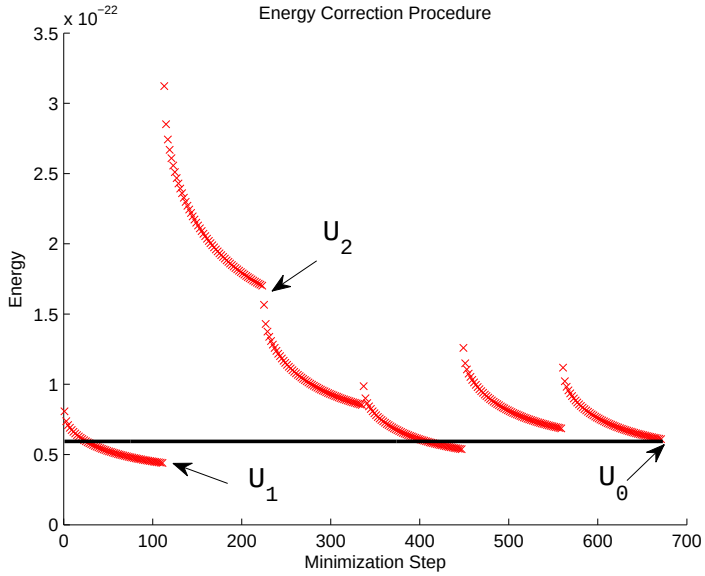


FIGURE 5.17. Several steps of the energy correction algorithm. The line is the target energy value $U_0$. The cross data shows the application of the energy minimization procedure to the same system with different initial strain states.

Results from the elasticity modulus comparison test are shown in Figure 5.18. The relative errors approach the 0.5% range and the rate of convergence is approximately -1.15.

## 5.2. PDF Extrapolation: Future Time

With the main reconstruction algorithm in place, the focus now turns to performing the reconstruction at future points in time. Instead of only collecting angle and strain data after the last microstep, data will be collected after every microstep. PDFs for the angles and strains will be constructed at each microstep in accordance with the binning method. This is done so that the evolution of the distributions can be tracked over time, and used to predict what a variable's distribution will look like in the future.

The algorithm will now be described in general mathematical terms for the extrapolation of the two angle PDFs. After a microscopic time step, the angle data is collected and used to construct two single variable distribution functions for the angles $\theta$ and $\phi$.

| Number of Filaments | Number of Segments | Relative Error of Elasticity Modulus |
|---|---|---|
| 200 | 401 | 0.13 |
| 500 | 1955 | 0.0592 |
| 1000 | 6619 | 0.03 |
| 2000 | 24785 | 0.00512 |
| 5000 | 146537 | 0.004509 |

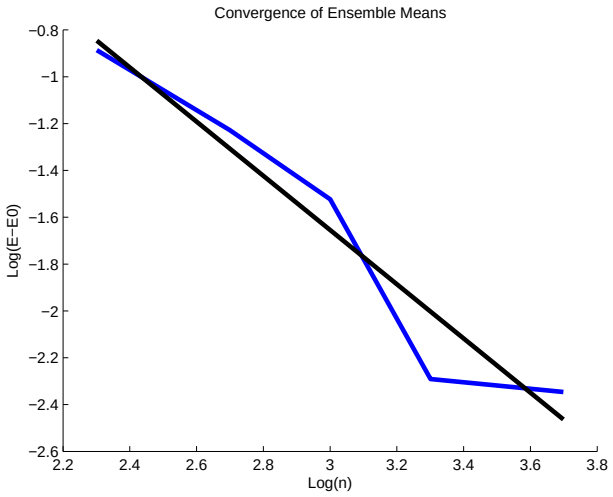


FIGURE 5.18. The relative error in elasticity modulus with increasing numbers of filaments in a 3D network. In this case the strains were assigned using the binning method, and adjusted to the correct total energy using the bisection method described above. This data has slope of -1.15

The PDFs are created via kernel estimation and have the general form:

$$f(\theta) = \frac{1}{m}\sum_{i=1}^{m} K\left(\frac{\theta - \theta_i}{h}\right)$$

$$g(\phi) = \frac{1}{m}\sum_{i=1}^{m} K\left(\frac{\phi - \phi_i}{h}\right)$$

where $m$ is the total number of data points (which will be the number of filament segments), and $K$ is the Gaussian kernel. These two functions give the PDFs of the angle distributions at one point in time. The goal is to understand how these PDFs evolve over

time, so these equations can be rewritten with a time dependence as:

$$f(\theta,t) = \frac{1}{m}\sum_{i=1}^{m} K\left(\frac{\theta-\theta_i(t)}{h}\right) \tag{5.7}$$

$$g(\phi,t) = \frac{1}{m}\sum_{i=1}^{m} K\left(\frac{\phi-\phi_i(t)}{h}\right) \tag{5.8}$$

(5.9)

The time dependence arises in the angle data points themselves which change from time step to time step as new stresses are applied to the network block. However, the goal is not to follow the evolution of the angles of the individual filament segments over time, but rather the general shape evolution of the distribution functions. To do this $N$ angle values, evenly spaced in the interval $\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$ for $\theta$ and $[0,\pi]$ for $\phi$ are chosen. The values look like $\theta_i = -\frac{\pi}{2} + \frac{\pi}{N-1}(i-1)$ and $\phi_i = \frac{\pi}{N-1}(i-1)$ with $i = 1..N$. Their function values at time $t_j$ are computed using 5.7-5.8. These function values are denoted as: $f(\theta_i,t_j), g(\phi_i,t_j)$ with $i = 1..N$ and $j = 1..n$ (the total number of microsteps). This gives $n$ coordinate pairs per angle $i$ for both $\theta$ and $\phi$:

$$(t_1, f(\theta_i,t_1)), (t_2, f(\theta_i,t_2)), ..., (t_n, f(\theta_i,t_n))$$

$$(t_1, g(\phi_i,t_1)), (t_2, g(\phi_i,t_2)), ..., (t_n, g(\phi_i,t_n))$$

The goal is to predict $f(\theta_i)$ and $g(\phi_i)$ at time $t_p$ (sometime in the future). Using the above data, an approximation function will be run through each set of coordinate pairs for $\theta_i$ and $\phi_i$. These functions will be constructed via least squares approximation, which creates a function that minimizes the distance between itself and the given data points (see Appendix A for details). The approximate functions are denoted

$$\hat{f}_i(t), \quad \hat{g}_i(t)$$

and are indexed by $i$ (one for each $f(\theta_i)$ and $g(\phi_i)$ set of data). These functions can then be used to extrapolate the function values at a new point in time $t_p$ where a reinstantiation

needs to occur. The desired function values are: $\hat{f}_i(t_p), \hat{g}_i(t_p)$ with $i = 1..N$. To create the PDF of the $\theta$ angle data at time $t_p$, the coordinate pairs $(\theta_i, \hat{f}_i(t_p))$ for $i = 1..N$ will be interpolated using cubic splines. The same is done for the $\phi$ angle using coordinate pairs $(\phi_i, \hat{g}_i(t_p))$. These PDFs are used to generate angle data for the new filaments. Similar procedures are done to extrapolate forward in time, the mean and variance of the strains in each angle bin, as well as the total energy of the system. Figure 5.19 shows a schematic of the PDF extrapolation concept.

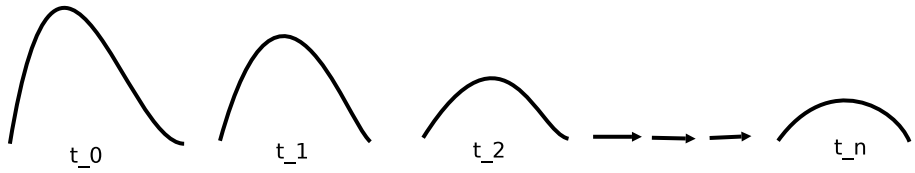


FIGURE 5.19. A schematic of PDF extrapolation forward in time.

To test this least squares extrapolation method, a network of filaments was created and placed under an extensional strain over $n$ small time steps. During each step, data was collected and after the $n$ steps, least squares functions were created as described above for the different variables. A future point in time $t_p$ was then chosen and PDFs for the variables were constructed based on the method described above. The reconstruction algorithm described in the previous section is then utilized to create a new network of filaments. This new network is then placed under an extensional strain for $m$ time steps. The hope is that the new network will behave similarly to how the original network would have behaved if it had been permitted to evolve to the same strain state. A full simulation of the original network was run for comparison purposes. Figure 5.20 shows the elasticity modulus of the original network in black and the elasticity modulus of different reconstructed networks in color (each colored line is a different new network). The three graphs differ in the size of the time interval between data collection and reinstantiation (denoted with yellow dots).

A convergence study was run on this data. The smaller the leap forward in time, the better the predictions should be. The error between the original elasticity modulus at a

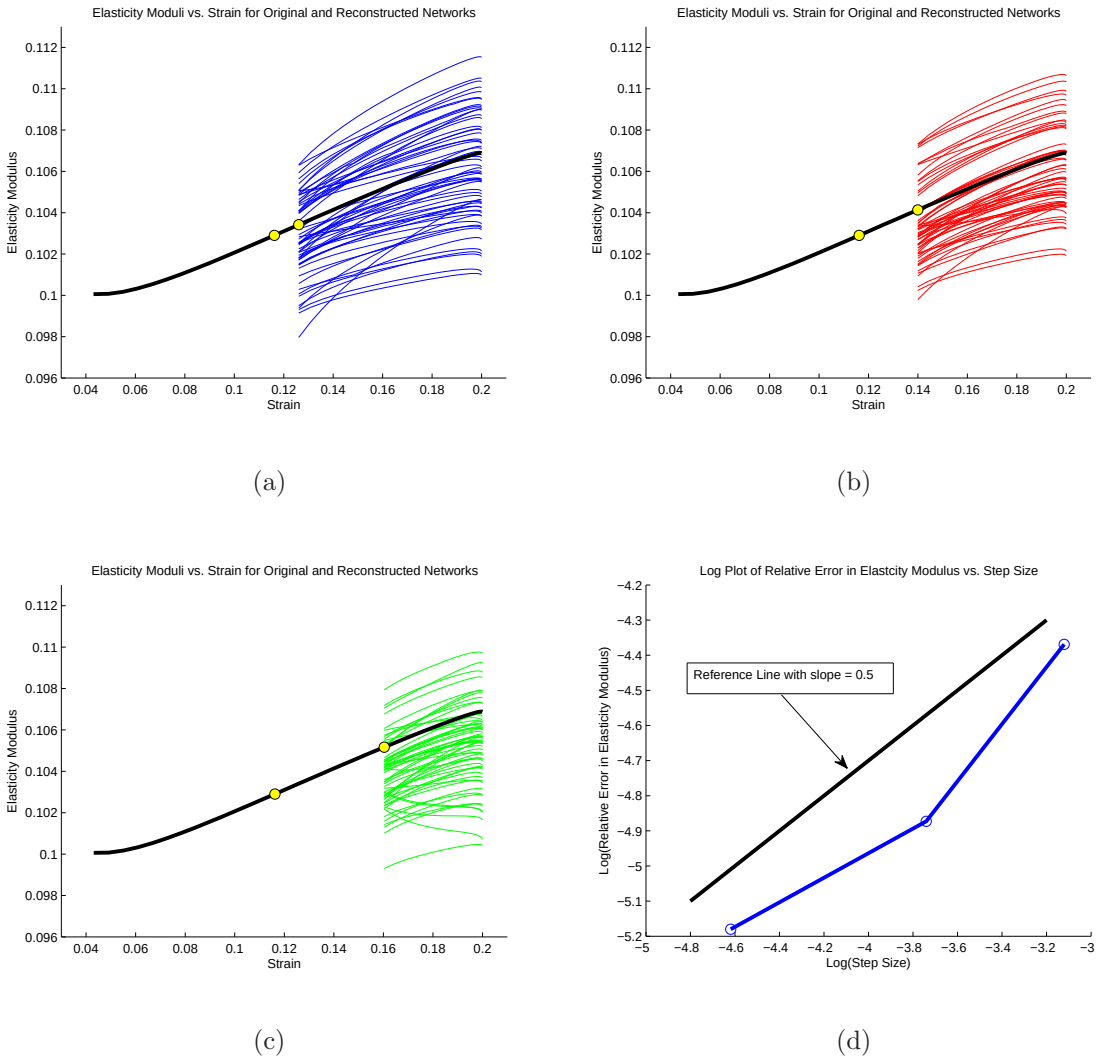


FIGURE 5.20. Three plots (a),(b),(c) depicting the original elasticity modulus vs. strain (in black) and the elasticity modulus vs. strain of reinstantiated networks (in color) after a leap forward in time has occurred. Each graph shows a different leap size (denoted by the two yellow circles). Graph (d) is a log-log plot of the relative error in elasticity modulus vs. the length of the time interval between data collection and network reconstruction.

later time and the average elasticity modulus from the new networks at the same later point in time should get larger as the network reconstruction occurs after larger time intervals. The average elasticity modulus of the reinstantiated networks was computed at instantiation and compared with the elasticity modulus of the original network at the same point in time. The result are shown in Figure 5.20 in a log-log plot of the relative

error in elasticity modulus versus the time interval size. The approximate slope is 0.54, which is in line with the expectations for this type of convergence study.

**5.2.1. Extrapolation Tests.** With the general framework for PDF extrapolation outlined above, the question now becomes one of algorithm efficiency. The goal of the microscopic simulation is to update the elasticity moduli for each continuum grid cell. These elasticity moduli will be used during the next continuum step. As seen in the one block example of Chapter 4, even during the microsteps the elasticity moduli are changing. If the next continuum step will take place from $t_n$ to $t_{n+1}$, a logical idea is to estimate the elasticity moduli from the microsteps at the midpoint in time, $t_{mid} = \frac{t_n + t_{n+1}}{2}$, of this next continuum interval. These new elasticity moduli are then utilized in the $t_n$ to $t_{n+1}$ step. The question is then, what is the minimum number of microsteps needed to get an accurate estimate of the elasticity moduli at time $t_{mid}$?

To answer this question, it is first helpful to look at the evolution of the elasticity moduli data. In the one block example of Chapter 4, the Young's modulus as the block is extensionally strained is shown in Figure 4.4. The block was strained by moving the corners of the block at constant velocities, thus the strain data on the $x$ axis can be replaced by time (Figure 5.21).

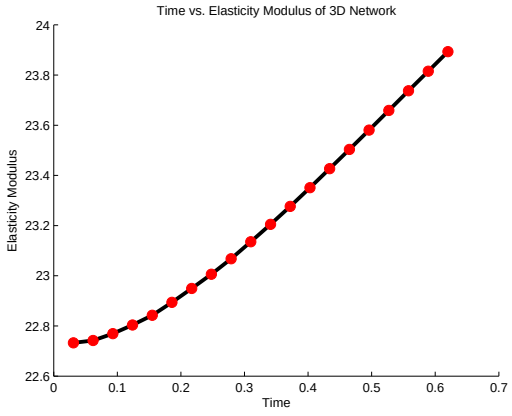


FIGURE 5.21. The Young's modulus vs. time curve for a box of crosslinked filaments under extensional strain

The data appears qualitatively linear. Suppose that the Young's modulus at the midpoint in time of the next continuum step is given by the last data point in Figure

5.21. How well can this value at time $t_{mid}$ be predicted by utilizing a subset of the data points in Figure 5.21 and a linear approximation function? Figure 5.22 shows least squares lines for varying numbers of data points and the predicted elasticity modulus at $t_{mid}$. The relative errors between the true value and the predicted value are shown in the table in Figure 5.2.1 along with a log-log plot of the number of steps versus the error. The choice in the number of data points to utilize will depend on the error one is willing to tolerate.

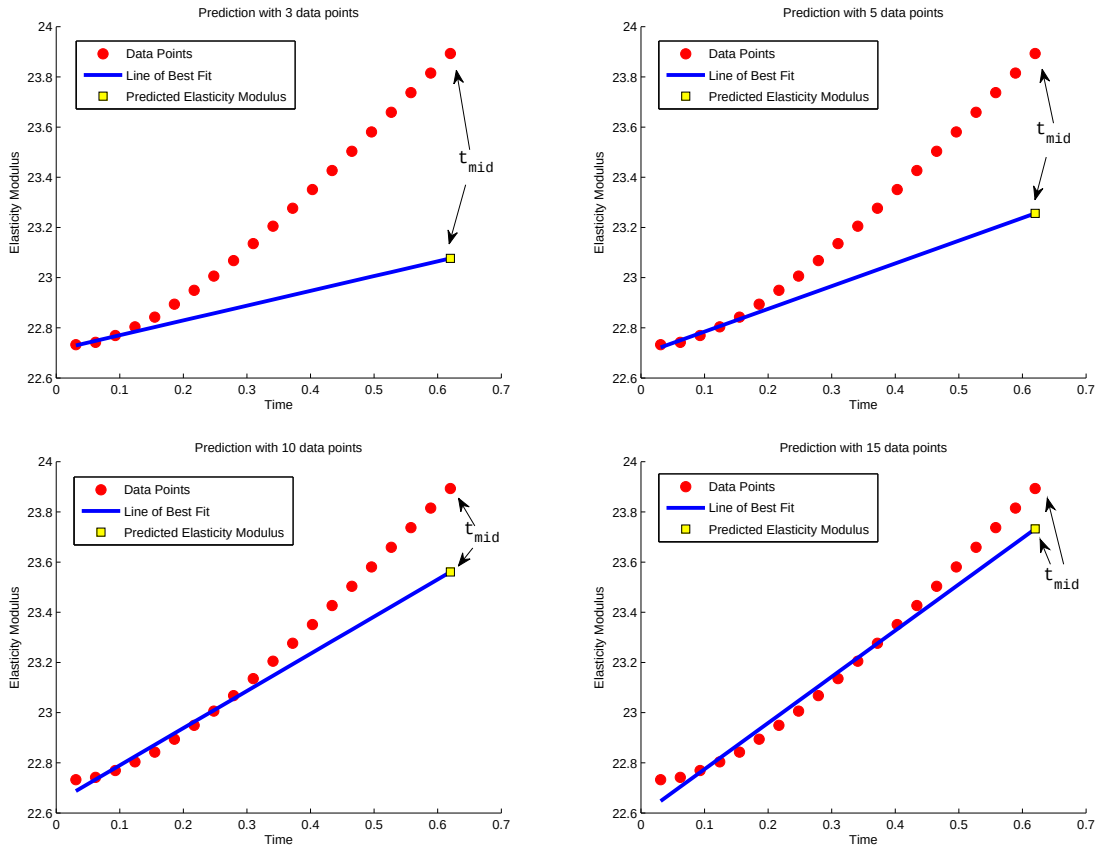


FIGURE 5.22. The true data points and linear least squares function constructed from 3, 5, 10, and 15 data points to predict the Young's modulus at $t_{mid}$.

A low number of microsteps may suffice for producing a good approximation to the elasticity moduli at $t_{mid}$, but this small amount of data may not be enough to extrapolate accurate PDFs for reinstantiation. To demonstrate this, a second test was run where data collected from varying numbers of microsteps was utilized in the extrapolation procedure

| Number of MicroSteps | Relative Error $(E - E_0)/E_0$ |
|---|---|
| 3 | 0.0353 |
| 5 | 0.0272 |
| 10 | 0.0141 |
| 15 | 0.00678 |
| 18 | 0.00402 |

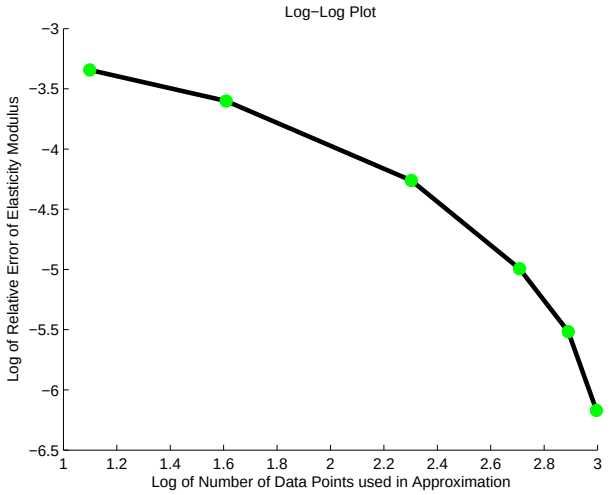


FIGURE 5.23. The relative error in Young's modulus found for increasing numbers of data points used to construct a linear least squares function to predict the Young's modulus at $t_{mid}$ in a log-log plot.

to predict the PDFs at time $t_p$. These PDFs were used to produce new networks. The average Young's modulus of the new networks at time $t_p$ was compared against the original network's Young's modulus at time $t_p$. The relative error versus number of microsteps used is shown in Figure 5.2.1 along with a log-log plot. In order to get the same level of accuracy in reinstantiation as the $t_{mid}$ elasticity modulus prediction, more data points must be included.

| Number of MicroSteps | Relative Error $(E - E_0)/E_0$ |
|---|---|
| 5 | 0.1682 |
| 10 | 0.0813 |
| 15 | 0.0615 |
| 20 | 0.0217 |

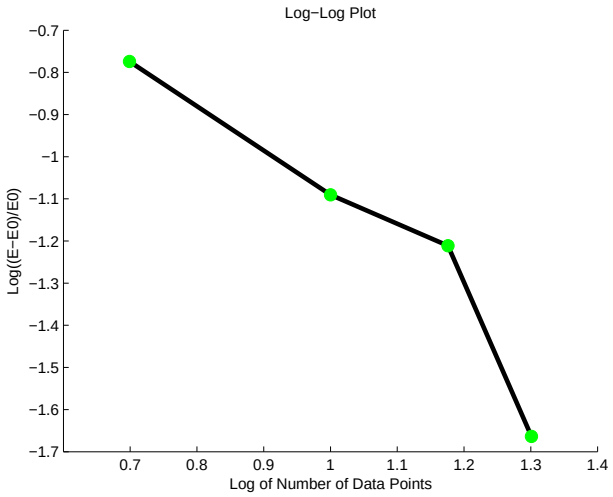


FIGURE 5.24. The relative error in Young's modulus found for increasing numbers of data points used in the PDF extrapolation algorithm to reinstantiate a new network at a future point in time in a log-log plot.

Different error tolerances may be deemed acceptable in the elasticity modulus prediction versus the network reinstantiation. Whichever error tolerance is more restrictive will determine the necessary number of microsteps. In this situation, it is important to have high accuracy for both approximations, however the accuracy of the elasticity modulus prediction becomes dependent on the network reinstantiation's accuracy after the first step. If large errors are incurred during network reinstantiation due to the use of a small number of microsteps, the elasticity modulus prediction during this next step will also likely have large errors. For these examples, it will therefore be a priority to make the network reinstantiation procedure as accurate as possible, while still maintaining computational efficiency. A number of microsteps $n$ will be chosen that is considered to be a reasonable compromise between the two goals.

CHAPTER 6

# Model Application

Now that the novel continuum-microscopic model and relevant biological background have been presented, the utility of the full algorithm will now be demonstrated. First, the three-dimensional, linear elasticity equations will be presented since these will be the equations that are solved at the continuum level. This will then be followed by simulations performed on a rectangular patch of model cytoskeleton.

## 6.1. Three-Dimensional Elasticity Equations

In order to describe the motion of the cytoskeleton as an elastic body, the three dimensional, linear elasticity equations will be utilized. The deformation the cytoskeleton experiences during a protrusive activity or during locomotion would be classified as large deformation as opposed to infinitesimal deformation. Generally, linear elasticity equations would not be valid in such a case. However, for these applications, the microscopic model provides locally valid elasticity moduli for each grid cell during each continuum step. These moduli will vary in both space and time, and will alter the linear elasticity equations being solved in each cell during each time step. In this format, the linear elasticity equations are being utilized in a local, piece-wise sense. The updated moduli from the microscopic model provide the non-linearity to this large deformation problem

The following derivation was modeled after the one found in Chapter 22 of [**50**]. Begin with the $3 \times 3$ stress and strain tensors:

$$\underline{\underline{\sigma}} = \begin{bmatrix} \sigma^{11} & \sigma^{12} & \sigma^{13} \\ \sigma^{21} & \sigma^{22} & \sigma^{23} \\ \sigma^{31} & \sigma^{32} & \sigma^{33} \end{bmatrix} \qquad \underline{\underline{\epsilon}} = \begin{bmatrix} \epsilon^{11} & \epsilon^{12} & \epsilon^{13} \\ \epsilon^{21} & \epsilon^{22} & \epsilon^{23} \\ \epsilon^{31} & \epsilon^{32} & \epsilon^{33} \end{bmatrix}$$

The strain tensor for small deformations is $\epsilon^{ij} = \frac{1}{2}\left(\frac{\partial \delta_i}{\partial x_j} + \frac{\partial \delta_j}{\partial x_i}\right)$ with $\delta_i$ the displacement and $x_j$ the direction. Equations of motion can be formulated from this definition by taking the time derivative of this equation and using the equality of mixed partials:

$$\begin{aligned} \epsilon_t^{11} - u_x = 0 \qquad & \epsilon_t^{12} - \frac{1}{2}(u_y + v_x) = 0 \\ \epsilon_t^{22} - v_y = 0 \qquad & \epsilon_t^{23} - \frac{1}{2}(v_z + w_y) = 0 \\ \epsilon_t^{33} - w_x = 0 \qquad & \epsilon_t^{13} - \frac{1}{2}(u_z + w_x) = 0 \end{aligned}$$

where $u = \frac{\partial \delta_1}{\partial t}, v = \frac{\partial \delta_2}{\partial t}, w = \frac{\partial \delta_3}{\partial t}$ are the velocities in the $x, y, z$ directions. We can obtain three more equations by using Newton's law, $F = ma$:

$$\begin{aligned} \rho u_t - (\sigma_x^{11} + \sigma_y^{12} + \sigma_z^{13}) &= 0 \\ \rho v_t - (\sigma_x^{21} + \sigma_y^{22} + \sigma_z^{23}) &= 0 \\ \rho w_t - (\sigma_x^{31} + \sigma_y^{32} + \sigma_z^{33}) &= 0 \end{aligned}$$

where $\rho$ is a density. So far, this provides a total of 9 equations for 15 unknowns (6 independent $\sigma$ values, 6 independent $\epsilon$ values, and 3 velocities), so 6 more equations are needed to close the system. This can be done by utilizing the linear stress-strain relationship: $\sigma^{ij} = C_{ijkl}\epsilon^{kl}$, with $C_{ijkl}$ containing two independent parameters, the Lamé coefficients $\lambda$ and $\mu$. The quantity $\mu$ is the shear modulus and $\lambda$ is relatable to the Young's modulus $E$ of the material by the following relationship:

$$\lambda = \frac{\mu(E - 2\mu)}{3\mu - E}$$

The values $E$ and $\mu$ are obtained from the microscopic model in the following way. The local values of $\sigma^{ij}$ and $\epsilon^{ij}$ are computed within each grid cell using the methods described in previous chapters and Appendix A. Then a least squares problem is used to estimate $E$ from the following three equations: $\sigma^{ii} = E\epsilon^{ii}$, $i = 1, 2, 3$. The $\mu$ value is then obtained

utilizing the relationship:

$$\mu = \frac{E}{2(1+\nu)}$$

where $\nu$ the material's Poisson ratio. The Poisson ratio can be found by the following formula:

$$\nu = -\frac{1}{2}\left(\frac{tr(\epsilon)E}{tr(\sigma)} - 1\right)$$

where $tr(\epsilon) = \epsilon^{11} + \epsilon^{22} + \epsilon^{33}$ and $tr(\sigma) = \sigma^{11} + \sigma^{22} + \sigma^{33}$. The stress-strain relationship looks like:

$$\begin{bmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{33} \\ \sigma^{12} \\ \sigma^{23} \\ \sigma^{13} \end{bmatrix} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{bmatrix} \begin{bmatrix} \epsilon^{11} \\ \epsilon^{22} \\ \epsilon^{33} \\ \epsilon^{12} \\ \epsilon^{23} \\ \epsilon^{13} \end{bmatrix}$$

These six equations complete the system of equations for the 15 unknowns. These six equations can be used to eliminate either the stress or strain variables to reduce the system to 9 equations for 9 unknowns. For instance if strain is eliminated, the resulting

set of equations is:

$$\begin{aligned}
\sigma_t^{11} - (\lambda + 2\mu)u_x - \lambda v_y - \lambda w_z &= 0 \\
\sigma_t^{22} - \lambda u_x - (\lambda + 2\mu)v_y - \lambda w_z &= 0 \\
\sigma_t^{33} - \lambda u_x - \lambda v_y - (\lambda + 2\mu)w_z &= 0 \\
\sigma_t^{12} - \mu(u_y + v_x) &= 0 \\
\sigma_t^{23} - \mu(v_z + w_y) &= 0 \\
\sigma_t^{13} - \mu(u_z + w_x) &= 0 \\
\rho u_t - (\sigma_x^{11} + \sigma_y^{12} + \sigma_z^{13}) &= 0 \\
\rho v_t - (\sigma_x^{21} + \sigma_y^{22} + \sigma_z^{23}) &= 0 \\
\rho w_t - (\sigma_x^{31} + \sigma_y^{32} + \sigma_z^{33}) &= 0
\end{aligned}$$

This system of equations can be written in conservation form as;

$$\mathbf{q}_t + A\mathbf{q}_x + B\mathbf{q}_y + C\mathbf{q}_z = 0$$

with $\mathbf{q} = [\sigma^{11}, \sigma^{22}, \sigma^{33}, \sigma^{12}, \sigma^{23}, \sigma^{13}, u, v, w]$, and $A, B, C$ matrices of dimension $9 \times 9$. Wave propagation techniques developed by LeVeque in [**50**] can then be applied to evolve the $q$ variables forward in time.

### 6.2. Patch Simulations

As a test of the full algorithm, a rectangular area of cytoskeleton will be simulated. The size of the portion of cytoskeleton will be $10\mu m \times 100\mu m$ and will be $1\mu m$ thick, which is a reasonable size for an area of the cytoskeleton of a large cell (for instance a neuron).

This rectangular patch of cytoskeleton is discretized into 1000, $1\mu m \times 1\mu m \times 1\mu m$ grid cells. These will be the continuum level grid cells upon which the elasticity equations

will be solved. This cytoskeleton will be placed under an extensional strain. The left side of the cytoskeleton is attached to a fixed wall and a sine wave stress will be applied uniformly on the right end. The top and bottom sides are free to move (Figure 6.1). These computational experiments can be likened to physical experiments done with optical tweezers on a real cytoskeleton.

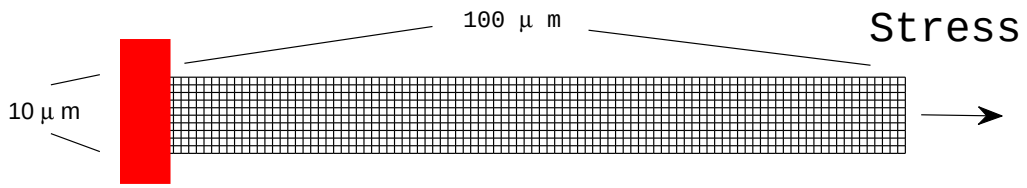


FIGURE 6.1. Discretized $10\mu m \times 100\mu m \times 1\mu m$ portion of cytoskeleton, attached at the left to a wall and under an extensional strain on the right.

**6.2.1. Simple Demonstration.** As a control case, the deformation of the cytoskeleton is first modeled only with the continuum equations where $E$ and $\mu$ are constants over space and time. The left column of plots in Figure 6.2 shows $\sigma_{xx}$ at several time steps during this continuum simulation. The right column of Figure 6.2 shows $\sigma_{xx}$ at the same time slices as the control case. However in this case, one of the 1000 grid blocks is simulated with the continuum-microscopic algorithm and has its elasticity moduli updated by the microscopic model. The block (denoted $B$) was chosen near the right edge of the cytoskeleton so that any changes in stress that result from different elasticity moduli would appear quickly in the simulation (Figure 6.3).

At the start of each continuum step, a microscopic network is constructed within block $B$ utilizing the PDF extrapolation and reinstantiation techniques described in the previous chapter. This network is then deformed for a short number of micro-time steps and new elasticity moduli are computed from the data. These elasticity moduli are then utilized in the continuum level advancement of the elasticity equations for this grid cell. Spring constants for the filaments in the network were chosen so that the elasticity moduli of block $B$ come out to be roughly one order of magnitude smaller than the $E$ and $\mu$

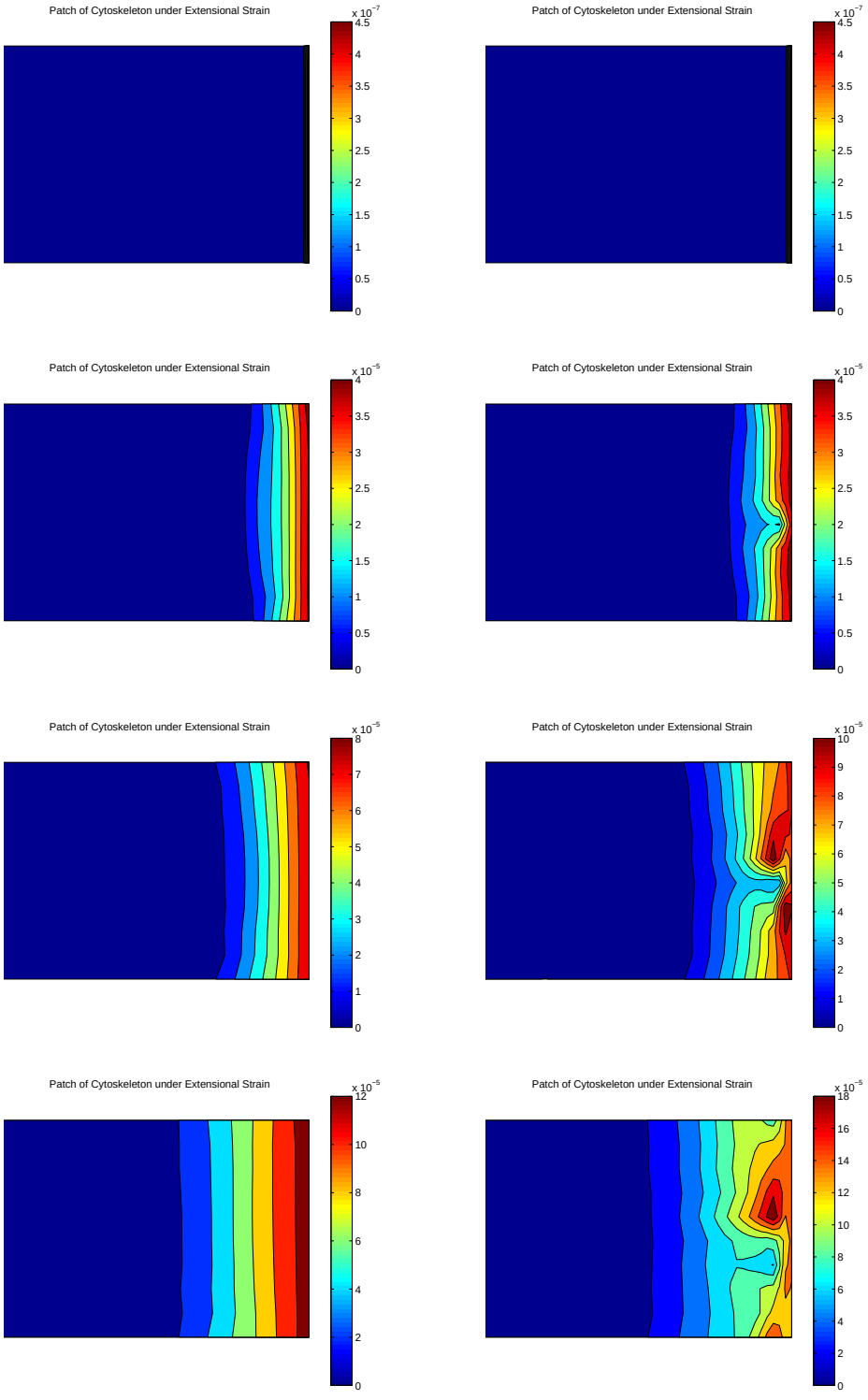


FIGURE 6.2. The left column shows time plots of $\sigma_{xx}$ in the cytoskeleton with elasticity moduli $E$ and $\mu$ constant in space and time. The right column shows the same time slices of $\sigma_{xx}$ with elasticity moduli $E$ and $\mu$ constant in space and time, except for one block whose moduli were determined by the microscopic model.

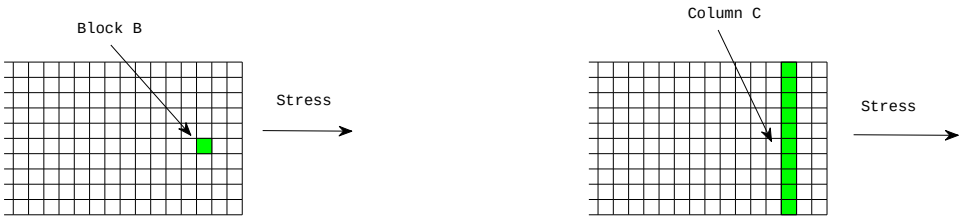


FIGURE 6.3. The right end of the cytoskeleton patch with block $B$ highlighted in green in the left picture and Column $C$ highlighted in green in the right picture. For the one block example, block $B$'s elasticity moduli will be updated via the microscopic algorithm, while the rest of the grid cells have constant $E$ and $\mu$ values. The same is done for the blocks in Column $C$ in the second example.

constant values in the rest of the domain. This was done so that the differences would be visible in the stress plots. Lower moduli mean that waves travel slower through block $B$. Forces do not propagate as fast through block $B$ as through its neighbors, resulting in a different stress field.

In a real cell, an area with a lower elasticity modulus could be explained physically in several ways. It could be an area with a lower number of crosslinks, perhaps due to a lack of crosslinking protein monomers in the interstitial fluid. It could also be an area that has a higher concentration of filament severing proteins that depolymerize the actin fibers at a faster rate, creating a gap-filled, looser network with a lower stiffness response.

**6.2.2. One Block Test.** Consider the same patch of cytoskeleton, again where the continuum equations will be solved everywhere utilizing constant $E$ and $\mu$ values except in block $B$ (where the microscopic model will update the moduli). Suppose one continuum step is of length $dt_{cont}$ and that one microstep is of length $dt_{micro} = dt_{cont}/m$ where $m$ is the number of microsteps in one continuum step. The goal of the continuum-microscopic algorithm is not to evolve the microscopic network all $m$ microsteps, but rather to evolve it a shorter number of steps $n$, $(n \ll m)$ and extrapolate the rest of the data up through

microstep $m$. The hope is that a network constructed from the extrapolated data at the next continuum step will be similar to the original network if it had been simulated over the full continuum step. To test this, the continuum-microscopic algorithm that takes $n$ microsteps to update block $B$'s moduli (denoted for short as case MR = Microscopic Reinstantiation) will be compared against results from a test where the original network in block $B$ was evolved through the whole simulation. This case will be denoted as the FM (Full Microscopic) case. The same spring constant values utilized in the initial demonstration will be used here (again so that differences will be more visible). Plots of $\sigma_{xx}$ are shown in Figure 6.4 with left and right column graphs depicting the FM and MR cases respectively at the same time slices. Qualitatively the results look similar however there are some small differences.

One can make a quantitative comparison between the two cases by looking at how the elasticity moduli change over time in block $B$ in each simulation. Figure 6.5 shows the Young's modulus of block $B$ during each continuum step for the FM (in black) and MR (in red) cases. Figure 6.6 shows the relative error in Young's modulus between the two cases over each continuum step.

There are several things to note from Figure 6.5. First, in general there is more variability of the Young's modulus in the MR case than the FM case, which is to be expected. The MR case has a different network reinstantiated at each continuum step, while the FM case follows the deformation of the original network through the full simulation. As seen from the data in Figure 5.18 the elasticity moduli of the reinstantiated networks can vary, even though their average is close to the original network's value. This variability is due to factors such as different connectivity of crosslinks and the changing number and locations of filaments attached to the walls. Unlike Figure 5.18, Figure 6.5 shows only one MR path over time. Because there are some random elements in the reinstantiation procedure, other runs with the same starting data will produce different elasticity moduli paths. The variability in the data points in these paths will increase with each continuum step (after each reinstantiation). The increase in variability is quantitatively visible if

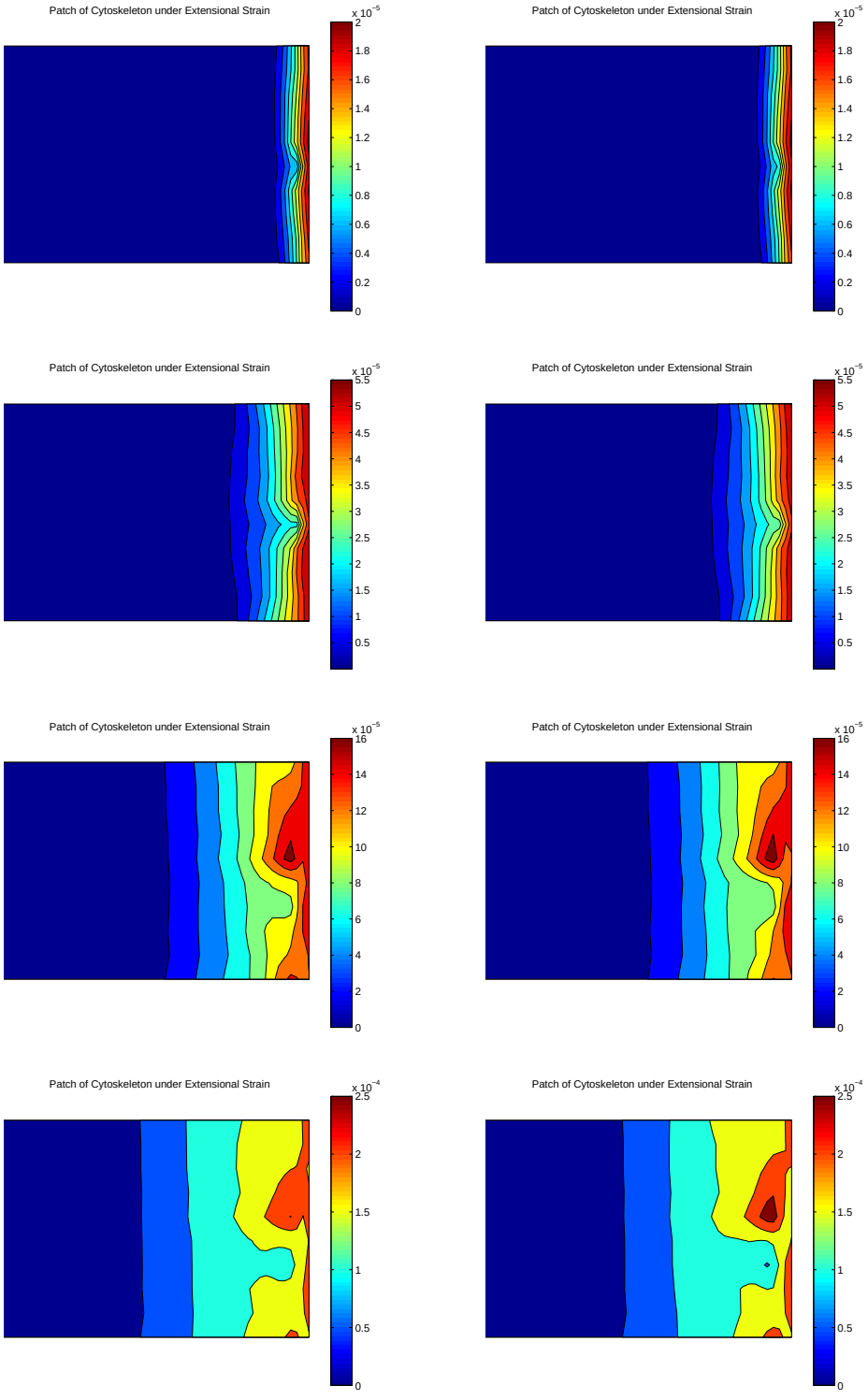


FIGURE 6.4. Time plots of the $\sigma_{xx}$ stress in the cytoskeleton with elasticity moduli $E$ and $\mu$ constant in space and time, except for block $B$ whose moduli were determined by the microscopic model. The left column is the FM case and the right column is the MR case.

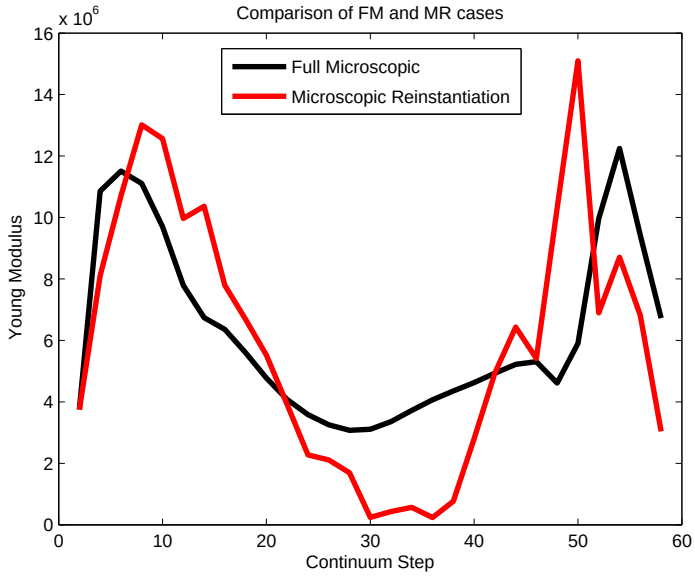


FIGURE 6.5. The Young's modulus of block $B$ during each continuum step for the FM (black) and MR (red) cases.

one looks at the relative error in Young's modulus (shown in Figure 6.6) between the MR and FM cases of Figure 6.5. The error is generally smallest during the first few continuum steps (errors in the range of $5-10\%$), but they become larger as the computation continues. The slope of a regression line through the data in Figure 6.6 is approximately 0.015 (if the two extreme relative error points are removed). This means that the relative error in Young's modulus increases by roughly 1.5% after each continuum step.

For these examples, the predicted data utilized in the construction of a new network at the start of each continuum step is based only on the previous network's data. This could be altered to include more past networks' info which would presumably improve the model's accuracy. (This is something to be investigated in future work). Also, it is important to note the elasticity moduli affect the speed of the elastic waves passing through the medium. Small variations in the elasticity moduli that result from the reinstantiation procedure will lead to different wave speeds, which in turn will lead to different boundary conditions for the grid cell during the next time step. These different boundary conditions are then used in the reinstantiation procedure. The grid cell in the FM and MR cases at the same continuum time step may not be in the exact same

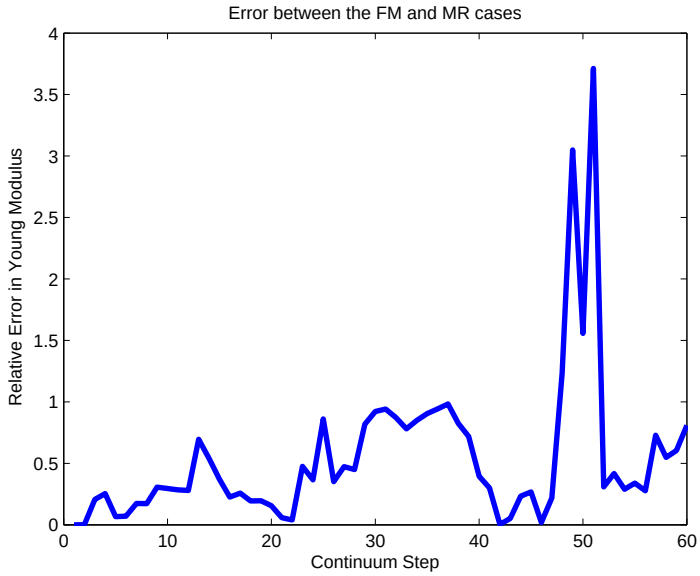


FIGURE 6.6. The relative error in Young's modulus of block $B$ between the FM and MR cases during each continuum step.

continuum state. Also as mentioned earlier the filaments in block $B$ were assigned spring constants that produced elasticity moduli well below the $E$ and $\mu$ values of the rest of the grid cells. Any small initial differences in the FM and MR cases are likely to be magnified as the computation goes on due to the large differences between this cell and its neighbors. This aspect will be corrected in the column example.

Having discussed these possible error sources, if one takes a qualitative look at Figure 6.5, the general shape of the Young's modulus curves in the MR and FM cases is quite similar. Both curves start off with an initial concave down section that then levels out and rises to another peak towards the end. The largest relative errors of approximately $300-400\%$ which occur around continuum steps 50 may seem unacceptably high. However looking at Figure 6.5, these relative errors occur at points where the second peak of the FM case is shifted over from the second peak of the MR case. Though the errors may seem large in a step by step comparison, the general behavior of the two cases is qualitatively similar and they produce qualitatively similar stress fields. A quantitative comparison of the stress ($\sigma_{xx}$) found in block $B$ over time for the FM and MR cases is shown in Figure 6.7, and the relative errors are shown in Figure 6.8. Overall the errors in stress are lower

than the errors in Young's modulus. Having lower errors in the continuum level variables is important since they determine the overall deformation state of the cytoskeleton.

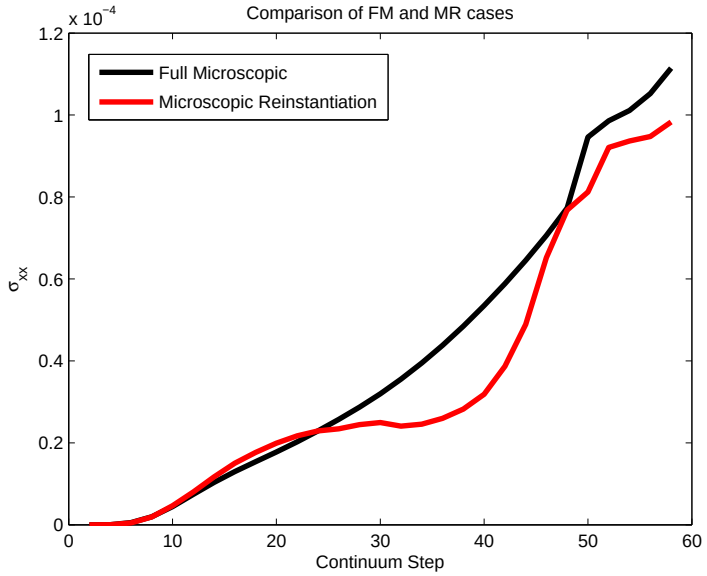


FIGURE 6.7. The relative error in Young's modulus of block $B$ between the FM and MR cases during each continuum step.

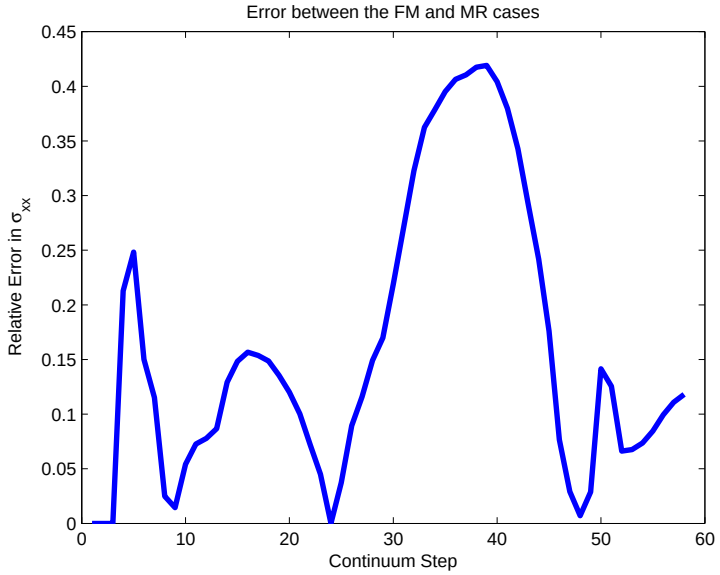


FIGURE 6.8. The relative error in Young's modulus of block $B$ between the FM and MR cases during each continuum step.

One last aspect to discuss about this first example is the computational time. The point of doing the continuum-microscopic model versus a full microscopic model is to

reduce computational expense. In these examples, the number of micro-time steps $m$ per continuum step was 25. The continuum-microscopic model performed $n = 10$ microsteps during each continuum step. The value of 10 was chosen based on the results of basic tests such as those conducted at the end of Chapter 5 that provide estimates of errors incurred by utilizing various numbers of microsteps. The MR simulation took approximately 4 hours to run, while the FM simulation took roughly double the amount of time at about 8 hours. The reason the MR simulation took half the time as opposed to 2/5 of the time (10 vs. 25 microsteps per continuum time step) is that the bisection algorithm of the reinstantiation procedure does take some computational time. This process equates to approximately 2 microsteps worth of time.

**6.2.3. One Column Test.** In this second example, the same patch of cytoskeleton receives the same extensional stress on its right end as in the one block example. In this case, the elasticity moduli will be computed with the microscopic algorithm for all the blocks in the highlighted column in Figure 6.3. Since the stress is applied uniformly across the right edge, these $B_i$ ($i = 1..10$) blocks in column $C$ (as they will be denoted) should generally experience the same stress/strain states at the continuum level with some differences arising due to edge effects at the top and bottom of the column. A microscopic network with the same number of filaments $N$ with uniformly distributed orientations will be laid in each block in column $C$ during the first instantiation. For this example, spring constants were assigned to the filaments that produce elasticity moduli close to the $E$ and $\mu$ values of the rest of the domain that match parameters found in the literature. The Young's moduli computed during this first step for the blocks in column $C$ is shown in Figure 6.9 on the left. The Young's moduli for the same column of blocks at a later point in time is also shown in Figure 6.9 on the right. The variance of the moduli at the later point in time is approximately an order of magnitude higher than at the beginning. This increase in variance is due to many of the same factors mentioned in the one block example. Small differences in the initial moduli will lead to differences in wave propagation through those cells which will alter the continuum variables. These

differences will then feed back into the reinstantiation process during the next step and create a wider variation in the new elasticity moduli.

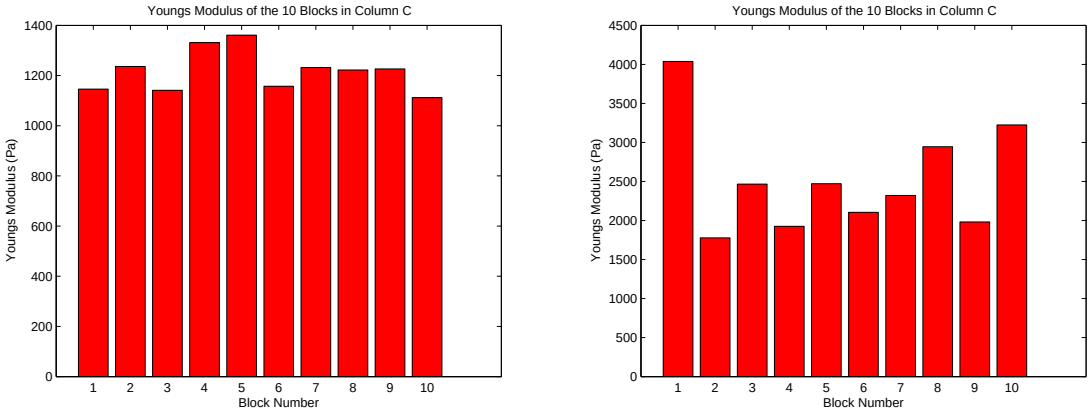


FIGURE 6.9. The Young's modulus of the 10 blocks in Column C at the beginning (left) and after several continuum steps (right).

As in the one block case, the stress field for $\sigma_{xx}$ will be compared for an MR and FM case. Time plots of $\sigma_{xx}$ for the two cases are shown side-by-side in Figure 6.10. The results are qualitatively similar. The average relative error in $\sigma_{xx}$ over the blocks $B_i$ in Column $C$ between the FM and MR case over time is shown in Figure 6.11. In general the errors are in the range of $5 - 10\%$ which is an improvement over the one block example.

As in the one block case, it is interesting to look at how the elasticity moduli of the MR case compare to those of the FM case. The left column of Figure 6.12 shows examples of the comparison in Young's moduli for four of the ten blocks in the column. The behavior is similar to that of the one block case in that there is more variability in the MR values than in the FM values, but overall the two curves take on the same general shapes in each block. The networks in the blocks in this column have the same potential error sources as the one block case. The relative errors in Young's modulus for these four example blocks over time are also shown in Figure 6.12 in the right column.

Relative errors over 1 are not present which is an improvement over the one block case. The errors are still in the 50% range, however as mentioned in the one block case this is the step-by-step comparison which may be unfair to examine since the blocks in the FM and MR cases may be in slightly different strain states stemming from initial variations in the

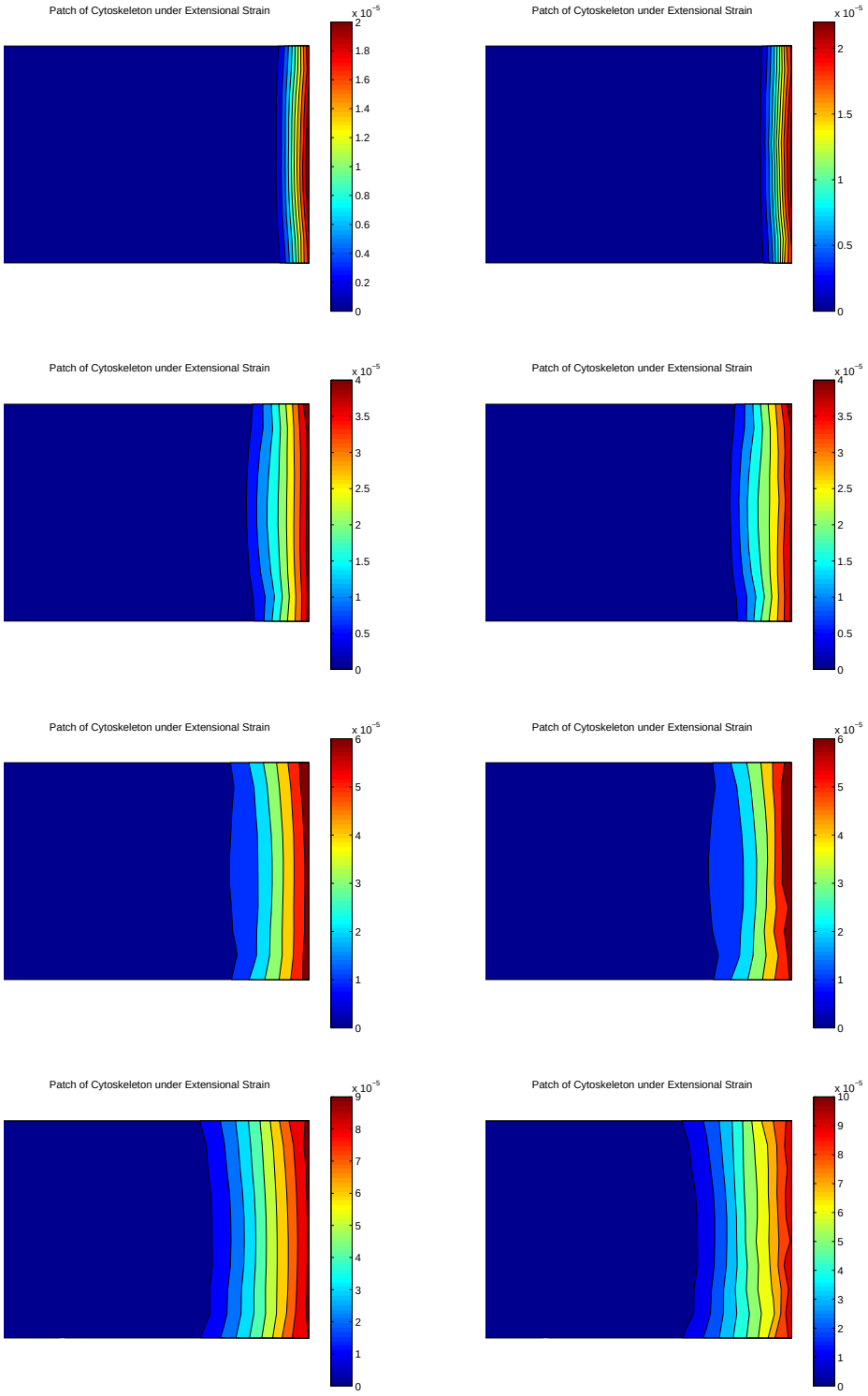


FIGURE 6.10. Time plots of $\sigma_{xx}$ in the cytoskeleton with elasticity moduli $E$ and $\mu$ constant in space and time, except for the blocks in Column $C$ whose moduli were determined by the microscopic model during each time step. The left column shows the FM case while the right column shows the MR case at the same time slices

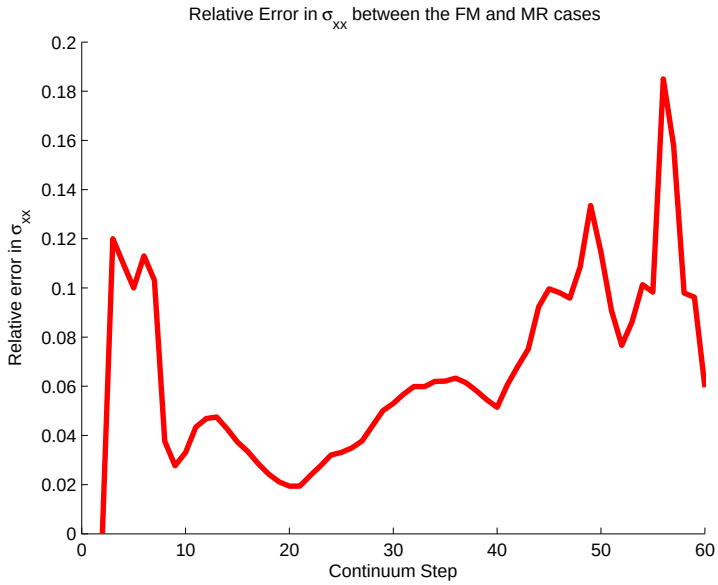


FIGURE 6.11. The average relative error in $\sigma_{xx}$ of the blocks in column $C$ between the FM and MR cases during each continuum step.

moduli. Regression lines through these relative errors have slopes between $0.002 - 0.006$. With each continuum step the relative errors increase approximately $0.2 - 0.6\%$.

As in the one block case, there is an approximately 50% reduction in computational time with the MR simulation versus the FM simulation. The MR simulation took approximately 40 hours to run in serial mode and the FM simulation took about 80 hours. The code was also run with parallelization. Because the microscopic blocks deform individually during the microsteps, they can be evolved simultaneously on parallel processors. This algorithm performs the microscopic advancement on the ten blocks in Column $C$, thus ideally the code should be able to run about ten times faster in parallel mode. The code with microscopic reinstantiation that took about 40 hours to run in serial mode, took approximately 14 hours in parallel mode, which is about three times faster. The main reason for this is that, the parallel code can only go as fast as the slowest microscopic advancement. If one block takes longer (more iterations) to reach a state of minimal energy than the rest of the blocks, the parallel code must wait for this minimization to finish before it can move back to the continuum solver. The speed up should improve for cases with more blocks where more processors are used.

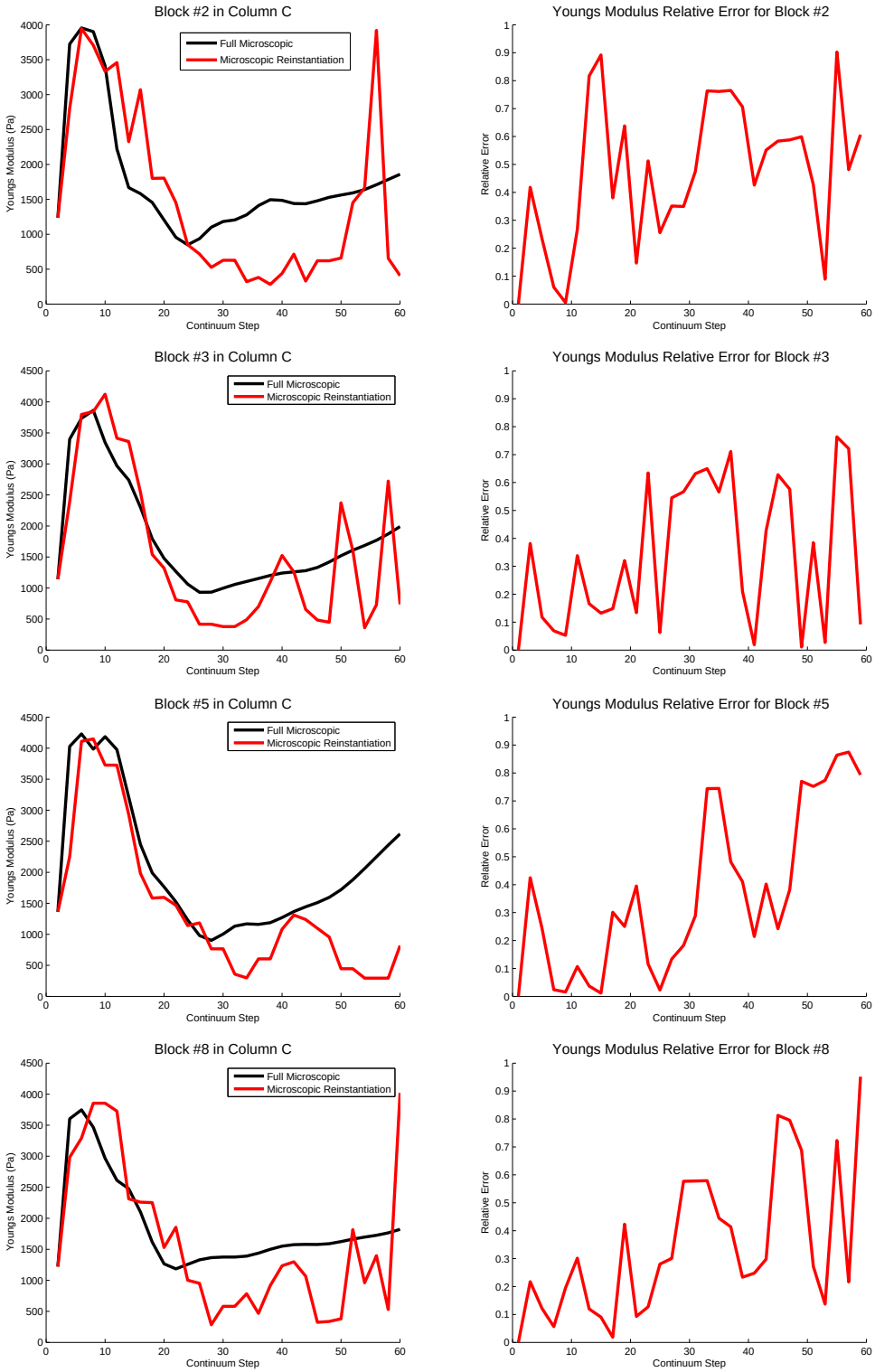


FIGURE 6.12. The left plots show the Young's modulus of blocks 3, 4, 8 and 9 of Column $C$ for the FM (in black) and MR (in red) cases during each continuum step. The right plots show the relative error in Young's modulus of blocks 2, 3, 5 and 8 of Column $C$ between the FM and MR cases during each continuum step.

The continuum-microscopic algorithm could be sped up further by utilizing a sampling technique to minimize the number of necessary reinstantiations. For example, if grid cell $i, j$ experiences no change in its strain state from continuum step $t_n$ to continuum step $t_{n+1}$, the elasticity moduli should also not change. Doing a microscopic reinstantiation during this time step wastes computational time and could also introduce error since the new network will likely have slightly different elasticity moduli than the original. The elasticity moduli found for time step $t_n$ should be used for time step $t_{n+1}$. There are several ways that sampling could be introduced into the continuum-microscopic algorithm. One possibility is to simply check the differences in strains from $t_n$ to $t_{n+1}$. If the magnitude of the change is above a given threshold then microscopic reinstantiation and computation of new elasticity moduli should take place. If the change is too small, this grid cell will be skipped and its elasticity moduli will remain the values they were at $t_n$. Another way of implementing the same idea is to utilize an idea similar to the Adaptive Mesh and Algorithm Refinement (AMAR) of Garcia et al. [**28**]. The microscopic algorithm is only utilized in regions where fine grids are required to resolve the solution. The continuum-microscopic algorithm presented in this thesis has been coded in the Bearclaw framework [**58**], and includes adaptive mesh refinement capabilities. If mesh refinement is required in an area of the domain to better resolve the variables (due perhaps to shock waves or other large differences in variable values between adjacent grid cells), then this grid cell will be flagged to be run through the microscopic algorithm in order to update its elasticity moduli. In this thesis work, the first technique will be tested. The method involving adaptive mesh refinement will be explored in future work.

The same one column simulation will now be run with the sampling technique described above. If the strain state of block $B_i$ in column $C$ does not change significantly from $t_n$ to $t_{n+1}$ then the elasticity moduli computed at $t_n$ will be utilized in the continuum advancement at $t_{n+1}$. Plots of the Young's modulus of the MR with sampling (MRS) versus the FM case for the same four blocks as Figure 6.12 are shown in Figure 6.13 along with plots of the relative error between the two cases. The FM, MR, and MRS

time plots of the Young's modulus for the four blocks are shown all together in the plots in Figure 6.14.

Although the relative errors may not have changed very much between the MR and MRS cases, doing the sampling does appear to provide some qualitative improvements. The MRS case has less variability than the MR case. The trends in the MRS case appear more stable, and are qualitatively similar to the FM case. The average relative error in $\sigma_{xx}$ for these ten blocks over time with the MRS case is shown in Figure 6.15. In general the errors tend to be smaller for the MRS case, especially later on in the simulation, which is likely a result of the more stable elasticity moduli that it produces.

Utilizing the sampling method also offers a reduction in computational expense. The MR code which took 14 hours in parallel mode, took approximately 10 hours to run in parallel with sampling.

## 6.3. Full Simulation

For the full simulation, all blocks in the domain had their moduli updated by the microscopic reinstantiation (with sampling) algorithm. An FM simulation was also carried out for comparison purposes. Figure 6.16 shows plots of the Young's modulus over time for the MRS and FM cases of six blocks in the domain. As in the one block and one column case, the trend of the curves is similar. Relative errors in the Young's modulus (though not shown) look similar to the one column case.

The average relative error in stress $\sigma_{xx}$ taken over a subset of the blocks is shown in Figure 6.17. These errors are again in the 10% range. A comparison of $\sigma_{xx}$ between the MRS and FM simulation for an example block is also shown in Figure 6.17.

What is important to take away from these examples is the following: (1) variability is present in the simulations with microscopic reinstantiations, however the general trend in the evolution of the Young's moduli is similar, (2) the continuum-level variables such as $\sigma_{xx}$ show significantly smaller errors than the moduli themselves which is important since these variables determine the overall state of the cytoskeleton, (3) errors in the

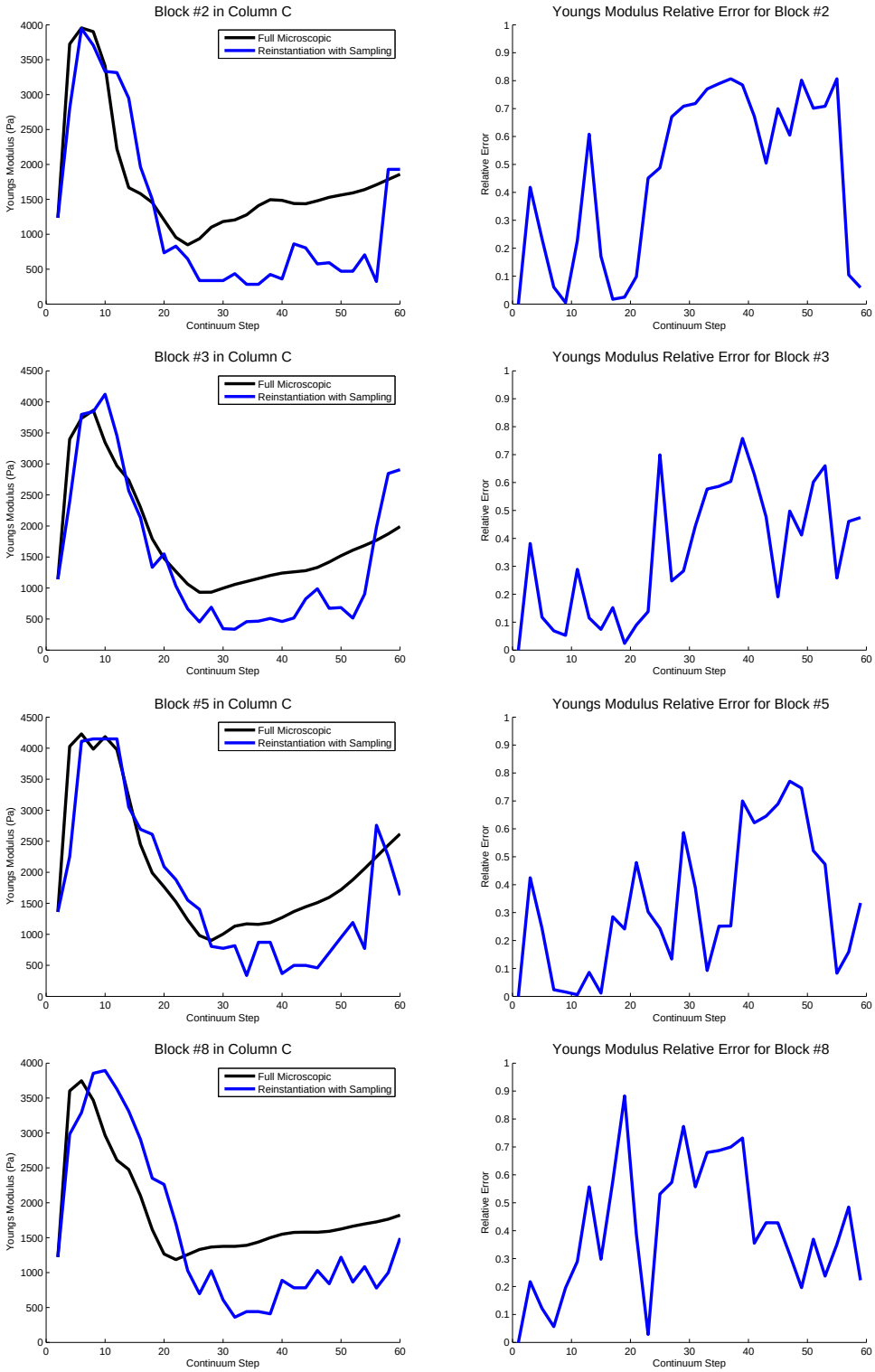


FIGURE 6.13. The left plots show the Young's modulus of blocks 2, 3, 5 and 8 of Column $C$ for the FM (in black) and MRS (in blue) cases during each continuum step. The right plots show the relative error in Young's modulus of blocks 2, 3, 5 and 8 of Column $C$ between the FM and MRS cases during each continuum step.

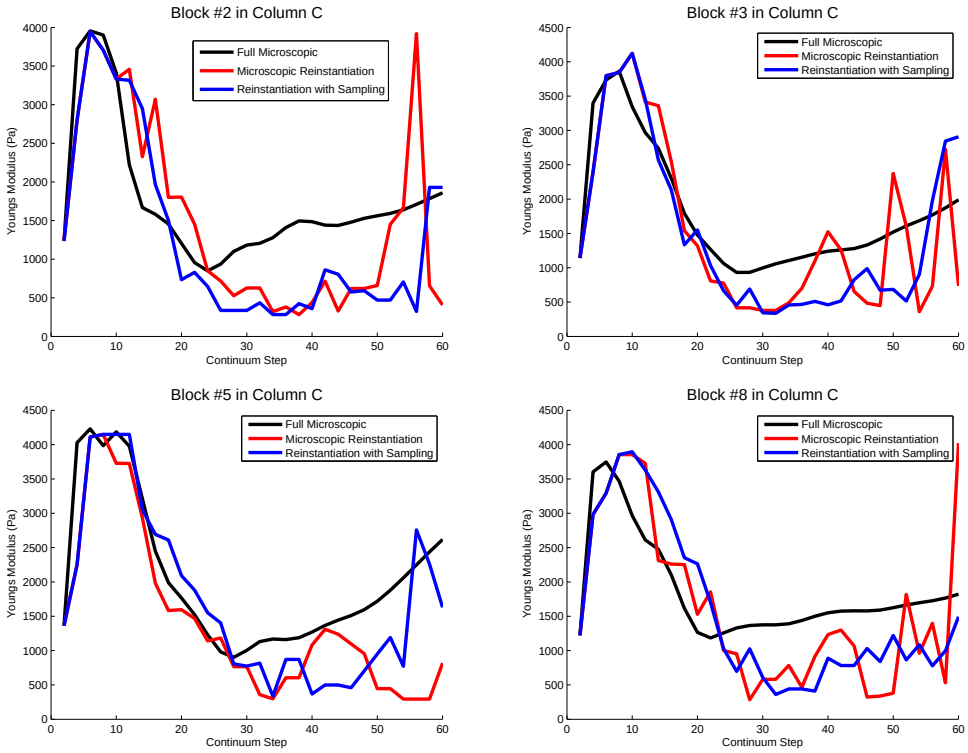


FIGURE 6.14. The plots show the Young's modulus of blocks 2, 3, 5 and 8 of Column $C$ for the FM (in black), MR (in red), and MRS (in blue) cases during each continuum step.

range of $5-10\%$ for the continuum variables are an acceptable trade-off for a savings of $50-75\%$ of the computational time.

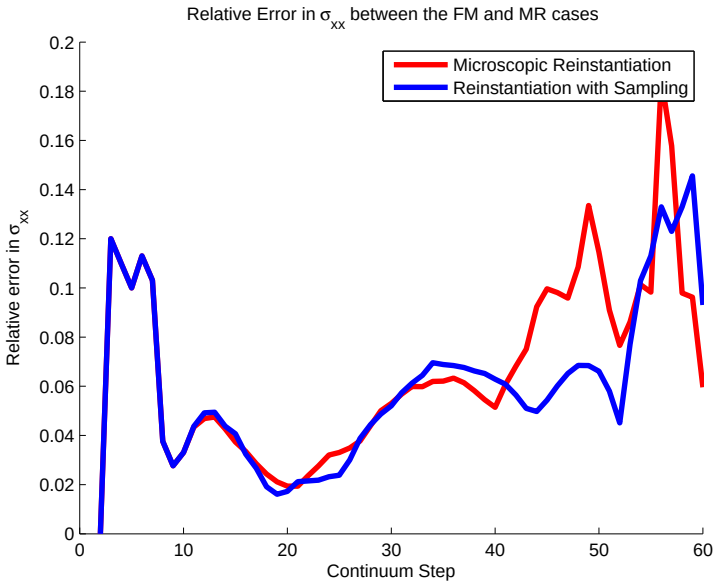


FIGURE 6.15. The average relative error in $\sigma_{xx}$ of the blocks in column $C$ between the FM and MRS cases during each continuum step.

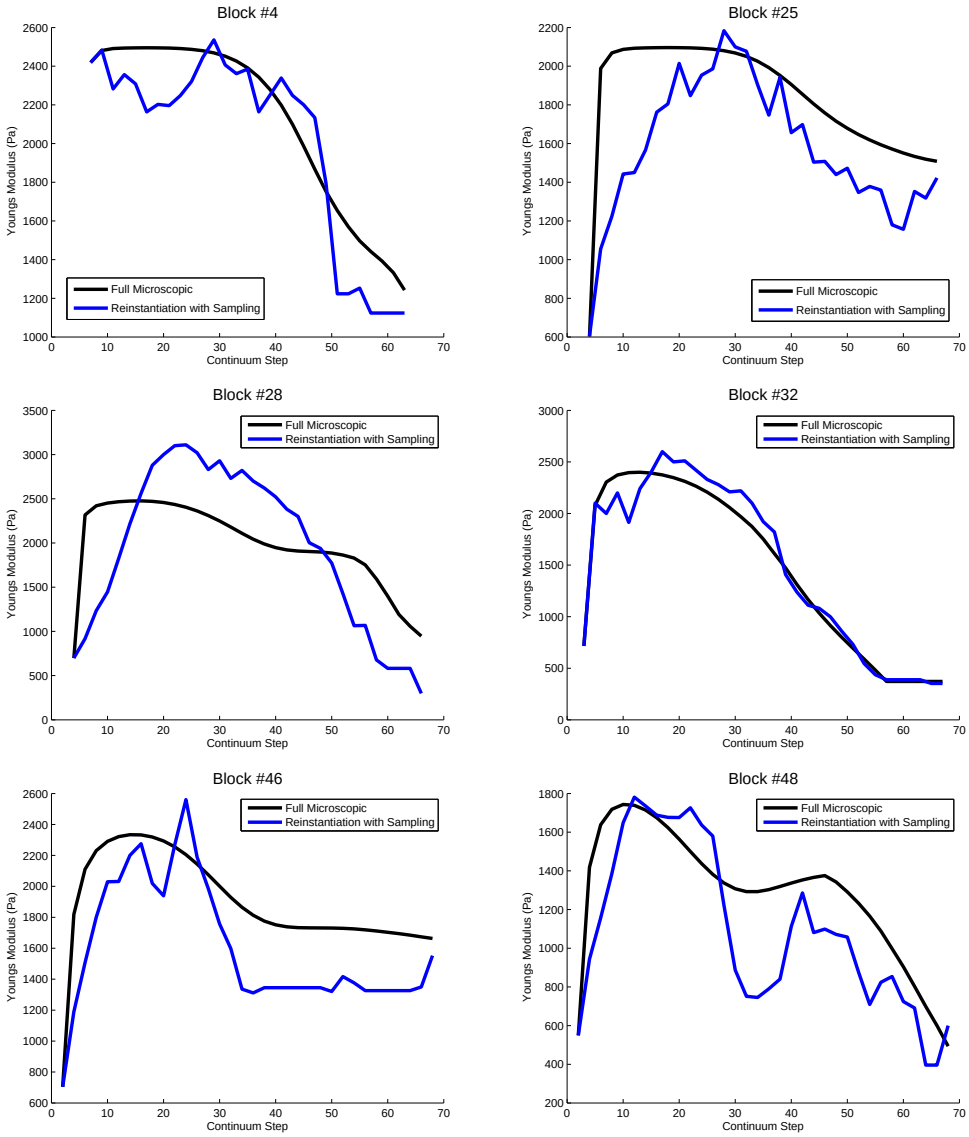


FIGURE 6.16. The Young's modulus of six blocks for the FM (in black) and MRS (in blue) cases during each continuum step.

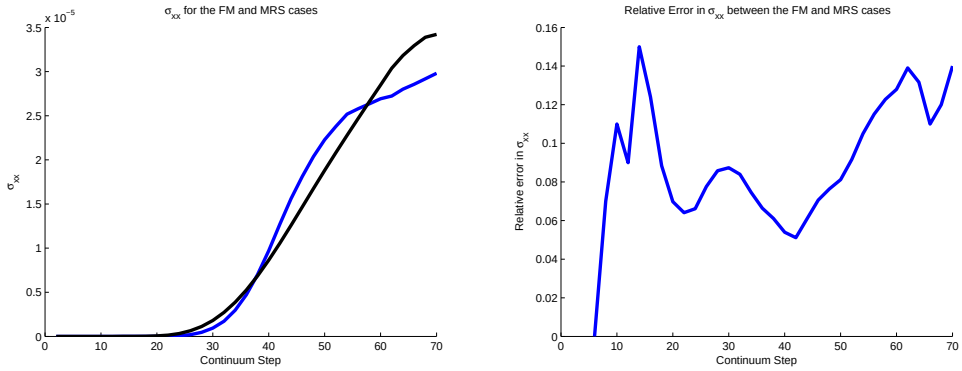


FIGURE 6.17. A comparison of the evolution of $\sigma_{xx}$ in the FM and MRS case. The average relative error in $\sigma_{xx}$ over a subset of blocks in the domain.

CHAPTER 7

# Conclusion

This research effort has focused on the development of a novel continuum-microscopic algorithm to model biological materials undergoing deformation. The work in this thesis will now be summarized. This will be followed by a discussion of the main contributions this reseach provides in the field of computational biomechanics and the conclusions that can be drawn from the presented results. Finally the chapter will conclude with a section on future work that will be done to further improve and extend the research.

## 7.1. Summary

The goal of this reseach was to examine the problem of modeling the cytoskeleton of an animal cell. At the whole-cell level the cytoskeleton is often modeled as a continuous medium, however its microstructure reveals a complex network of crosslinked filaments. This structure is heterogeneous in space and in time and thus its elastic properties also vary with space and time. This means that the tensor $C_{ijkl}$ in the constitutive law $\sigma_{ij} = C_{ijkl}\epsilon_{kl}$ which closes the elasticity equations utilized to model the cytoskeleton as continuous media cannot be assumed to be constant. The microscopic structure of the cytoskeleton must play a role in modeling this material's deformation. Developing an algorithm that can perform this task was the goal of this thesis work.

Chapter 1 began with a brief history of the development of continuum mechanics, and the notions of stress and strain. The derivation of several basic constitutive relationships for homogeneous objects such as a string, beam, membrane and plate was then presented. These equations were straightforward to develop due to the material's assumed homogeneity. These derivations are not so simple when the material is heterogeneous,

because the constitutive laws are no longer spatially independent. As a simple introduction, a constitutive relationship was derived for a composite material. Composites still have a regular pattern to their heterogeneity and thus homogenization techniques are applicable. In materials like the cytoskeleton, this is not the case. If a continuous model of the cytoskeleton is abandoned in favor of a full microscopic model, one quickly determines that such a model is computationally intractable. A class of methods that has been developed to try to address this issue is known as continuum-microscopic (CM) models. Examples include the Heterogeneous Multiscale Method [**25**], Adaptive Mesh and Algorithm Refinement [**28**], and the Equation-Free Method [**43**]. The basic idea behind these algorithms is to have two physical scales (continuum and microscopic) that are governed by two different physical laws (e.g. Navier-Stokes and molecular dynamics). These methods try to capitalize on the speed of a continuum level simulation while still including microscopic information. The equations at the micro-scale are advanced a short number of time steps, and data collected from this evolution is used to update parameters at the continuous scale before those equations are advanced next. The benefit of a CM algorithm is that it is a computationally efficient way of incorporating microscopic information into the continuum level equations. An issue that has not been addressed by the CM models mentioned above is how to maintain an accurate representation of the microstructure throughout the simulation. Since the microscopic model is not being fully advanced during each continuum step, the question becomes how to predict the state of the microstructure at the beginning of the next continuum step. The CM models mentioned here utilize known families of distributions (e.g. Gaussian and uniform distributions) to instantiate microscopic states at future points in time. All information from previous micro-states is lost and no memory of the material's heterogeneous structure is retained. In a material like the cytoskeleton, the microscopic data is valuable for determining the elastic properties of the cell and retaining memory of the microstructure is necessary for capturing observed behaviors such as strain hardening. This research effort is focused on developing computationally efficient algorithms that address this issue.

Chapter 2 began with detailed background information on the biology of the cytoskeleton and the cellular activities it participates in. This was followed by a discussion of the computational modeling difficulties of performing a purely microscopic simulation of the entire cytoskeletal network in a cell during deformation. Because of the expense, microscopic models of the cytoskeleton are reserved for small patches of the network to understand how the cytoskeleton behaves in a local region of the cell. If a whole cell model is desired, the typical avenue is to use a continuum model of the cytoskeleton as a porous medium or highly viscous fluid, or to utilize a coarse grained network representation. A list of cytoskeleton modeling methods both continuous and discrete was presented in Section 2.4 along with the applications for which they were utilized. In Section 2.5, the difficulties with utilizing these methods for a whole cell model of cellular deformation is explained. The basic continuum-microscopic algorithm that will be used to model the cytoskeleton is then presented, with the main goal being to retain memory of the microstructure from continuum step to continuum step.

Chapter 3 provided the basic definitions and theorems from probability theory that were utilized in the development of the microscopic reconstruction algorithm. The microscopic algorithm requires PDF estimation and the extrapolation of PDFs forward in time. PDF estimation is a central problem in the field of statistics, and non-parametric estimation (required for this algorithm) is particularly challenging. Several different non-parametric PDF estimation techniques were presented and tested on a data set (whose underlying PDF was known) to try to determine which method would be the best to use for the microscopic algorithm. Kernel estimation and histogram interpolation produced the lowest errors and were thus chosen as two possible methods for the algorithm. Least squares approximation was chosen as the PDF extrapolation technique.

Concepts from thermodynamics were also presented in Chapter 3. It is important that the microscopic systems being reinstantiated during each continuum step obey certain continuum level constraints such as having the correct number of elements in a system, and correct values for the volume and energy of a system, while still being in a state

of equilibrium. Thermodynamics provides a means for constructing microscopic systems that are both in equilibrium and embody certain macroscopic characteristics. This is done by setting up an entropy maximization problem, or an equivalent energy minimization problem.

Chapter 4 contains an example of a one block cytoskeletal network. The example is utilized to demonstrate that the microscopic data such as filament orientation and strain does not conform to known PDF families of distributions and also that the distribution of data changes over time as the network is deformed. This provides evidence for the need for non-parametric PDF estimation, and also the need to follow the evolution of these PDFs over time. The chapter also contains several simple examples of finding equilibrium states for systems of filaments, given certain macroscopic constraints. At first an entropy formulation is used since an energy constraint is provided. However, writing an expression for the entropy becomes difficult for complex systems, thus the problem is recast as an energy minimization problem that includes an energy constraint.

Chapter 5 detailed the development of the microscopic algorithm. One block examples were utilized to test different ideas. Simple cases such as a block of parallel filaments and a block of non-crosslinked filaments were presented first. These first two cases did not utilize an energy minimization procedure since they were already in equilibrium states from their construction. In each block example, a network was created, the block was deformed via an extensional strain, data on filament orientation and strain was collected, PDFs of these data sets were constructed and utilized to generate a new network either at the same point in time or sometime in the future. The parallel filament case was useful for confirming that the general algorithm was working properly. In the non-crosslinked filament case, it was determined that single variable PDFs for the two angles and strains was not an acceptable data collection method due to the correlation between the angles and the strain. Two different techniques to address this issue were presented. The first technique utilized a deterministic relationship to assign a strain to each filament based on the orientation of that filament and the overall strain of the block. This method

was successful at creating networks of non-crosslinked filaments with similar elasticity moduli to the original network. Such a deterministic relationship would likely no longer hold true in a crosslinked network. This prompted the development of a strain binning method. In this method, the strains of filaments with similar orientations were saved into bins. A normal distribution for the strain was constructed within each bin based on this saved data, creating a piecewise joint PDF which was then utilized to assign strains to filaments during reinstantiation. This method was also successful in its endeavor to create networks with similar elastic properties to the original.

Finally an example with a fully crosslinked network of filaments was tested. When this network was deformed an energy minimization procedure was utilized to rearrange the internal filament segments and crosslinks. Network reinstantiation was done utilizing the strain binning method. This method failed to produce a network with similar elastic properties to the original. The new network began with a configuration and stored energy close to that of the original, however an application of the energy minimization procedure to settle the network was causing a large drop in stored energy and thus a large drop in the elasticity moduli of the block. To fix this problem, a new target function similar to ones developed in the thermodynamics examples, in conjunction with a bisection type method, was employed to create a network with the correct stored energy that was also in a state of mechanical equilibrium. This resolved the problem and the method was successful at producing networks with similar properties to the original at current and future points in time.

With the basic algorithm complete, Chapter 6 contained a demonstration of the full continuum-microscopic model for the simulation of a rectangular piece of cytoskeleton. Simulations where the microscopic reinstantiation procedure was used to update the elasticity moduli of one block, a column of blocks, and all the blocks were compared against simulations where the elasticity moduli of the same blocks were updated by full microscopic evolution of the original networks. Results showed greater variability in the moduli of the microscopic reinstantiation cases versus the full microscopic cases, but the

data took on the same general shape. Possible sources of error for these comparisons were discussed. The continuum level variables that dictate the overall state and shape of the cytoskeleton had much lower errors. The small errors of the microscopic reinstantiation procedure are a small price to pay for the large savings in computational time that this algorithm offers in comparison to a full microscopic simulation.

## 7.2. Discussion

The motivation for the development of this algorithm was to model complex biological materials such as the cytoskeleton undergoing deformation. At the cellular level the cytoskeleton is often perceived as a continuous medium, but its microscopic scale structure is a highly heterogeneous network of fibers. Creating computational models of this material at the whole cell scale that accurately model its microstructure is a challenge. The solution has often been to ignore the heterogeneous nature of the cytoskeleton in a purely continuum level computation or to create a detailed model of a small portion of cytoskeleton [**59**]. Continuum-microscopic models are one possible way of combining the two scales into one algorithm. The problem with current CM models is that the microscopic data utilized to update continuum level parameters is discarded after each continuum step. No memory of the microstructure is retained. In heterogeneous materials like the cytoskeleton this is a problem, because the filaments form certain patterns and configurations under different strains and losing this information will lead to inaccurate predictions for continuum level parameters such as elasticity moduli.

The new method developed in this research performs the basic CM algorithm, but with the added feature that the microscopic data is saved in the form of probability distribution functions. These PDFs are then extrapolated forward in time and utilized to instantiate a microstructure at the next continuum step that resembles the microstructure of the original network at the same future point in time. This method provides an accurate way of determining the local elastic parameters (both in space and in time) that are then passed on to the continuum level equations to close the system. A purely

continuum level equation set with constant elasticity parameters would not capture the variability of the elastic moduli that come about from the structure's heterogeneous nature.

A full microscopic simulation of the cytoskeleton is computationally intractable. The CM model presented here cuts the computational expense by 50%, if reinstantiations are done during each continuum time step, and by 75% when a sampling method is applied.

In conclusion, this new method offers a computationally efficient algorithm for modeling continuous media that incorporates the variability of its mechanical properties that stem form its heterogeneous structure. Though the method has been demonstrated for modeling the cytoskeleton, the general algorithm should be employable in a variety of applications.

**7.2.1. Publications from this Research.** This research effort has produced several publications. The first paper entitled "A numerical model of cellular blebbing: A volume-conserving, fluidstructure interaction model of the entire cell" was published in the *Journal of Biomechanics*, Volume 43, Issue 2, January 2010. This paper presented a computational model of cellular blebs, a fluid-filled protrusion of the cell membrane that forms when the cytoskeleton and membrane separate. The main focus of this model was the interaction between the cellular fluid (cytosol) and the membrane since this process is what drives bleb formation. The model included a coarse-grained representation of the actin-myosin cytoskeleton, which plays a role in bleb initiation and retraction. Constant mechanical parameters were used to describe the elastic response of the cytoskeleton. This model successfully captured the blebbing phenomenon.

The desire to improve this first model by replacing the coarse-grained cytoskeletal model with a model that would capture the medium's time and space-varying behavior was the main inspiration for this thesis work. The novel continuum-microscopic algorithm and results presented in this thesis has been summarized in an article entitled "A continuum-microscopic model of fibrous, heterogeneous media with dynamic microstructures" and submitted in May 2010 to *SIAM Multiscale Modeling and Simulation.* A

generic fibrous medium was utilized to demonstrate the model. The macroscopic and microscopic models for this material were described, as well as the microscopic reinstantiation and energy correction procedures. Results from a full microscopic simulation were compared to the results from the new algorithm. Emphasis was placed on the efficiency and accuracy of this new model.

The first publication on blebbing also spawned an invitation to contribute a chapter to the book *Cellular and Biomolecular Mechanics and Mechanobiology* edited by Amit Gefen, in the Springer series *Studies in Mechanobiology, Tissue Engineering and Biomaterials*. The chapter was entitled "Multiscale computation of cytoskeletal mechanics during blebbing" and was accepted for publication in May 2010. The goal of this work was to incorporate the large range of scales present in the blebbing process into one computational model. These scales range from the molecular level where biochemical reactions take place, to the cellular level where large changes in cytoskeleton and membrane shape occur. The computation includes three models: continuum, kinetic and molecular, that are advanced together and interact with one another in a time-parallel algorithm. This efficient model is able to capture changes at the macroscopic scale brought on by changes at the microscopic scale.

A third paper is planned for the fall of 2010 that will present the statistics-based, continuum-microscopic model from the thesis for a cytoskeletal application. As will be discussed in the "Future Work" section below, some modifications need to be made to the current algorithm in order to make it more applicable to biological materials. Namely, viscous damping from interstitial fluid will need to be added both to the macroscopic elasticity equations and to the microscopic model's energy minimization procedures.

## 7.3. Future Work

This research has laid the basic framework for a microscopic reinstantiation procedure that can be incorporated in a continuum-microscopic model. There are several computational improvements still to be made and other avenues to explore to increase the model's

efficient. For the cytoskeleton application, additions will be made to the network model to include more realistic biological features. Finally as stated before this new algorithm is not confined to work only for the cytoskeleton, but could be utilized for numerous applications involving heterogeneous media.

One goal of future work will be to improve the accuracy of the reinstantiation procedure. Several possible methods have already been suggested in the thesis but will be mentioned again here. The current model bases the reinstantiation on only the previous network. The algorithm will be adjusted to include more past information, which should improve the accuracy of the extrapolated data. The extrapolation procedure itself will also be examined. Currently the extrapolation is done utilizing least squares approximation functions created from the data. The basis functions utilized in the approximation are orthogonal Legendre polynomials, but other basis functions should be tested to see if another set produces better approximations. One can also experiment with the number of terms or degree of the approximating function. The functions in the examples presented in this work are parametrized by time, but one could also parametrize them by continuum strain state and/or strain rates. Parameterizing the functions with more variables will require more data points, but this may be possible if more past network data is included.

In terms of computational efficiency one area that will be explored further is that of sampling. As mentioned previously the current algorithm does not perform a microscopic reinstantiation in a cell whose strain state has not changed over the previous continuum step. If a particular activity in the cell (such as a protrusion forming) is confined to a small area, then only the grid cells in this localized area will have their continuum level parameters recomputed. The rest of the grid cells will use the parameter valuse computed during their last strain state change. Another way of implementing this would be to utilize the adaptive mesh refinement routines already built into the Bearclaw code as a flagging mechanism for microscopic reinstantiation. If an area of the domain needs to be resolved to a finer grid to compute its solution, this is likely an area where variables are

changing and is thus an area that should have its continuum level parameters updated. This method will be tested against the computational efficiency of the current method to see whether any speed-up can be gained.

The microscopic model utilized in this algorithm captures the crosslinked network structure of the cytoskeleton. However several additions could be made to make the model more biologically realistic. In the real cytoskeleton, crosslinks can break and reform based on forces and the presence or absence of various proteins. When the network is strained, crosslinks should be allowed to break if too much force is generated in the deformation. Likewise, after a deformation has taken place, new crosslinks should be allowed to reform if two filaments are close enough together. One could also add the presence of various crosslinking proteins by assigning concentration levels of each type of monomer over the domain. These levels could be utilized to compute probabilities of crosslink formation or severing in different areas of the cell. The presence of other types of cytoskeletal proteins such as actin and myosin could also be introduced to the model. Myosin is known to generate forces in the cytoskeleton by sliding actin filaments over one another [**2**]. The presence of myosin filaments in the cytoskeleton can change the elastic properties of the network, and thus would be another interesting addition to make to the model.

One important issue that has not been addressed by this cytoskeletal model is the presence of cytosol (the interstitial fluid). A fluid-structure interaction algorithm was completed with a very simple cytoskeleton in previous work [**83**]. This new model focused on creating an improved model of the cytoskeleton, but has not yet been coupled to a fluid solver. The presence of fluid surrounding the network will produce friction, which is a loss of energy. The system would no longer be conservative. The energy minimization procedure used to move the filaments during a microscopic deformation step will have to be altered to account for this damping. The reinstantiation procedure that minimizes a function of the form $F(X) = (U(X) - U_0)^2 + U(X)$ that was derived from thermodynamics

will also have to change to reflect the presence of the fluid. A full cytoskeleton-fluid-membrane continuum-microscopic model of the cell will be a longer term goal of this research.

The networks created in the examples presented in this thesis were non-affine, meaning that even though a uniform strain was applied to the network, strain was not uniformly distributed among the filaments in the network (due to energy minimization). During reinstantiation strains were assigned to the filaments based on strain data collected from the previous network in that grid cell. A new continuum strain state of the block is found by advancing the elasticity equations. Theoretically the average strain state of the microscopic network should match the strain state of its overlying continuum grid cell. How to enforce this is unclear. This issue is a very interesting open research question and one that will be pursued in future work as well.

Lastly, there are many interesting, possible applications of this new algorithm to pursue. The cytoskeleton alone is involved in a diverse array of cellular activies including mitosis, apoptosis, and cell spreading and migration. Continuum-microscopic models have yet to be applied to many of these phenomena and thus there is plenty of research opportunities available in these areas. Though the model in this thesis was developed for the cytoskeleton, the general framework of this novel continuum-microscopic model should be applicable to many multiscale problems that require microscopic data to update continuum level parameters. One particular problem of interest is to utilize these algorithms to model the human retina. Many diseases and abnormal conditions of the eye stem from the malformation or breakdown of the retina. The retina at the scale of the human eye looks like a continuous tissue that lines the back of the eye. However, microscopically it has a very complex structure composed of interconnected layers of different types of neurons. Very little computational modeling of the retina has been done thus far, and the development of a continuum-microscopic model will hopefully shed light on the retina's mechanical behavior in people with visual impairments caused by retinal problems.

This research effort has laid the groundwork for a new class of continuum-microscopic models that retain memory of the microscopic data over time. With continued development and improvement, hopefully these new algorithms will be useful for many different multiscale problems in a variety of reseach fields.

# APPENDIX A: Numerical Methods

## A.1. Energy Minimization

After a block has undergone an overall strain, the internal filament segments and crosslinks need to react to this strain. The expectation is that they will rearrange themselves to positions of minimal energy. The energy function looks like:

$$E = \sum_{j=1}^{n} \frac{k_j (L^j - L_0^j)^2}{2}$$

where this sum is over all filament segments and crosslinking springs. The $L^j$ is the current length of the spring and $L_0^j$ is the equilibrium length. The quantity $k_j$ is the spring constant for that spring, which is different for springs of different lengths. If $E$ is the elasticity modulus of the filament material, then the spring constant is computed by the following relationships:

$$\begin{aligned} F &= k\Delta l \quad \text{Hooke's Law for springs} \\ \frac{F}{A} &= E\frac{\Delta l}{l} \quad \text{Hooke's Law in general} \\ F &= \frac{AE}{l}\Delta l \\ k &= \frac{AE}{l} \end{aligned}$$

where $l$ is the spring length and $\Delta l$ is the displacement of the length from equilibrium. The cross-sectional area $A$ of the filament is assumed to be the same for all filaments and a unit value.

This energy system may look like a simple quadratic, but it is actually not due to the three-dimensionality of the problem. What needs to move in order to minimize the energy are the interior points of filaments and crosslinks, which each have 3 coordinates $(x, y, z)$ that define their location. The energy is dependent on the spring's current length

$L^i$, which can be written in terms of the coordinates as:

$$L^j = \sqrt{(x_1^j - x_0^j)^2 + (y_1^j - y_0^j)^2 + (z_1^j - z_0^j)^2}$$

where the 0 and 1 index the two endpoints of this filament segment. The energy is therefore:

$$E = \sum_{j=1}^{n} \frac{k_j(\sqrt{(x_1^j - x_0^j)^2 + (y_1^j - y_0^j)^2 + (z_1^j - z_0^j)^2} - L_0^j)^2}{2}$$

Taking partial derivatives of this expression with respect to each $x_0, y_0, z_0, x_1, y_1, z_1$ and setting these equal to zero will yield a nonlinear system of equations. To solve such a system an iterative procedure is used.

A gradient search algorithm was employed to find the optimal position of the filament and crosslink endpoints that minimize the total energy. The basic idea of a gradient search is to taken the current position of the system and move each point in the negative gradient direction. The gradient vector of a surface always points in the direction of steepest increase, so the negative of this vector will be in a decreasing direction towards a minimum. Close to the minimum, the gradient vector should be close to the zero vector, and this will be the indication to stop the iteration. The iteration looks like:

$$x_{new} = x_{old} - \alpha \nabla E$$

The parameter $\alpha$ is the distance traveled down the gradient before looking for a new direstion of travel. Typically, $\alpha$ is not known ahead of time, but ideally it should be chosen so that the move $\alpha \nabla E$ finds a minimum for $E$ in this direction. A method for estimating $\alpha$ can be established by assuming that in the vicinity of the current position $x_{old}$ on the energy landscape along the gradient direction $\nabla E$ line, the surface is a parabola. An equation for this parabola by finding three points along the gradient line, with the first and third point having larger energy values then the middle, second point. With these points, a Lagrange polynomial of degree two is constructed. The minimum of the parabola is found and this gives an estimate for $\alpha$.

## A.2. Grid Mappings

Due to the complex shape of a typical cell, the cytoskeleton cannot be gridded by a simple three-dimensional Cartesian grid. An unstructured grid will be used to discretize the domain where the grid cells will be hexahedra. During each instantiation step, new filament networks must be laid within each grid cell, and doing so in a complex 3D shape is computationally challenging. A more efficient and simple method is to to construct the filament networks within a unit cube cell and then map the network to its corresponding hexahedra in physical space (Figure 0.1).

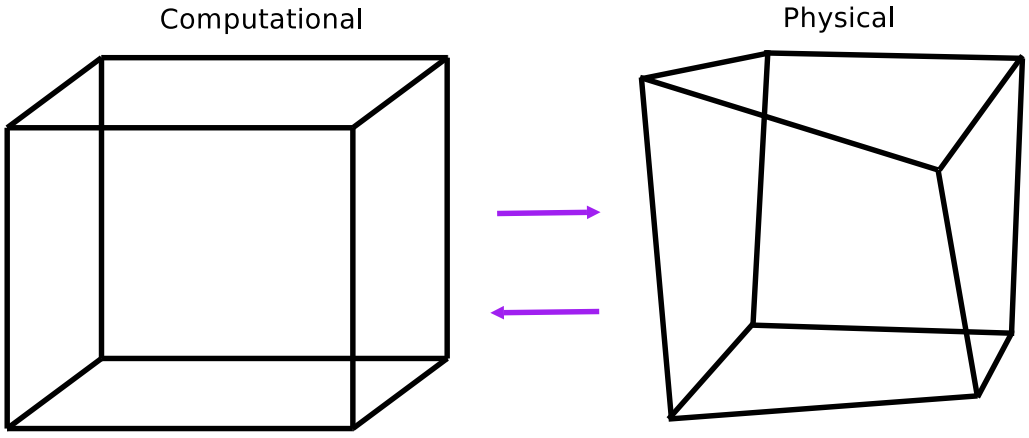


FIGURE 0.1. A unit cell in computational space and a corresponding grid block in physical space.

Let $\xi, \eta, \tau$ be the coordinate variables in computational space and $x, y, z$ the variables in physical space. The unit box where each network will be constructed has dimensions $[-1, 1] \times [-1, 1] \times [-1, 1]$ in computational space. This box needs to be mapped to a hexahedra in physical space defined by eight corner points: $(x_i, y_i, z_i)$ $i = 1..8$. Given a point $\xi, \eta, \tau$ in the unit box, it can be mapped to a point $(x, y, z)$ in physical space by:

$$x = f(\xi, \eta, \tau) = \sum_{i=1}^{8} x_i N_i(\xi, \eta, \tau) \tag{A.1}$$

$$y = g(\xi, \eta, \tau) = \sum_{i=1}^{8} y_i N_i(\xi, \eta, \tau) \tag{A.2}$$

$$z = h(\xi, \eta, \tau) = \sum_{i=1}^{8} z_i N_i(\xi, \eta, \tau) \tag{A.3}$$

where $N_i(\xi, \eta, \tau)$ are the following functions:

$$N_1(\xi,\eta,\tau) = \frac{1}{8}(1-\xi)(1-\eta)(1-\tau) \qquad N_2(\xi,\eta,\tau) = \frac{1}{8}(1+\xi)(1-\eta)(1-\tau)$$
$$N_3(\xi,\eta,\tau) = \frac{1}{8}(1+\xi)(1+\eta)(1-\tau) \qquad N_4(\xi,\eta,\tau) = \frac{1}{8}(1-\xi)(1+\eta)(1-\tau)$$
$$N_5(\xi,\eta,\tau) = \frac{1}{8}(1-\xi)(1-\eta)(1+\tau) \qquad N_6(\xi,\eta,\tau) = \frac{1}{8}(1+\xi)(1-\eta)(1+\tau)$$
$$N_7(\xi,\eta,\tau) = \frac{1}{8}(1+\xi)(1+\eta)(1+\tau) \qquad N_8(\xi,\eta,\tau) = \frac{1}{8}(1-\xi)(1+\eta)(1+\tau)$$

There are some instances in the code where it is necessary to do the reverse mapping from physical to computational space. Given $x, y, z$, the goal is to find $\xi, \eta, \tau$. This can be done by solving the equations in A.1-A.3 for $\xi, \eta, \tau$. This is a nonlinear system of 3 equations. It can be solved using Newton's method for three variables. One variable Newton's method finds the solution to the nonlinear equation $f(x) = 0$ by carrying out successive iterations of:

$$x_{new} = x_{old} - \frac{f(x_{old})}{f'(x_{old})}$$

For this case with 3 variables, the updating formula becomes:

$$\text{(A.4)} \qquad \begin{bmatrix} \xi_{new} \\ \eta_{new} \\ \tau_{new} \end{bmatrix} = \begin{bmatrix} \xi_{old} \\ \eta_{old} \\ \tau_{old} \end{bmatrix} - \begin{bmatrix} f_\xi & f_\eta & f_\tau \\ g_\xi & g_\eta & g_\tau \\ h_\xi & h_\eta & h_\tau \end{bmatrix}^{-1} \begin{bmatrix} f(\xi_{old}, \eta_{old}, \tau_{old}) \\ g(\xi_{old}, \eta_{old}, \tau_{old}) \\ h(\xi_{old}, \eta_{old}, \tau_{old}) \end{bmatrix}$$

Due to the simplicity of equations A.1-A.3 the partial derivatives in A.4 can be computed exactly, and the inverse of the matrix can also be found analytically. This iterative procedure is run until convergence is reached.

## A.3. Least Squares Approximation

Least squares approximation is a technique which seeks to create a function that minimizes the distance between itself and a set of data points (like a regression line). This method is used to construct functions through the time data of variables such as angle, strains and total energy in order to have a method of approximating future data.

In mathematical terms, data points $(x_i, y_i)$, $i = 1..m$ are given, and the goal is to find a function $f(x) = \sum_{j=1}^{n} a_j \phi_j(x)$ such that:

$$S = \frac{1}{m}\left[\sum_{i=1}^{m}(f(x_i) - y_i)^2\right]^{1/2}$$

is minimized. Here, $\phi_j(x)$ are a set of basis functions and $a_j$ are coefficients. In order to minimize this sum, partial derivatives with respect to each $x_i$ are taken and set equal to zero. This system of equations is equivalent to setting up and solving the following overdetermined system of equations: $Ax = b$ with $A$ an $m \times n$ matrix containing the basis functions $\phi_j(x)$ evaluated at the $x_i$'s, $b$ an $m \times 1$ vector containing the $y_i$'s, and $x$, an $n \times 1$ vector containing the unknown $a_j$ coefficients.

$$\begin{bmatrix} \phi_1 x_1 & \phi_2(x_1) & ... & \phi_n(x_1) \\ \phi_1 x_2 & \phi_2(x_2) & ... & \phi_n(x_2) \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ \phi_1 x_m & \phi_2(x_m) & ... & \phi_n(x_m) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ .... \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ ... \\ y_m \end{bmatrix}$$

The choice of basis functions is dependent on the type of data. In the simulations in this thesis, the Legendre polynomials: $[1, x, \frac{1}{2}(3x^2 - 1), \frac{1}{2}(5x^3 - 3x)...]$ were used for the basis set.

To solve the overdetermined system $Ax = b$, both sides of the equation are multiplied by $A^T$ (the transpose of $A$):

$$A^T A x = A^T b$$

This is the normal equation for the least squares problem. This system is equivalent to the system of partial derivatives of the $S(x)$ sum. This linear equation set can be solved via Gaussian elimination. Once the coefficients $a_j$ in vector $x$ have been found for the approximating function $f(x)$, it can be used to predict the behavior of the data at other points $x_{new}$.

## A.4. Stress and Strain Tensor Computation

This section describes how the three-dimensional stress and strain tensors will be computed.

Ideally, when the energy minimization is run, the system should settle to a position where the forces over all the faces of the block sum to zero. If this is the case, then the stress tensor can be filled in by finding the forces on the faces and dividing by the area of the faces. However, using a gradient search procedure to find a solution to the energy minimization problem can create some numerical error. It is an iterative algorithm that stops when the error in solution is below a certain threshold. The lower the threshold the more accurate the solution. However a lower threshold will require more iterations to reach a solution and thus more computational time. One must balance the two goals of accuracy and low computational expense.

Before explaining how the stress tensor is computed in such a situation, evidence of convergence of the gradient procedure will first be presented. As stated above, the more iterations that are done, the more accurate the results should be. To test this, the energy minimization procedure was run for an increasing number of iterations. At each level, the total force in the $x$ direction on the block was computed. This value represents the error made by the energy minimization procedure. The graph of the number of iterations versus the error in total force is shown in Figure A.4. Convergence is occurring at an approximately linear rate.

These numerical errors (though small) produce a non-physical situation. An imbalance of forces suggests that the block is in motion, which is not the case. Before computing the stress tensor the forces need to be adjusted so that they represent a physical scenario. There are six faces on a block, and each face has a computed force vector. The forces on a face are determined by adding up the spring forces of any filaments attached to that face. The force vector for face $i$ is given as $(f_x^i, f_y^i, f_z^i)$, thus there are 18 total forces computed for the block. Figure 0.3 shows a diagram of a block that will be

| Number of Iterations | Relative Error in Total Force |
|---|---|
| 1000 | 0.0395 |
| 2000 | 0.0354 |
| 5000 | 0.0226 |
| 10000 | 0.0113 |
| 20000 | 0.00383 |
| 50000 | 0.000193 |

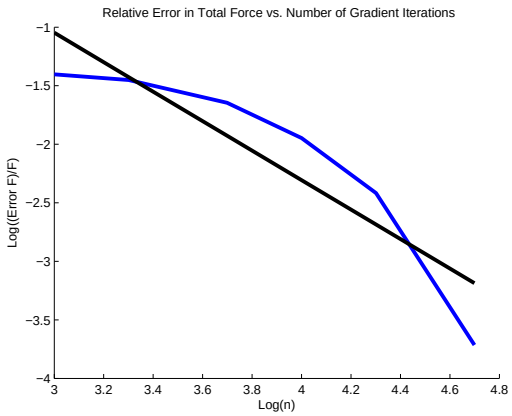


FIGURE 0.2. A log-log plot of the number of gradient procedure iterations versus the error in total force (in the $x$ direction) on the three-dimensional block. The blue data is the actual values, and the black line is the regression line through the data, with slope -1.25.

used as a reference to set up the equations that need to be satisfied to have a physically correct system.

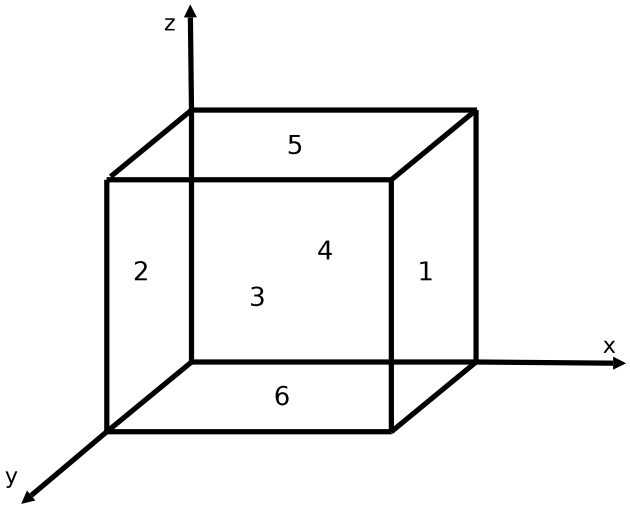


FIGURE 0.3. Diagram of a network block to use for reference in the force calculations. Face 3 is the front face, and face 4 is the back face of the box.

To satisfy a zero translation criteria, the following equations must be satisfied:

$$f_x^1 + f_x^2 + f_x^3 + f_x^4 + f_x^5 + f_x^6 = 0 \tag{A.5}$$

$$f_y^1 + f_y^2 + f_y^3 + f_y^4 + f_y^5 + f_y^6 = 0 \tag{A.6}$$

$$f_z^1 + f_z^2 + f_z^3 + f_z^4 + f_z^5 + f_z^6 = 0 \tag{A.7}$$

Besides these equations, zero rotation conditions must also be satisfied. This gives three more equations that need to be satisfied:

$$f_y^2 + f_x^3 - f_y^1 - f_x^4 = 0 \tag{A.8}$$

$$f_z^1 + f_x^6 - f_z^2 - f_x^5 = 0 \tag{A.9}$$

$$f_z^4 + f_y^5 - f_z^3 - f_y^6 = 0 \tag{A.10}$$

The 18 force values must satisfy the 6 conditions in A.5 - A.10. With the numerical error from the gradient procedure, these conditions are likely not exactly satisfied. To adjust the 18 force values to make them satisfy the conditions a least squares projection will be done. The 18 forces can be represented in an 18-dimensional phase space with one point. The 6 physical conditions can be represented in this phase space as a hypersurface. The 6 equations intersect in this 18-dimensional space with a 12-dimensional hypersurface. To goal is to project the 18-dimensional force point onto this surface and do so with an orthogonal projection to minimize the changes made to the force values. The procedure goes as follows:

(1) Find a basis for the 12-dimensional space.
(2) Form the $18 \times 12$ matrix $A$ with the basis vectors as the columns.
(3) Put the current force values into an $18 \times 1$ vector $b$.
(4) Set up the overdetermined linear system: $Ax = b$, where $x$ represents the projection of $b$ onto the $12 \times 1$ vector $x$, which can be found on the 12-dimensional surface.
(5) This is now a least squares type problem, which can be solved by forming the normal equations: $A^T Ax = b$ and finding $x$ via Gaussian elimination.

For this situation, to find the basis, it will simplify notation to use the alphabet letters $[a, b, c, d, e, f, g, h, i, j, k, l, m, n, p, q, r, s]$ for $[f_x^1, f_x^2, f_x^3, f_x^4, f_x^5, f_x^6, f_y^1, f_y^2, f_y^3, f_y^4, f_y^5, f_y^6, f_z^1, f_z^2, f_z^3, f_z^4, f_z^5$
If Equations A.5 - A.10 are converted to the alphabet letters, and 6 variables $(a, c, i, l, m, r)$

are eliminated, a vector belonging to the desired hypersurface will look like:

$$\begin{bmatrix} -b-g-2d+h-e-f \\ b \\ g+d-h \\ d \\ e \\ f \\ g \\ h \\ -g-h-j-2k+p-q \\ j \\ k \\ q+k-p \\ n+e-f \\ n \\ p \\ q \\ -2n-e+f-p-q-s \\ s \end{bmatrix} \tag{A.11}$$

This vector can be written as a sum of 12 orthogonal vectors, one for each of the 12 remaining letters. The $x$ solution vector will give the values of $(b, d, e, f, g, h, j, k, n, p, q, s)$ and then Equation A.11 can be used to find the remaining 6 letters.

The strain tensor for each block must also be computed. This can be done by using the definition of strain in terms of displacement. For finite strains:

$$\underline{\underline{\mathbf{E}}} = \frac{1}{2}((\nabla \mathbf{u})^T + \nabla \mathbf{u} + (\nabla \mathbf{u})^T \cdot \nabla \mathbf{u})$$

where $\mathbf{u}$ is the displacement vector. The displacement of each block is available from the continuum level equations by using the velocities and the time step. Finite differencing can be used to find the spatial derivatives of the displacement to fill in the components of the strain tensor.

# BIBLIOGRAPHY


1. A. Abdulle, Analysis of heterogeneous multiscale fem for problems in elasticity, Mathematical Models and Methods in Applied Sciences **16** (2006), no. 4, 615–635.

2. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, Molecular biology of the cell, fourth ed., Garland, 1994.

3. G. Albrecht-Buehler, Does blebbing reveal the convulsive flow of liquid and solutes through the cytoplasmic meshwork?, Cold Spring Harbor Symposia on Quantitative Biology **46** (1982), no. 1, 45–49.

4. W. Alt and M. Dembo, Cytoplasm dynamics and cell motion: two-phase fluid models, Mathematical Biosciences **156** (1999), no. 1-2, 207–228.

5. E. Arruda and M. Boyce, A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials, Journal of the Mechanics and Physics of Solids **41** (1993), no. 2, 389–412.

6. J.A. Aström, P.B. Sunil Kumar, I. Vattulaine, and M. Karttunen, Strain hardening, avalanches, and strain softening in dense cross-linked actin networks, Physical Review E **77** (2008), no. 5, 051913.

7. K. E. Atkinson, An introduction to numerical analysis, second ed., John Wiley and Sons, 1989.

8. S. Baeurlea, T. Usamid, and A. Guseve, A new multiscale modeling approach for the prediction of mechanical properties of polymer-based nanomaterials, Polymer **47** (2006), no. 26, 8604–8617.

9. J. Bell, J. Foo, and A. Garcia, Algorithm refinement for the stochastic burgers' equation, Journal of Computational Physics **223** (2007), no. 1, 451–468.

10. D. Boal, Mechanics of the cell, Cambridge University Press, 2002.

11. S. Boey, D. Boal, and D. Discher, Simulations of the erythrocyte cytoskeleton at large deformation i. microscopic models, Biophysical Journal **75** (1998), no. 3, 1573–1583.

12. K. Bold, Y. Zou, I. Kevrekidis, and M. Henson, An equation-free approach to analyzing heterogeneous cell population dynamics, Journal of Mathematical Biology **5** (2007), no. 3, 331–352.

13. G. A. Buxton, N. Clarke, and P. J. Hussey, Actin dynamics and the elasticity of cytoskeletal networks, Express Polymer Letters **3** (2009), no. 9, 579–587.

14. H.B. Callen, Thermodynamics and an introduction to thermostatistics, second ed., John Wiley and Sons, 1985.

15. M. L. Cano, D. A. Lauffenburger, and S. H. Zigmond, Kinetic analysis of f-actin depolymerization in polymorphonuclear leukocyte lysates indicates that chemoattractant stimulation increases actin filament number without altering the filament length distribution, The Journal of Cell Biology **115** (1991), 677–687.

16. G. T. Charras, A short history of blebbing, Journal of Microscopy **231** (2008), no. 3, 466–478.

17. G. T. Charras, M. Coughlin, T. J. Mitchison, and L. Mahadevan, Life and times of a cellular bleb, Biophysical Journal **94** (2008), no. 5, 1836–1853.

18. G. T. Charras, C. Hu, M. Coughlin, and T. J. Mitchison, Reassembly of contractile actin cortex in cell blebs, The Journal of Cell Biology **175** (2006), no. 3, 477–490.

19. G. T. Charras, J. C. Yarrow, M. A. Horton, L. Mahadevan, and T. J. Mitchison, Non-equilibration of hydrostatic pressure in blebbing cells, Nature **435** (2005), 365–369.

20. O. Chaudhuri, S.H. Parekhand, and D.A. Fletcher, Reversible stress softening of actin networks, Nature **445** (2007), no. 18, 295–298.

21. J. Cisternas, C. Gear, S. Levin, and I. Kevrekidis, Equation-free modeling of evolving disease: Coarse-grained computations with individual-based models, Proc. Roy. Soc. London UK in press (2004).

22. C. C. Cunningham, Actin polymerization and intracellular solvent flow in cell surface blebbing, The Journal of Cell Biology **129** (1995), no. 6, 1589–1599.

23. D. Discher, D. Boal, and S. Boey, Simulations of the erythrocyte cytoskeleton at large deformation ii. micropipette aspiration, Biophysical Journal **75** (1998), no. 3, 1584–1597.

24. W. E and B. Engquist, The heterogeneous multiscale methods, Comm. Math. Sci. **1** (2003), no. 1, 87–132.

25. Weinan E and B. Engquist, Multiscale modeling and computation, Notices of the AMS **50** (2003), no. 9, 1062–1070.

26. R. Erban, I. Kevrekidis, D. Adalsteinsson, and T. Elston, Gene regulatory networks: a coarse-grained, equation-free approach to multiscale computation, Journal of Chemical Physics **124** (2006), no. 8, 084106.

27. N. Foger, R. Marhaba, and M. Zoller, Involvement of cd44 in cytoskeleton rearrangement and raft reorganization in t cells, Journal of Cell Science **114** (2001), no. 6, 1169–1178.

28. A. Garcia, J. Bell, W. Brutchfield, and B. Alder, Adaptive mesh and algorithm refinement using direct simulation monte carlo, Journal of Computational Physics **154** (1999), no. 1, 134–155.

29. M. L. Gardel, F. Nakamura, J. H. Hartwig, J. C. Crocker, T. P. Stossel, and D. A. Weitz, Prestressed f-actin networks cross-linked by hinged filamins replicate mechanical properties of cells, PNAS **103** (2006), no. 6, 1762–1767.

30. M. L. Gardel, J. H. Shin, F. C. MacKintosh, L. Mahadevan, P. Matsudaira, and D. A. Weitz, Elastic behavior of cross-linked and bundled actin networks, Science **304** (2004), 1301–1305.

31. W. Goddard, T. Cagin, M. Blanco, N. Vaidehi, S. Dasgupta, W. Floriano, M. Belmares, J. Kua, G. Zamanakos, S. Kashihara, M. Iotov, and G. Gao, Strategies for multiscale modeling and simulation of organic materials: polymers and biopolymers, Computational and Theoretical Polymer Science **11** (2001), no. 5, 329–343.

32. G. Grimmett and D. Welsh, Probability an introduction, Clarendon Press, 1986.

33. J.C. Hansen, R. Skalak, S. Chien, and A. Hoger, An elastic network model based on the structure of the red blood cell membrane, Biophysical Journal **70** (1996), 146–166.

34. D. Head, A. Levine, and F. MacKintosh, Deformation of cross-linked semiflexible polymer networks, Physical Review Letters **91** (2003), no. 10.

35. D. A. Head, A. Levine, and F. MacKintosh, Distinct regimes of elastic respone and deformation modes of cross-linked cytoskeletal and semiflexible polymer networks, Physical Review E **68** (2003), 061907.

36. P. Howell, G. Kozyreff, and J. Ockendon, Applied solid mechanics, Cambridge University Press, 2009.

37. E. M. Huisman, T. van Dillen, P. R. Onck, and E. Van der Giessen, Three-dimensional cross-linked f-actin networks: Relation between network architecture and mechanical behavior, Physical Review Letters **99** (2007).

38. G. Hummer and I. Kevrekidis, Coarse moledular dynamics of a peptide fragment: Free energy, kinetics and long time dynamics computations, J. Chem. Phys. **118** (2003), no. 23.

39. D.E. Ingber, Cellular tensegrity: defining new rules of biological design that govern the cytoskeleton, Journal of Cell Science **104** (1993), 613–627.

40. H. Isambert and A. C. Maggs, Dynamics and rheology of actin solutions, Macromolecules **29** (1996), 1036–1040.

41. P. A. Janmey, J. X. Tang, and C. F. Schmidt, Actin filaments (unpublished).

42. H. Keller and P. Eggli, Protrusive activity, cytoplasmic compartmentalization, and restriction rings in locomoting blebbing walker carcinosarcoma cells are related to detachment of cortical actin from the plasma membrane, Cell Motility and the Cytoskeleton **41** (1998), no. 2, 181–193.

43. I. Kevrekidis, C. Gear, and G. Hummer, Equation-free: The computer-aided analysis of complex multiscale systems, AIChE Journal **50** (2004), no. 7, 1346–1355.

44. I. Kevrekidis, C. Gear, J. Hyman, P. Kevrekidis, O. Runborg, and C. Theodoropoulos, Equation-free multiscale computation: enabling microscopic simulators to perform system-level tasks, Comm. Math. Sci **1** (2003), no. 4, 715–762.

45. M. Kröger, I. Stankovic, and S. Hess, Towards multiscale modeling of metals via embedded particle computer simulation, Multiscale Model Simulations **1** (2003), no. 1, 25–39.

46. R. Y. Kwon, A. J. Lew, and C. R. Jacobs, A microsturcturally informed model for the mechanical response of three-dimensional actin networks, Computer Methods in Biomechanics and Biomedical Engineering **11** (2008), no. 4, 407–418.

47. W. M. Lai, D. Rubin, and E. Krempl, Introduction to continuum mechanics, Pergamon Press Inc., 1974.

48. L. Landau and E. Lifshitz, Theory of elastcity, third ed., Pergamon Press, 1986.

49. J. Lee, B. Dünweg, and J. Schumacher, Multiscale modelling strategy using the lattice boltzmann method for polymer dynamics in a turbulent flow, Submitted to Computers and Matehmatics with Applications (2008), special issue conference ICMMES 2008.

50. R. Leveque, Finite volume methods for hyperbolic problems, Cambridge University Press, 2002.

51. J. Li, M. Dao, C. T. Lim, and S. Suresh, Spectrin-level modeling of the cytoskeleton and optical tweezers stretching of the erythrocyte, Biophysical Journal **88** (2005), 3707–3719.

52. R. Liboff, Kinetic theory classical, quantum, and relativistic descriptions, Prentice Hall, 1990.

53. R. Liption, Homogenization and field concentrations in heterogeneous media, SIAM Journal of Mathematical Analysis **38** (2006), no. 4, 1048–1059.

54. A.E.H. Love, A treatise on the mathematical theory of elasticity, fourth ed., Cambridge University Press, 1934.

55. A. Makeev and I. Kevrekidis, Equation-free multiscale computations for a lattice-gas model: coarse-grained bifurcation analysis of the no+co reaction on pt(1 0 0), Chemical Engineering Science **59** (2004), no. 8-9, 1733–1743.

56. J.C. Michel, H. Moulinec, and P. Suquet, Effective properties of composite materials with periodic mcirostructure: a computational approach, Compute Methods in Applied Mechanics an Engineering **172** (1999), no. 1-4, 109–143.

57. T. J. Mitchison and L. P. Cramer, Actin-based cell motolity and cell locomotion, Cell **84** (1996), 371–379.

58. S. Mitran, Bearclaw, 2001, http://coanda.amath.unc.edu/bearclaw.

59. M. Mofrad and R. Kamm, Cytoskeletal mechanics: Models and measurements, Cambridge University Press, 2006.

60. P. Morse and K. Ingard, Theoretical accoustics, Princeton University Press, 1968.

61. J. A. Nichol and O. F. Hutter, Tensile strength and dilatational elasticity of giant sarcolemmal vesicles shed from rabbit muscle, The Journal of Physiology **493** (1996), no. 1, 187–198.

62. G. Oster and A. Perelson, Cell spreading and motility: A model lamellipod, Journal of Mathematical Biology **21** (1985), 383–388.

63. J. Palmer and M. Boyce, Constitutive modeling of the stress strain behavior of f-actin filament networks, Acta Biomaterialia **4** (2008), no. 3, 597–612.

64. R.K. Pathria, Statistical mechanics, second ed., Elsevier Ltd., 1996.

65. I. Pivkin and G. Karniadakis, Accurate coarse-grained modeling of red blood cells, Physical Review Letters **101** (2008), no. 1, 118105.

66. P. A. Pullarkat, Loss of cell-substrate adhesion leads to periodic shape oscillations in fibroblasts, eprint arXiv:physics/0612156 (2006).

67. R. P. Rand and A. C. Burton, Mechanical properties of the red cell membrane. i. membrane stiffness and intracellular pressure, Biophysical Journal **4** (1964), no. 2, 115–135.

68. W. Ren and Weinan E, Heterogeneous multiscale method for the modeling of complex fluids and micro-fluidics, Journal of Computational Physics **204** (2005), no. 1, 1–26.

69. D. Scott, On optimal and data-based histograms, Biometrika **66** (1979), no. 3, 605–610.

70. U. Seifert, Modeling nonlinear red cell elasticity, Biophysical Journal **75** (1998), no. 3, 1141–1142.

71. M. P. Sheetz, J. E. Sable, and H. Döbereiner, Continuous membrane-cytoskeleton adhesion requires continuous accommodation to lipid and cytoskeleton dynamics, Annual Review of Biophysics and Biomolecular Structure **35** (2006), 417–434.

72. B. W. Silverman, Density estimation for statistics and data analysis, Chapman and Hall, 1986.

73. D. Stamenovic and M. Coughlin, A quantitative model of cellular elasticity based on tensegrity, Journal of Biomechanical Engineering **122** (2000), no. 1, 39–43.

74. H. Sturges, The choice of a class interval, Journal of the American Statistical Association **21** (1926), 65–66.

75. Y. Takagi, E. Homsher, Y. Goldman, and H. Shuman, Force generation in single conventional actomyosin complexes under high dynamic load, Biophysical Journal **90** (2006), 1295–1307.

76. P. Tracqui, E. Promayou, T. Sauvaget, V. Norris, and J. Martiel, Tensegrity and oscillations: exploring some constitutive and emergent features of virtual cell models (unpublished).

77. N.G. van Kampen, Stochastic processes in physics and chemistry, North Holland Publishing Co., 1981.

78. J. Wilhelm and E. Frey, Elasticity of stiff polymer networks, Physical Review Letters **91** (2003), no. 10.

79. Z. Xia, W. Curtin, and P. Peters, Multiscale modeling of failure in metal matrix composites, Acta Materialia **49** (2001), no. 2, 273–287.

80. J. Xu, Y. Tseng, and D. Wirtz, Strain hardening of actin filament networks, Journal of Biological Chemistry **275** (2000), no. 46, 35886–35892.

81. K. Xu and K. Prendergast, Numerical navier-stokes solutions from gas kkinetic theory, Journal of Computational Physics **114** (1994), no. 1, 9–17.

82. Y. Yoon, J. Kotar, G. Yoon, and P. Cicuta, The nonlinear mechanical response of the red blood cell, Physical Biology **5** (2008), no. 3, 36007.

83. J. Young and S. Mitran, A numerical model of cellular blebbing: A volume-conserving, fluidstructure interaction model of the entire cell, Journal of Biomechanics **43** (2010), no. 2, 210–220.