

THESIS

MODELING THE LARGE DEFORMATION OF PROTEINS:  
FROM COARSE-GRAINING TO CONTINUUM

Submitted by  
George Borleske  
Department of Mathematics

In partial fulfillment of the requirements  
For the Degree of Master of Science  
Colorado State University  
Fort Collins, Colorado  
Fall 2016

Master's Committee:

Advisor: Yongcheng Zhou

Patrick Shipman  
Ashok Prasad

Copyright by George Borleske 2016

All Rights Reserved

## ABSTRACT

There has been much research in the area of mathematically modeling protein complexes using coarse-grained models or continuum models. We are proposing a new approach based on data obtained from steered molecular dynamic (SMD) simulations performed with NAMD along with coarse-grained MARTINI force field. We developed a mathematical approach to evaluate the forces acting on each amino acid, which allows us to compute the stress tensor and the Green-Lagrangian strain tensor at each amino acid using the trajectories of SMD simulations. Continuum linear elastic moduli can then be derived by using these strain and stress tensors. Our results of Young's moduli at alpha helices are very consistent with general experimental measurements, and show strong inhomogeneity as seen in their sequence dependence. The Young's moduli at coils are particularly small, which support the hinge motion as observed in many large scale deformations of protein complexes. These continuum elastic properties will be used for finite element modeling of the large scale protein deformation in the future.

# TABLE OF CONTENTS

1	Introduction and Overview.....	1
1.1	Biology Background . . . . .	1
1.2	Motivation and Goal . . . . .	2
1.3	Computational Limits . . . . .	2
1.4	MARTINI Force Field . . . . .	3
1.5	Overview . . . . .	4
2	Strain.....	7
2.1	Derivation of the Green-Lagrangian Strain Tensor . . . . .	7
2.2	First Computation of the Green-Lagrangian Strain Tensor . . . . .	8
2.3	Second Computation of the Green-Lagrangian Strain Tensor . . . . .	12
3	Stress.....	13
3.1	History of the Derivation of the Stress Tensor . . . . .	13
3.2	Derivation of the Cauchy Stress Tensor in Eulerian Coordinates . . . . .	13
3.3	Derivation of the First Piola-Kirchoff Stress Tensor in Lagrangian Coordinates	18
3.4	Computation of Stress . . . . .	21
3.4.1	Bond Potential Energy . . . . .	22
3.4.2	Bonding Angle Potential Energy . . . . .	23
3.4.3	Dihedral Angle Potential Energy . . . . .	26
3.4.4	Lennard-Jones Potential Energy . . . . .	28
3.4.5	Coulombic Potential Energy . . . . .	29
3.5	Simulation . . . . .	29

4	Continuum Elastic Properties of the Material.....	31
4.1	Transversely Isotropic material . . . . .	31
4.2	Isotropic Simplification . . . . .	34
5	Computation Simulation Using Toy Model Proteins .....	35
5.1	Verlet Integration of Molecular Dynamic Simulation . . . . .	35
5.2	One Dimensional Toy Model Protein . . . . .	37
5.3	Elastic Modulus In Two Dimensions . . . . .	40
5.4	Two-Dimensional Toy Model . . . . .	41
6	Computation Simulation with Proteins .....	44
6.1	Toy Protein with All Backbone Nodes . . . . .	44
6.2	Alpha Helices . . . . .	47
6.3	Protein 1B6Q . . . . .	50
6.4	Coils . . . . .	51
7	Concluding Remarks .....	54
7.1	Conclusion . . . . .	54
7.2	Future Work . . . . .	55

## LIST OF FIGURES

1.1	Example of a large deformation of a protein. . . . .	3
2.1	Strain components of protein 1MV4 using first computation method . . . . .	10
2.2	Strain components of protein 1MV4 using second computation method. . . . .	11
3.1	Motions of a three node system that do not affect the angle formed. . . . .	25
5.1	Picture of the two springs used in toy simulation . . . . .	35
5.2	One dimensional spring before and after a simulation. . . . .	38
5.3	Computed Young's Modulus of the one dimensional system. . . . .	39
5.4	Computed Young's Modulus of the two dimensional system. . . . .	42
5.5	Poisson Ratio of the two dimensional system. . . . .	42
5.6	Bond length of the two bonds of the node tested within the spring. . . . .	43
6.1	Computed Young's Moduli of the toy alpha helix comprising of only Glycine backbone nodes. . . . .	45
6.2	Computed Poisson ratio of the fictitious alpha helix comprising of only Glycine backbone nodes. . . . .	46
6.3	$y$ -component of the bond length of several interior nodes of the fictitious toy alpha helix comprising of only Glycine backbone nodes. . . . .	47
6.4	Young's moduli of amino acids within a tolerance of 0 to 10. Unit is GPa. . .	48
6.5	Young's moduli of amino acids within a tolerance of 0 to 10. Unit is GPa. . .	48
6.6	Young's moduli of amino acids within a tolerance of 0 to 10. Unit is GPa. . .	49
6.7	Young's moduli of amino acids within a tolerance of 0 to 10. Unit is GPa. . .	49

6.8	Protein 1B6Q displayed in atomic and coarse-grained resolution. . . . .	50
7.1	Probe being rolled over a van der Waals surface [20] . . . . .	56
7.2	Finite element scheme representation of an energy-minimized molecular structure of T4 lysozyme. . . . .	57

## LIST OF TABLES

6.1	Young's Moduli of coil and nearby alpha helix nodes of 1B6Q. . . . .	51
6.2	The Computed Young's Moduli of All Coils Examined. . . . .	52
6.3	The Computed Young's Moduli of Coils that are within 0 and 1.5 GPa . . .	53



# CHAPTER 1

## INTRODUCTION AND OVERVIEW

### 1.1 BIOLOGY BACKGROUND

The term “protein” originates from the Greek work *proteios*, meaning “primary” or “of first rank” [21]. The name was adapted by Jöns Berzelius in 1838 to emphasize the importance of this class of molecules. Proteins play crucial, life-sustaining biological roles such as being triggers of physiological processes for all living things, and providing fundamental services of transport and storage for molecules such as oxygen and iron in muscle and blood cells, crucial regulatory roles such as digestion, immunological and hormonal functions, the coordination of neuronal activity, cell and bone growth, and cell differentiation.

Proteins are composed of linked amino acids. The amino acid composition is known as the *primary structure*. Each amino acid consists of a central tetrahedral carbon known as the alpha carbon ( $C^\alpha$ ), which is attached to four units: a hydrogen atom, a protonated amino group ( $NH_3^+$ ), a dissociated carboxyl group ( $COO^-$ ) and a distinguishing side-chain or R group. Including the primary structure there are four basic levels used to describe protein structure. The *secondary structure* consists of the regular local structural patterns such as  $\alpha$ -helices and  $\beta$ -sheets, or combination motifs thereof called *supersecondary structure*. *Tertiary structure* is the 3D arrangement of all atoms in the polypeptide chain in space. Finally, the *quaternary structure* which is used for large proteins with independent subunits is the complete three dimensional interaction network among the different subunits.

There are twenty naturally-occurring amino acids, about a dozen of which humans can synthesize within the body. The remaining nine must be obtained through ingestion.

## 1.2 MOTIVATION AND GOAL

Proteins play a major role in the biological functions of much of the life on this planet, including humans. Therefore protein's functions are of great interest to biologists. Equilibrium conformational fluctuations of proteins about their folded, native structure play an important role in their biological function [2]. A thorough understanding on the theoretical level, and accurate models are of utmost importance for future applications in the engineering realm of such complex materials [3]. Therefore, mathematical models are required in order to gain insight into these deformations.

The goal of the research in this thesis is to study the deformation of proteins using continuum mechanics. This can be done through the use of finite element frameworks to compute the normal modes, the strain and stress tensors, and the history of deformation. There has been much research into this field such as [2]. The finite element method (FEM) is convenient here because of its flexibility and ease of implementation compared to other methods such as finite differences or a spectral method, for solving the partial differential equation created by the system. We will be extending the work done by Bathe [2] by considering the coarse-grained protein as inhomogeneous rather than homogeneous. Therefore the elastic moduli must be computed by finding the stress and strain tensors at each backbone node, which will be considered as an amino acid at our resolution. The computation implementation of the algorithm is done in the language C.

## 1.3 COMPUTATIONAL LIMITS

Few molecular dynamic (MD) simulations have been performed on proteins under loaded conditions at an atomic resolution [15]. These types of simulations are called steered molecular dynamics (SMD) due to the "steering" of load application. A large difficulty in attempting these SMD simulations is the computational cost of running such a large system. If water is included and consists of roughly 95% of the atoms in the system the number of degrees

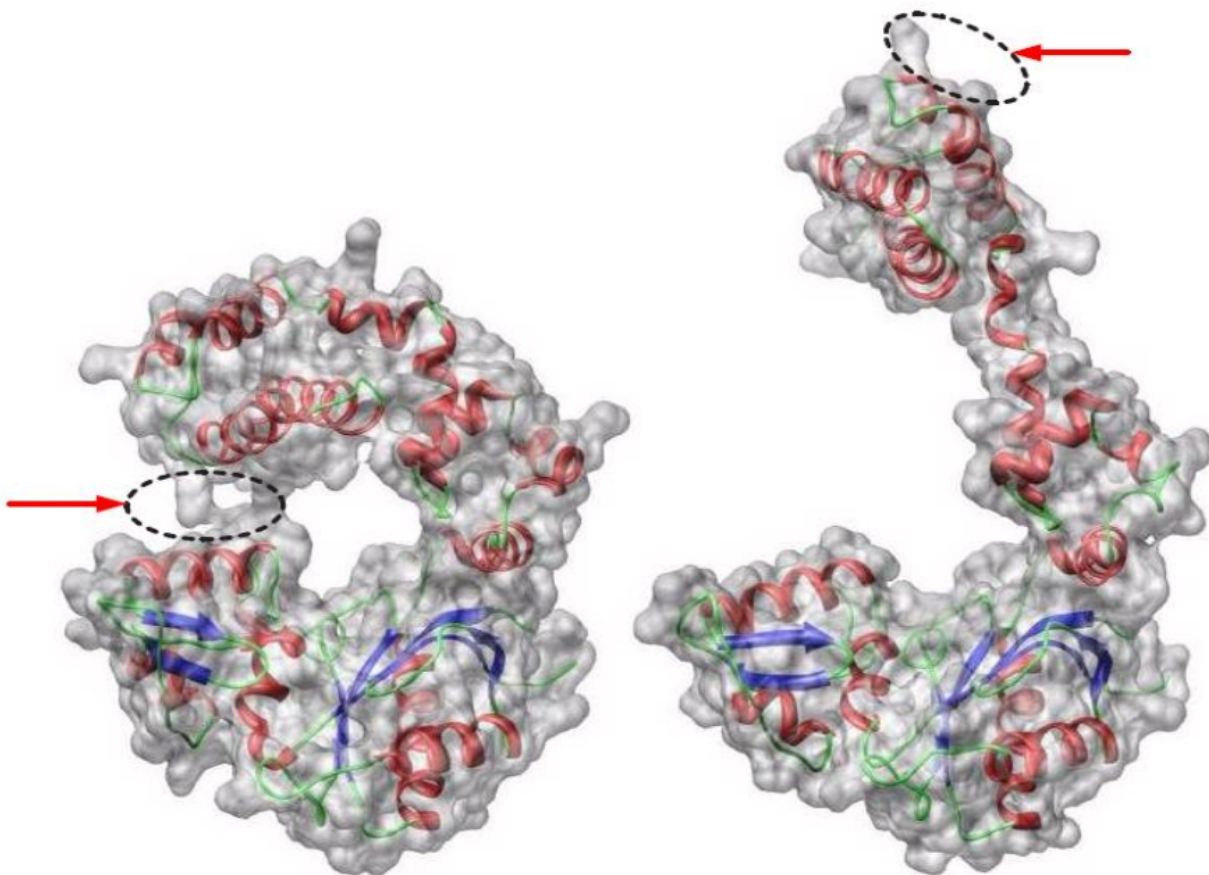


Figure 1.1: Example of a protein before and after a large deformation consisting of a rotation of the top portion of the protein [5].

of freedom from the elastic network model increases by one hundred fold. It was reported in 2010 that running an SMD simulation on a 9 monomer structure of F-actin with ninety-six cores requires over four hours for a 500 ps simulation while applying the same load to the structure in *CalculiX* FEM package requires around two seconds on a single quad core computer [15].

#### 1.4 MARTINI FORCE FIELD

A coarse-grain model is implemented for two reasons. The first is the computational costs of an performing an atomic simulation as described in the previous section. This will allow for a much larger and diverse amount of proteins to be studied. The second reason

is we are interested in the large scale deformations. The coarse-grained force field ignores the small scale deformations that are present within an atomic force field while preserving information regarding these large scale deformations.

Due to this computational intensity, the MARTINI Force Field [16] will be implemented. Each amino acid or residue will be mapped to a backbone node representing the residue along with zero to four side-chain nodes, coarse-graining the protein. The energy of the force field is separated into bonded and non-bonded interactions. Bonded interactions included bonds between two nodes, bonds between three nodes, and bonds between four nodes. They are called the bond potential energy, angle potential energy, and dihedral (torsion) angle potential energy. All three bonded energies depend on the secondary structure and the amino acid. The non-bonded energy consists of the Lennard-Jones (Van der Waals) potential energy and the Coulombic potential energy. Both of the non-bonded potentials implement a cut-off distance to reduce the computational complexity. The non-bonded interactions depend on distance between nodes. The Coulombic potential energy also depends on the charge.

## 1.5 OVERVIEW

Chapter 2 concerns the strain of an individual node. The strain will be represented by the Green-Lagrangian strain tensor. A derivation of this tensor is shown based on differential line segments. Followed by this are two applications of how the strain can potentially be computed in our model. The first is a three dimensional version of the one dimensional quantity, Green strain. Unfortunately this approach produced poor results when applied to our system, so a second application was attempted. This involved computing the strain from the deformation tensor which is used within the derivation of Green-Lagrangian strain tensor. This second method produced good results, and was used in the computation of the elastic modulus.

Chapter 3 concerns the stress tensor of an individual node. This model uses the virial

stress tensor, which is the atomistic version of mechanical stress tensor. The history of some notable papers of the virial stress tensor are first discussed. This is followed by a derivation of the virial version of the Cauchy stress tensor in Eulerian coordinates. Within our application, Lagrangian coordinates will be used, which is more natural to apply to a molecular dynamic simulation compared to the Eulerian coordinates. As a result, the next section describes a derivation for the virial version of the first Piola-Kirchoff (PK) stress tensor. Through the central force decomposition the potential energy of the stress tensor can be broken down into the sum of its components [1]. Namely the energy from bonds, angles, dihedral (torsion) angles, Lennard-Jones (Van der Waals) potential, and the Coulombic potential. The force from these energy components is used to compute the PK stress tensor. This is then transformed into Cauchy stress tensor which will be applied in the generalized Hooke's law computations in Chapter 4.

Chapter 4 shows the computation of the elastic modulus through the generalized Hooke's law. Initially there are eighty-one components to compute, however due to major and minor symmetry combined with the assumption of transversely isotropic material, the system reduces to six components. The engineering constants for transversely isotropic material will be found from the stress and strain. These will be used to compute the compliance matrix, which is the inverse of the elastic modulus. In order to test the code, and to compare our approach with Bathe, a separate model was created where the protein was considered isotropic instead of transversely isotropic. This is followed by a description of the differences of this model, and how the three engineering constants, Young's modulus, Poisson ratio, and Shear modulus are computed.

Chapter 5 introduces a simplistic toy model in order to test the validity of the method. Both a one dimensional and a two dimensional spring are fixed on one end while the opposite end is pulled with a constant velocity. The system is coded into MATLAB and a Verlet integration scheme is used to update the position of the nodes. The spring is considered isotropic.

Chapter 6 reports the results of testing a fictitious toy protein that is an alpha helix. This is followed by the computed Young's moduli of almost all of the twenty amino acids within alpha helical structures. The Young's moduli for coils is also reported. However, due to the short length of the coils, the elastic moduli cannot be computed for all amino acids.

Chapter 7 summarizes the work that has been done. This includes the derivations and computations of the stress and strain tensors, a brief description of the toy models used to test the algorithm and code, and a summary of the results when the algorithm is applied to amino acids with alpha helical and coil secondary structure. Future work regarding the proteins long term deformations is briefly described. This includes created a surface of the protein and applying FEM.

## CHAPTER 2

### STRAIN

#### 2.1 DERIVATION OF THE GREEN-LAGRANGIAN STRAIN TENSOR

The deformed state of the network is described by the Green-Lagrangian strain tensor  $\mathbf{G}$ :

$$\mathbf{C} = \mathbf{F}^T \mathbf{F}, \quad (2.1)$$

$$\mathbf{G} = \frac{1}{2}(\mathbf{C} - \mathbf{I}). \quad (2.2)$$

In its most general form (2.2) this finite strain tensor can be represented in terms of the Right Cauchy-Green deformation tensor  $\mathbf{C}$  Eq. (2.1), which is composed of the deformation tensor  $\mathbf{F}$ .  $\mathbf{F}$  is the mapping of a differential line element from the undeformed configuration,  $d\mathbf{X}$ , to the deformed configuration,  $d\mathbf{x}$  as seen in Eq. (2.3) [8],

$$d\mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} d\mathbf{X} = \mathbf{F} d\mathbf{X}. \quad (2.3)$$

The Green-Lagrangian strain tensor is measured by the difference between the squares

of the differential line elements  $d\mathbf{X}$  and  $d\mathbf{x}$  as seen in Eq. (2.4),

$$\begin{aligned}
|d\mathbf{x}|^2 - |d\mathbf{X}|^2 &= (\mathbf{F} d\mathbf{X})(\mathbf{F} d\mathbf{X}) - d\mathbf{X} d\mathbf{X} \\
&= \mathbf{F}_{ij} d\mathbf{X}_j \mathbf{F}_{ik} d\mathbf{X}_k - d\mathbf{X}_k d\mathbf{X}_k \\
&= (\mathbf{F}_{ij}\mathbf{F}_{ik} - \delta_{jk}) d\mathbf{X}_j d\mathbf{X}_k \\
&= d\mathbf{X}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) d\mathbf{X} \\
&= d\mathbf{X}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) d\mathbf{X} \\
&= d\mathbf{X}(2\mathbf{G}) d\mathbf{X}.
\end{aligned} \tag{2.4}$$

In Eq. (2.4), we see the Green-Lagrangian strain tensor  $\mathbf{G}$ . In one dimension it takes of the form of Eq. (2.5),

$$e_G = \frac{1}{2} \left( \frac{r^2 - r_0^2}{r_0^2} \right), \tag{2.5}$$

where  $r$  is the length of a line segment after deformation and  $r_0$  is the length before. This should be not confused with Cauchy strain tensor or engineering strain tensor as seen in Eq. (2.6),

$$e_C = \frac{r - r_0}{r_0}. \tag{2.6}$$

## 2.2 FIRST COMPUTATION OF THE GREEN-LAGRANGIAN STRAIN TENSOR

Since our model uses the Green-Lagrangian Strain, we will denote the strain tensor using the typical convention by letting  $\mathbf{G} = \epsilon$  where  $\epsilon_i$  represents the strain at a given node  $i$ .  $\epsilon_i$  has 9 components that are decomposed into  $\epsilon_i^{\alpha,\beta}$ , where  $\alpha$  and  $\beta$  represents the  $x$ ,  $y$ , and  $z$  directions in the Cartesian Coordinates. The strain tensor is computed only from backbone nodes. Therefore, within Chapter 2, each subscript  $i$ ,  $j$  does not include side-chain nodes



within this section. The strain tensor is given by Eq (2.7) [9],

$$(r_{i,i+1})^2 - (r_{0,i,i+1})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i,i+1}^\alpha)(\mathbf{r}_{0,i,i+1}^\beta), \quad (2.7)$$

$$\mathbf{r}_{i,j} = (x_i, y_i, z_i) - (x_j, y_j, z_j), \quad (2.8)$$

$$r_{i,j} = \|\mathbf{r}_{i,j}\|_2. \quad (2.9)$$

Each backbone node is bonded with two other backbone nodes forming one long chain of bonded backbone nodes.  $\mathbf{r}_{0,i,i+1}^\alpha$  represents the  $\alpha$  component of the vector starting at node  $i$  and ending at node  $i + 1$  before deformation. Similarly,  $\mathbf{r}$  is the vector between nodes after deformation.  $r_0$  and  $r$  will be the magnitude of  $\mathbf{r}_0$  and  $\mathbf{r}$  before and after deformation respectively. Since  $\epsilon_i^{\alpha,\beta} = \epsilon_i^{\beta,\alpha}$  we have 6 unknowns. Therefore, in order to solve for the components of  $\epsilon_i$ , we will need six equations. These equations will come from the strain of nearby bonded backbone nodes,

$$(r_{i,i+1})^2 - (r_{0,i,i+1})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i,i+1}^\alpha)(\mathbf{r}_{0,i,i+1}^\beta), \quad (2.10)$$

$$(r_{i+1,i+2})^2 - (r_{0,i+1,i+2})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i+1,i+2}^\alpha)(\mathbf{r}_{0,i+1,i+2}^\beta), \quad (2.11)$$

$$(r_{i+2,i+3})^2 - (r_{0,i+2,i+3})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i+2,i+3}^\alpha)(\mathbf{r}_{0,i+2,i+3}^\beta), \quad (2.12)$$

$$(r_{i-1,i})^2 - (r_{0,i-1,i})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i-1,i}^\alpha)(\mathbf{r}_{0,i-1,i}^\beta), \quad (2.13)$$

$$(r_{i-2,i-1})^2 - (r_{0,i-2,i-1})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i-2,i-1}^\alpha)(\mathbf{r}_{0,i-2,i-1}^\beta), \quad (2.14)$$

$$(r_{i-3,i-2})^2 - (r_{0,i-3,i-2})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i-3,i-2}^\alpha)(\mathbf{r}_{0,i-3,i-2}^\beta). \quad (2.15)$$

Unfortunately this approach produced extremely poor results as seen in Figure 2.1. A simplified fictitious protein was created. One with only backbone nodes consisting of the residue GLY. The alpha helical structure formed was rotated to point in the x direction.

The first node was fixed while the last was pulled in the x direction at a constant velocity used the software NAMD. This produced a simple well behaved steered molecular dynamic simulation.

The large spikes are caused by the linear system Eqs. (2.10 - 2.15) becoming numerically close to singular. However much more alarming are the frequent osculations between positive and negative strain tensor along smooth sections within the figures for every component. Because the simulation is extremely well behaved these results are clearly incorrect.

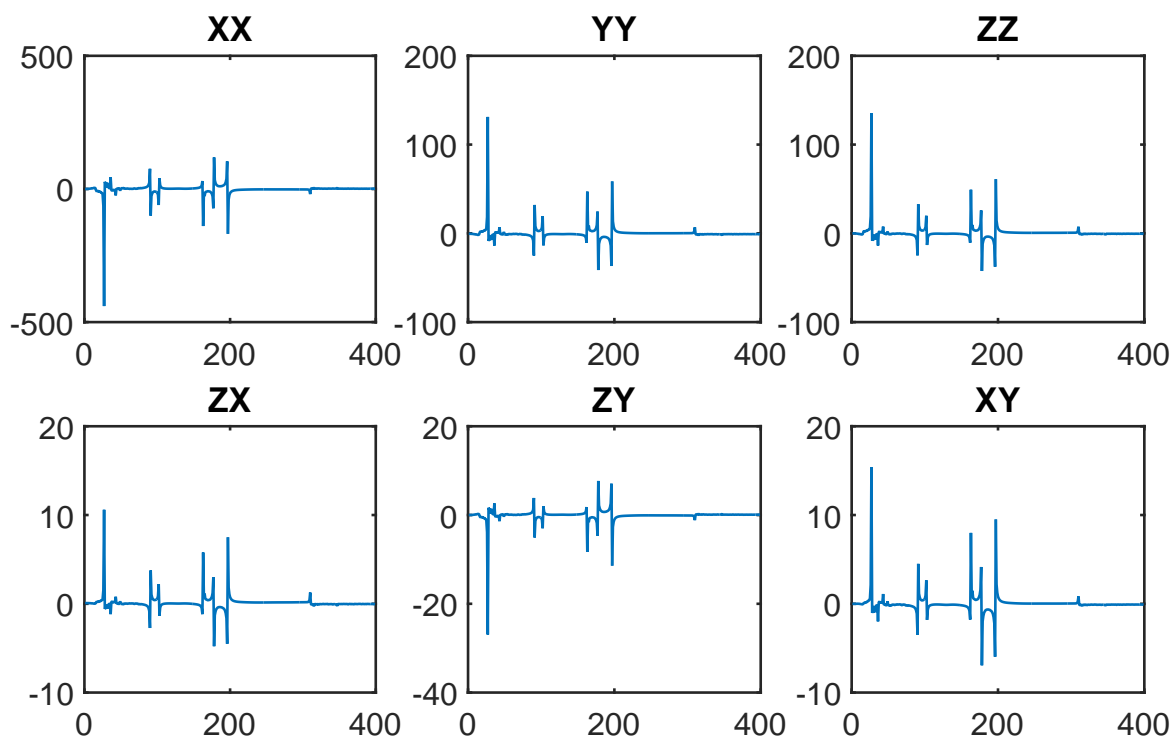


Figure 2.1: Strain components of a node within a fictional protein comprised of only backbone nodes of the amino acid GLY. Computed from Eqs. (2.10 - 2.15)

A slight modification was attempted after considering evidence provided in the toy model described in Section 5.3 in which the bonds vectors were linearly dependent. This was

addressed by changing Eq. (2.11) and Eq. (2.14) into:

$$(r_{i,i+2})^2 - (r_{0,i,i+2})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i,i+2}^\alpha)(\mathbf{r}_{0,i,i+2}^\beta), \quad (2.16)$$

$$(r_{i-2,i})^2 - (r_{0,i-2,i})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i-2,i}^\alpha)(\mathbf{r}_{0,i-2,i}^\beta). \quad (2.17)$$

While this approach does help to smooth out singularities as seen in Figure 2.2, the results are still unrealistic. If only the smooth section is taken, the strain tensor does not significantly change in magnitude. It would be expected that pulling in the  $x$  direction would result in an increased strain in the  $xx$  component of the tensor.

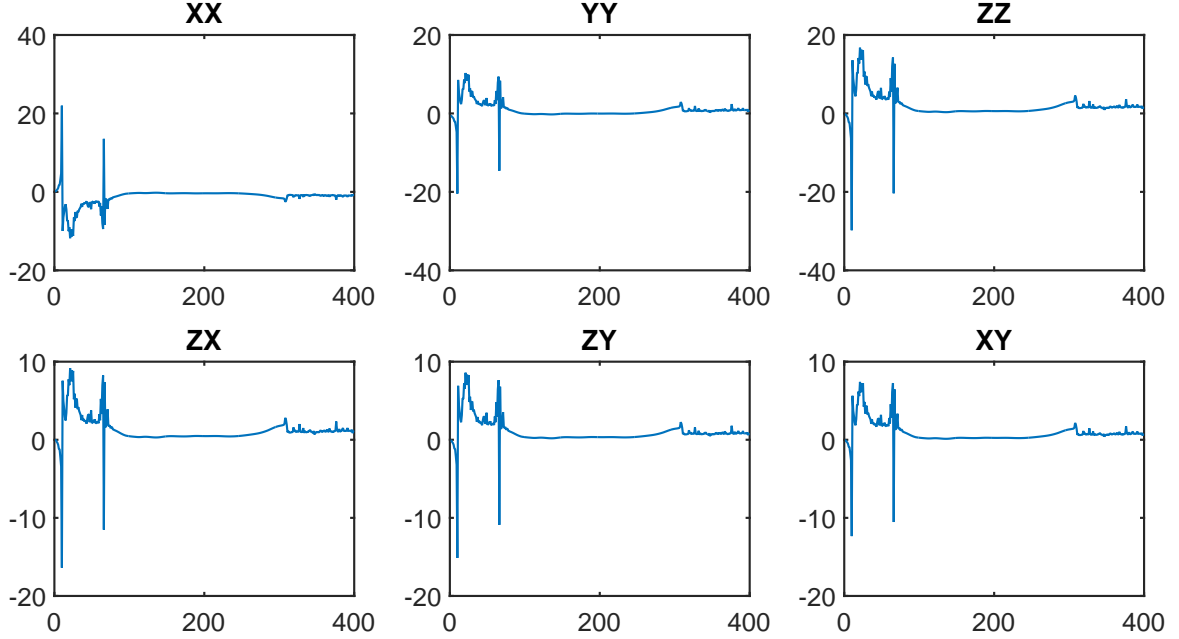


Figure 2.2: Strain components of a fictional protein comprised of only backbone nodes of the amino acid GLY. The  $y$ -axis is the strain. Each point on the  $x$ -axis is a frame during the simulation.

### 2.3 SECOND COMPUTATION OF THE GREEN-LAGRANGIAN STRAIN TENSOR

A second method was attempted in the calculation of the strain tensor in order to compute more realistic results. In this method, the deformation tensor will be computed directly from the change in bond length over the simulation in Eq. (2.3). However, each backbone node where the tensor must be computed, is only bonded to two other backbone nodes. Again, side-chain nodes are excluded in strain tensor calculation. This creates 6 equations to solve for 9 unknowns. Therefore one extra backbone node will be used in each direction to form two separate deformation tensors that will be averaged together in Eq. (2.18),

$$\mathbf{F}_i = \frac{\mathbf{F}_{i,1} + \mathbf{F}_{i,2}}{2}, \quad (2.18)$$

$$\begin{pmatrix} \mathbf{r}_{i-2,i-1} & \mathbf{r}_{i-1,i} & \mathbf{r}_{i,i+1} \end{pmatrix} = \mathbf{F}_{i,1} \begin{pmatrix} \mathbf{r}_{0,i-2,i-1} & \mathbf{r}_{0,i-1,i} & \mathbf{r}_{0,i,i+1} \end{pmatrix}, \quad (2.19)$$

$$\begin{pmatrix} \mathbf{r}_{i-1,i} & \mathbf{r}_{i,i+1} & \mathbf{r}_{i+1,i+2} \end{pmatrix} = \mathbf{F}_{i,2} \begin{pmatrix} \mathbf{r}_{0,i-1,i} & \mathbf{r}_{0,i,i+1} & \mathbf{r}_{0,i+1,i+2} \end{pmatrix}. \quad (2.20)$$

Each  $\mathbf{r}$  is considered a column vector in Eqs. (2.19) and (2.20). Finally the Green-Lagrangian Strain Tensor can be computed using Eq. (2.21) as derived in Eq. (2.4),

$$\epsilon_i = \frac{1}{2}(\mathbf{F}_i^T \mathbf{F}_i - \mathbf{I}). \quad (2.21)$$

## CHAPTER 3

### STRESS

#### 3.1 HISTORY OF THE DERIVATION OF THE STRESS TENSOR

The original scalar equation of virial stress tensor is credited to Clausius [4]. This definition was extended by Maxwell [6] to produce the tensor form of the virial stress tensor. However, the virial stress tensor is only applicable to homogeneous systems, or systems in equilibrium. In the pioneering work by Irving and Kirkwood (IK) [12], this stress tensor was extended to non-equilibrium systems. Noll (English Translation [18]) provided a rigorous mathematical foundation for the IK formalism by providing an explicit form of the local stress tensor.

Due to the Dirac delta functions used in the IK expressions, direct numerical implementation is not straightforward. Hardy [11] overcame this obstacle by replacing the Dirac delta functions with kernel functions, which act as a spatial averaging. Independently Murdoch and co-authors [17] considered both spatial and temporal averaging to address the problem of the Dirac delta functions.

#### 3.2 DERIVATION OF THE CAUCHY STRESS TENSOR IN EULERIAN COORDINATES

The following derivation of the virial stress tensor is a more modern version [1]. To begin, consider a system of  $N$  interacting point masses. The “origin” of this system will be its center of mass,  $\mathbf{x}$ . Therefore the coordinates of each point,  $\mathbf{x}_i$ , will be relative to the center of mass

by their distance from the center of mass  $\hat{\mathbf{r}}_i$ ,

$$\mathbf{x}_i = (x_i, y_i, z_i) = \mathbf{x} + \hat{\mathbf{r}}_i, \quad (3.1)$$

$$\mathbf{x} = \frac{\sum_i m_i \mathbf{x}_i}{\sum_i m_i}. \quad (3.2)$$

Due to defining the coordinates relative to the center of mass, we have the identities in Eq. (3.3) and Eq. (3.4),

$$\sum_i m_i \hat{\mathbf{r}}_i = 0, \quad (3.3)$$

$$\sum_i m_i \mathbf{v}_i^{rel} = 0, \quad (3.4)$$

where  $m_i$  is the mass of node  $i$ , and  $\dot{\hat{\mathbf{r}}}_i = \mathbf{v}_i^{rel}$  is the relative velocity with respect to the center of mass.

Eq. (3.5) is Newton's second law, where  $F_i$  is the force acting on node  $i$ , and  $\rho_i$  is the linear momentum given in equation (3.6),

$$F_i = \dot{\rho}_i, \quad (3.5)$$

$$\rho_i = m_i(\dot{\mathbf{x}}_i + \dot{\hat{\mathbf{r}}}_i) = m_i(\dot{\mathbf{x}}_i + \mathbf{v}_i^{rel}). \quad (3.6)$$

A tensor product is then applied to equation (3.5),

$$\hat{\mathbf{r}}_i \otimes F_i = \hat{\mathbf{r}}_i \otimes \dot{\rho}_i, \quad (3.7)$$

$$\frac{d}{dt}(\hat{\mathbf{r}}_i \otimes \rho_i) = \mathbf{v}_i^{rel} \otimes \rho_i + \hat{\mathbf{r}}_i \otimes \dot{\rho}_i, \quad (3.8)$$

$$\frac{d}{dt}(\hat{\mathbf{r}}_i \otimes \rho_i) = 2T_i + W_i. \quad (3.9)$$

Eq. (3.7) and Eq. (3.8) produces Eq. (3.9) which is called the *dynamical tensor virial theorem*. This can be thought of as an alternative form for the balance of linear momentum.

Here  $W_i = \hat{\mathbf{r}}_i \otimes F_i$  is the *virial tensor* and  $T_i = \frac{1}{2}(\hat{\mathbf{r}}_i \otimes \rho_i)$  is the *kinetic tensor* of node  $i$ . In order to use this “theorem” we must assume that the atoms in the system are in *stationary motion*. For atoms in stationary motion there exists a time scale  $\tau$  which is short relative to macroscopic processes, but long relative to the characteristic time of the atomic system, over which the atoms remain close to their original positions with bounded positions and velocities.

This timescale captures the thermal vibrations of atoms about their mean position, not the motion of the system as a whole. In order to use this property, the time average of any quantity  $q$  over the time  $\tau$  is given in Eq. (3.10), and denoted with a bar above the variable,

$$\bar{q} = \frac{1}{\tau} \int_0^\tau q(t) dt. \quad (3.10)$$

This can be applied to Eq. (3.9) to produce Eq. (3.11),

$$\left. \frac{1}{\tau} (\hat{\mathbf{r}}_i \otimes \rho_i) \right|_0^\tau = \bar{W}_i + 2\bar{T}_i. \quad (3.11)$$

The term on the left-hand side of Eq. (3.11) can be made as small as desired by taking  $\tau$  sufficiently large given that  $\hat{\mathbf{r}}_i \otimes \rho_i$  is bounded, and a separation of time scales between microscopic and continuum processes exists. Therefore Eq. (3.9) reduces to the *tensor virial theorem* in Eq. (3.12), after summing over all nodes,

$$\bar{W} = \sum_i \bar{W}_i = -2 \sum_i \bar{T}_i = -2\bar{T}. \quad (3.12)$$

Substituting Eq. (3.6) into Eq. (3.13) yields Eq. (3.14),

$$\bar{T} = \frac{1}{2} \sum_i m_i \overline{\mathbf{v}_i^{rel} \otimes \rho_i}, \quad (3.13)$$

$$\bar{T} = \frac{1}{2} \sum_i m_i \overline{\mathbf{v}_i^{rel} \otimes \mathbf{v}_i^{rel}} + \left[ \overline{\sum_i m_i \mathbf{v}_i^{rel}} \right] \otimes \dot{\mathbf{x}}, \quad (3.14)$$

Here due to the separation of time scales,  $\dot{\mathbf{x}}$  can be considered constant on the atomistic time scale  $\tau$ . This means that continuum motion occurs so slowly relative to atomistic processes, that the motion is essentially constant on this time scale. This allows for the application of the virial theorem to be applied to both continuum and discrete systems, that are in either equilibrium, or in a state of motion. Furthermore, due to Eq. (3.3), the term inside the square brackets is zero simplifying to (3.15),

$$\overline{T} = \frac{1}{2} \sum_i m_i \overline{\mathbf{v}_i^{rel} \otimes \mathbf{v}_i^{rel}}. \quad (3.15)$$

The force acting on each node can be decomposed into the force due to other nodes within the system,  $F_i^{int}$ , and an external force,  $F_i^{ext}$ ,

$$F_i = F_i^{int} + F_i^{ext}, \quad (3.16)$$

Because of this force decomposition, the virial tensor can be decomposed into internal and external as well,

$$\overline{W} = \overline{W}_{int} + \overline{W}_{ext} = \sum_i \overline{\hat{\mathbf{r}}_i \otimes F_i^{int}} + \sum_i \overline{\hat{\mathbf{r}}_i \otimes F_i^{ext}}. \quad (3.17)$$

From the perspective of continuum mechanics, the external force can be identified with the mass density,  $\rho$ , multiplied by the body force per unit mass,  $b$ , and the traction  $t$ . This leads to Eq. (3.18),

$$\overline{W}_{ext} = \sum_i \overline{\hat{\mathbf{r}}_i \otimes F_i^{ext}} := \int_{\Omega} \mathbf{x} \otimes \rho b \, dV + \int_{\partial\Omega} \mathbf{x} \otimes t \, dA, \quad (3.18)$$

where  $\Omega$  is the domain of the system, and  $\partial\Omega$  is the closed surface bounding the system.  $\mathbf{x}$  is a position vector in the domain  $\Omega$  and the boundary  $\partial\Omega$ .

Substituting in the Cauchy relation  $t = \sigma \cdot \mathbf{n}$  into Eq. (3.18), and applying the divergence



theorem in Eq. (3.19) along with the product rule yields Eq. (3.20),

$$\int_{\partial\Omega} \mathbf{x} \otimes (\boldsymbol{\sigma} \cdot \mathbf{n}) dA = \int_{\Omega} \mathbf{x} \otimes (\nabla_{\mathbf{x}} \cdot \boldsymbol{\sigma}) dV, \quad (3.19)$$

$$\begin{aligned} \overline{W}_{ext} &= \int_{\Omega} [\mathbf{x} \otimes \rho b + \nabla_{\mathbf{x}}(\mathbf{x} \otimes \boldsymbol{\sigma})] dV \\ &= \int_{\Omega} [\boldsymbol{\sigma}^T + \mathbf{x} \otimes (\rho b + \nabla_{\mathbf{x}} \boldsymbol{\sigma})] dV, \end{aligned} \quad (3.20)$$

where  $\boldsymbol{\sigma}$  is the pointwise Cauchy stress tensor, and  $\mathbf{n}$  is the vector normal to the surface of the boundary  $\partial\Omega$ . The term  $(\rho b + \nabla_{\mathbf{x}} \boldsymbol{\sigma})$  is zero under equilibrium conditions with the assumption that the pointwise fields satisfy the same balance laws as the macroscopic continuum fields [1] producing Eq. (3.21),

$$\overline{W}_{ext} = V \boldsymbol{\sigma}^T. \quad (3.21)$$

The continuum stress field can be defined as the average value of  $\boldsymbol{\sigma}$  over the domain  $\Omega$  shown in Eq. (3.22),

$$\boldsymbol{\sigma}_{avg} = \frac{1}{V} \int_{\Omega} \boldsymbol{\sigma} dV, \quad (3.22)$$

$$\begin{aligned} &= -\frac{1}{V} [\overline{W}_{int} + 2\overline{T}]^T \\ &= -\frac{1}{V} \left[ \sum_i \hat{\mathbf{r}}_i \otimes \overline{F_i^{int}} + \sum_i m_i \overline{\mathbf{v}_i^{rel}} \otimes \overline{\mathbf{v}_i^{rel}} \right]^T \\ \boldsymbol{\sigma}_{avg} &= -\frac{1}{V} \left[ \sum_i \overline{F_i^{int}} \otimes \hat{\mathbf{r}}_i + \sum_i m_i \overline{\mathbf{v}_i^{rel}} \otimes \overline{\mathbf{v}_i^{rel}} \right]. \end{aligned} \quad (3.23)$$

Through several substitutions, the virial stress tensor in Eq. (3.23) is derived. This virial stress tensor is an expression for the Cauchy stress tensor given solely in terms of nodal quantities.

The symmetry of the virial stress tensor still needs to be shown. The internal force  $F_i^{int}$

can be decomposed into a summation between the all the nodes acting on node  $i$ ,

$$F_i^{int} = \sum_{\substack{j \\ i \neq j}} F_{ij}, \quad (3.24)$$

$$\begin{aligned} \sum_{\substack{i,j \\ i \neq j}} F_{ij} \otimes \hat{\mathbf{r}}_i &= \frac{1}{2} \sum_{\substack{i,j \\ i \neq j}} (F_{ij} \otimes \hat{\mathbf{r}}_i + F_{ji} \otimes \hat{\mathbf{r}}_i) \\ &= \sum_{\substack{i,j \\ i \neq j}} F_{ij} \otimes (\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j), \end{aligned} \quad (3.25)$$

where Eq. (3.25) makes use of the identity  $F_{ij} = -F_{ji}$ . Substituting Eq. (3.25) into the virial stress tensor yields Eq. (3.26),

$$\sigma_{avg} = -\frac{1}{V} \left[ \sum_{\substack{i,j \\ i \neq j}} \overline{F_{ij} \otimes (\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j)} + \sum_i m_i \overline{\mathbf{v}_i^{rel} \otimes \mathbf{v}_i^{rel}} \right]. \quad (3.26)$$

This alternate version of the virial stress tensor shows its symmetric property due the fact that  $F_{ij}$  is parallel to  $\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j$ .

### 3.3 DERIVATION OF THE FIRST PIOLA-KIRCHOFF STRESS TENSOR IN LAGRANGIAN COORDINATES

We begin by letting  $\mathbf{X} \in \mathbb{R}$  be a point in the reference frame. One of the equations from continuum mechanics models are often written as in Eq. (3.27).  $\mathbf{q}$  is the spatially averaged momentum and  $\mathbf{P}$  is the first Piola-Kirchhoff stress tensor,

$$\frac{\partial}{\partial t} \mathbf{q} = \nabla_{\mathbf{x}} \cdot \mathbf{P}. \quad (3.27)$$

Stress can be defined based on the Irving-Kirkwood formalism [12]. This begins by defining:

$$\tilde{\mathbf{q}}(\mathbf{X}, t) = \sum_i m_i \mathbf{v}_i \varphi(\mathbf{X} - \mathbf{X}_i), \quad (3.28)$$

where  $\varphi$  is refereed to as a weight function. The Irving-Kirkwood formula for stress tensor was derived using the Dirac delta function making direct computation of the tensor difficult. However Hardy introduced these weight functions in order to simplify the computation within actual implementation. While the weight function is not unique, Hardy established some criteria in selecting  $\varphi$  [11]:

- $\varphi(\mathbf{X}_i - \mathbf{X})$  has its maximum at  $\mathbf{X}_i = \mathbf{X}$ ,
- $\varphi(\mathbf{X}_i - \mathbf{X}) \rightarrow 0$  is as  $|\mathbf{X}_i - \mathbf{X}| \rightarrow \infty$ ,
- $\varphi(\mathbf{X}_i - \mathbf{X})$  is smooth an non-negative,
- $\int_{\mathbb{R}^3} \varphi(\mathbf{X}_i - \mathbf{X}) d\mathbf{X} = 1$ .

Following the conservation laws in Eq. (3.27), we can take the derivative with respect to time of Eq. (3.28).

$$\begin{aligned} \frac{\partial}{\partial t} \tilde{\mathbf{q}} &= \frac{\partial}{\partial t} \sum_{i=1}^N m_i \mathbf{v}_i \varphi(\mathbf{X} - \mathbf{X}_i), \\ &= \sum_{i=1}^N F_i \varphi(\mathbf{X} - \mathbf{X}_i). \end{aligned} \quad (3.29)$$

Taking advantage of the fact that  $F_{ij} = -F_{ji}$  it follows that,

$$\begin{aligned}
\frac{\partial}{\partial t} \tilde{\mathbf{q}} &= \sum_{i=1}^N \sum_{j \neq i} \frac{1}{2} F_{ij} [\varphi(\mathbf{X} - \mathbf{X}_i) + \varphi(\mathbf{X} - \mathbf{X}_j)] \\
&= \sum_{i=1}^N \sum_{j \neq i} \frac{1}{2} F_{ij} \varphi(\mathbf{X} - \mathbf{X}_i) + \sum_{i=1}^N \sum_{j \neq i} \frac{1}{2} F_{ij} \varphi(\mathbf{X} - \mathbf{X}_j) \\
&= \sum_{i=1}^N \sum_{j \neq i} \frac{1}{2} F_{ij} \varphi(\mathbf{X} - \mathbf{X}_i) + \sum_{i=1}^N \sum_{j \neq i} \frac{1}{2} F_{ji} \varphi(\mathbf{X} - \mathbf{X}_j) \\
&= \sum_{i=1}^N \sum_{j \neq i} \frac{1}{2} F_{ij} \varphi(\mathbf{X} - \mathbf{X}_i) - \sum_{i=1}^N \sum_{j \neq i} \frac{1}{2} F_{ij} \varphi(\mathbf{X} - \mathbf{X}_j) \\
&= \sum_{i=1}^N \sum_{j \neq i} \frac{1}{2} F_{ij} [\varphi(\mathbf{X} - \mathbf{X}_i) + \varphi(\mathbf{X} - \mathbf{X}_i)].
\end{aligned} \tag{3.30}$$

Next is to define the bond function  $b_{ij}(\mathbf{X})$ ,

$$b_{ij}(\mathbf{X}) \doteq \int_0^1 \varphi(\mathbf{X} - (\mathbf{X}_i + \lambda \mathbf{r}_{0,ij})) d\lambda, \tag{3.31}$$

$$\frac{\partial \varphi(\mathbf{X} - (\mathbf{X}_i + \lambda \mathbf{r}_{0,ij}))}{\partial \lambda} = -\mathbf{r}_{0,ij} \cdot \nabla \varphi, \tag{3.32}$$

$$\begin{aligned}
-\mathbf{r}_{0,ij} \cdot \nabla b_{ij} &= \int_0^1 -\mathbf{r}_{0,ij} \cdot \nabla \varphi(\mathbf{X} - (\mathbf{X}_i + \lambda \mathbf{r}_{0,ij})) d\lambda \\
&= \int_0^1 \frac{\partial \varphi(\mathbf{X} - (\mathbf{X}_i + \lambda \mathbf{r}_{0,ij}))}{\partial \lambda} d\lambda \\
&= \varphi(\mathbf{X} - (\mathbf{X}_i + \lambda \mathbf{r}_{0,ij})) \Big|_0^1 \\
&= \varphi(\mathbf{X} - \mathbf{X}_i) - \varphi(\mathbf{X} - \mathbf{X}_j).
\end{aligned} \tag{3.33}$$

Substituting Eq. (3.33) into Eq. (3.30) yields,

$$\begin{aligned}\frac{\partial}{\partial t}\tilde{\mathbf{q}} &= \sum_{i=1}^N \frac{1}{2} \sum_{j \neq i} -F_{ij} \mathbf{r}_{0,ij} \cdot \nabla b_{ij}(\mathbf{X}) \\ &= \nabla \cdot \left( -\frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} F_{ij} \otimes \mathbf{r}_{0,ij} b_{ij}(\mathbf{X}) \right).\end{aligned}\quad (3.34)$$

By looking at Eq. (3.27) we see the Hardy stress tensor,

$$\mathbf{P}^{Hardy}(\mathbf{X}, t) = -\frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} F_{ij} \otimes \mathbf{r}_{0,ij} b_{ij}(\mathbf{X}). \quad (3.35)$$

The virial stress tensor is a special case of the Hardy stress tensor which can be seen by letting the weight function be constant on its support. The bond function is then given by,

$$b_{ij}(\mathbf{X}) = \begin{cases} 1/\text{vol}(\Omega) & \text{if the bond between nodes } i \text{ and } j \in \Omega, \\ 0 & \text{Otherwise.} \end{cases} \quad (3.36)$$

Given that our domain  $\Omega$  is the entire volume of the protein the virial stress tensor is derived [22],

$$\mathbf{P}^v = -\frac{1}{2V} \sum_i \sum_{j \neq i} F_{ij} \otimes \mathbf{r}_{0,ij}. \quad (3.37)$$

### 3.4 COMPUTATION OF STRESS

Due to the discrete nature of model of the protein, Lagrangian coordinates are the natural choice over Eulerian coordinates. Therefore, the first Piola-Kirchoff stress tensor will be computed using Eq. (3.37). The volume,  $V$ , used in Eq. (3.37) will be found using a density of  $1170 \frac{kg}{m^3}$  [2], since the mass of the amino acids are known. This density is an average measured from a large sample of proteins.

The energy from bonds, angles, dihedral angles, Lenard Jones potential, and Coulumbic

potential will all contribute to the potential energy. This is referred to the central force decompositions [1],

$$F_{i,j} = -\nabla E_{i,j}, \quad (3.38)$$

$$E_{i,j} = E_{i,j}^{\text{bd}} + E_{i,j}^{\text{ang}} + E_{i,j}^{\text{dih}} + E_{i,j}^{\text{LJ}} + E_{i,j}^{\text{el}}. \quad (3.39)$$

The derivative of each of the energy quantities will be computed, and then summed to compute  $F_{ij}$  in Eq. (3.37). In order to compute the engineering constants, the Generalized Hooke's Law implements the Cauchy stress tensor,  $\sigma$ , instead of the first Piola-Kirchoff stress tensor. Therefore the classical relationship in Eq. (3.40) will be used where  $\mathbf{F}$  is the deformation tensor [23],

$$\sigma = \det(\mathbf{F})^{-1} \mathbf{F} \cdot \mathbf{P}. \quad (3.40)$$

This model is at the resolution of amino acids. Therefore, we are only interested in the stress, and as a results energy of backbone nodes that represent their amino acid.

### 3.4.1 BOND POTENTIAL ENERGY

Nodes that are bonded together are treated in the simulation as if they are attached by a spring. Therefore Hooke's Law can be applied to Eq. (3.41).  $E^{bd}$  is the potential energy from two bonded nodes, and  $k_b$  is the spring constant.  $r_{i,j}$  and  $r_{0,i,j}$  is the distance between nodes  $i$  and  $j$  after the simulation, and at equilibrium respectively. At equilibrium the distance will be constant for all backbone-backbone bonds set at 0.35 nm [16].  $x_{i,j}$  is the  $x$ -component of

the vector between nodes  $i$  and  $j$ ,  $\mathbf{r}_{i,j}$ , and  $x_{i,j} = x_i - x_j$ ,

$$\begin{aligned} E_{i,j}^{\text{bd}} &= \frac{1}{2} k_b (r_{i,j} - r_{0,i,j})^2 \\ &= \frac{1}{2} k_b \left( \sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2} - r_{0,i,j} \right)^2, \end{aligned} \quad (3.41)$$

$$F_{i,j}^x = -k_b \left( \sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2} - r_{0,i,j} \right) \frac{1}{\sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2}} x_{i,j}, \quad (3.42)$$

$$F_{j,i}^x = -k_b \left( \sqrt{x_{j,i}^2 + y_{j,i}^2 + z_{j,i}^2} - r_{0,j,i} \right) \frac{1}{\sqrt{x_{j,i}^2 + y_{j,i}^2 + z_{j,i}^2}} x_{j,i}, \quad (3.43)$$

$$\begin{aligned} F_{i,j} &= -k_b \left( 1 - \frac{r_{0,i,j}}{\sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2}} \right) \langle x_{i,j}, y_{i,j}, z_{i,j} \rangle \\ &= -k_b \left( 1 - \frac{r_{0,i,j}}{\sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2}} \right) \mathbf{r}_{i,j} \\ &= -k_b \left( 1 - \frac{r_{0,i,j}}{\sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2}} \right) (-\mathbf{r}_{j,i}) = -F_{j,i}. \end{aligned} \quad (3.44)$$

Two important properties of the force are  $F_{i,j}$  is parallel to the direction of  $\mathbf{r}_{i,j}$  and  $F_{i,j} = -F_{j,i}$ . Part of this force will remain constant no matter which cardinal direction component is taken. This will be taken advantage of in the code,

$$F_c = k_b \left( 1 - \frac{r_{0,i,j}}{\sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2}} \right). \quad (3.45)$$

### 3.4.2 BONDING ANGLE POTENTIAL ENERGY

An angle energy is needed so any three nodes maintain their shape within a plane.  $\theta$  is a function of  $(x_1, y_1, z_1, x_2, y_2, z_2)$ , where  $x_1 = x_{i,j}$ ,  $x_2 = x_{j,k}$ .  $\Theta$  is the initial angle between the three nodes (during equilibrium), and  $\theta$  is the angle between the same nodes during the simulation. This energy involves three nodes each with three coordinates  $x$ ,  $y$ , and  $z$ . The vector from the node  $j$  to  $i$  is  $\mathbf{r}_{i,j}$ , while the vector from the node  $j$  to  $k$  is  $\mathbf{r}_{k,j}$ . The angle

between these vectors is computed from the dot product:

$$E_{i,j,k}^{\text{ang}} = \frac{1}{2}k_{\theta}(\cos(\theta) - \cos(\Theta))^2, \quad (3.46)$$

$$\cos(\theta) = \frac{\mathbf{r}_{i,j} \cdot \mathbf{r}_{k,j}}{\|\mathbf{r}_{i,j}\| \|\mathbf{r}_{k,j}\|} = \frac{(x_1x_2 + y_1y_2 + z_1z_2)}{\|\mathbf{r}_{i,j}\| \|\mathbf{r}_{k,j}\|}, \quad (3.47)$$

$$\begin{aligned} x_1x_2 &= (x_i - x_j)(x_k - x_j) \\ &= x_ix_k - x_ix_j - x_jx_k + x_j^2. \end{aligned} \quad (3.48)$$

This produces nine degrees of freedom for each  $E_{i,j,k}^{\text{ang}}$ . However we are only concerned with changes in the angle between these nodes so some of the degrees of freedom can be eliminated. First, if we translate all three nodes equally, the angle will not change. So three degrees of freedom can be ignored for this case. If the middle node, denoted as node  $j$ , remains fixed and the other two nodes rotate around node  $j$  symmetrically, this does not affect the angle. This eliminates two more degrees of freedom. Finally, if the vector of the distance between nodes  $i$  and  $k$  changes in magnitude but preserves its direction, this will not affect the angle. This included with the vector between nodes  $j$  and  $k$  reduce the degrees of freedom by two. So we are left with  $9 - 3 - 2 - 2 = 2$  degrees of freedom.



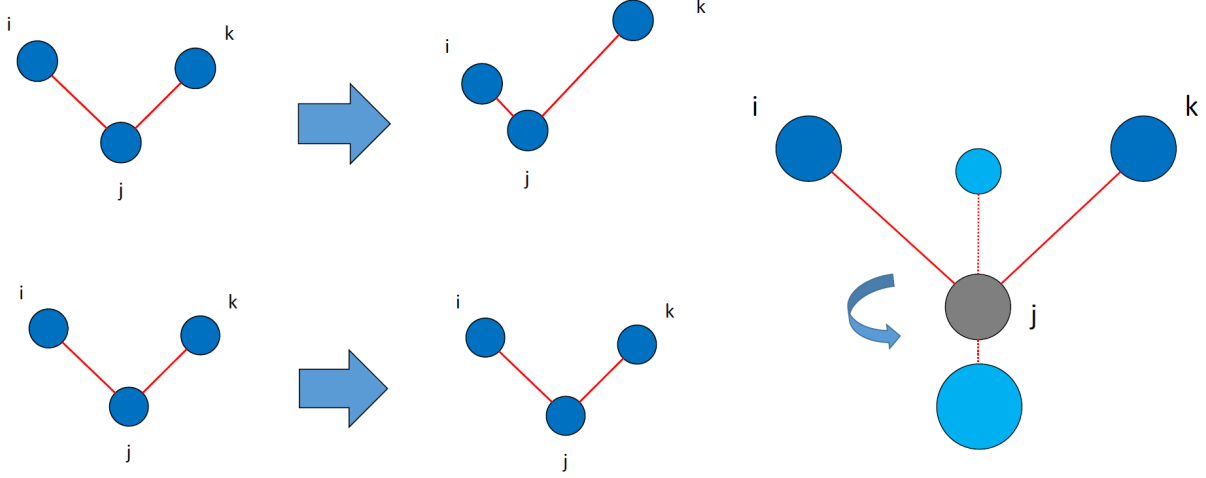


Figure 3.1: Three types of motion of a three node system that do not affect the angle formed. Top left: Change in magnitude of the bond lengths. Bottom left: Translation of the system. Right: Rotation of equal magnitude of nodes  $i$  and  $k$  about node  $j$ .

The force is computed by taking the derivative of the energy,

$$\begin{aligned}
F_{i,j}^x &= -\frac{\partial E_{i,j,k}^{\text{ang}}}{\partial x_{i,j}} = -\frac{\partial E_{i,j,k}^{\text{ang}}}{\partial \theta} \frac{\partial \theta}{\partial x_{i,j}} \\
&= -k_\theta (\cos(\theta) - \cos(\Theta)) (-\sin(\theta)) \left( \frac{-1}{\sqrt{1 - \left( \frac{\mathbf{r}_{i,j} \cdot \mathbf{r}_{k,j}}{\|\mathbf{r}_{i,j}\| \|\mathbf{r}_{k,j}\|} \right)^2}} \right) \\
&\quad \frac{\partial}{\partial x_{i,j}} \left( \frac{x_{i,j}x_{k,j} + y_{i,j}y_{k,j} + z_{i,j}z_{k,j}}{\sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2} \sqrt{x_{k,j}^2 + y_{k,j}^2 + z_{k,j}^2}} \right) \\
&= -k_\theta (\cos(\theta) - \cos(\Theta)) \left( \frac{x_{k,j}r_{i,j}r_{k,j} - (\mathbf{r}_{i,j} \cdot \mathbf{r}_{k,j})r_{k,j}\frac{1}{2}(x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^{-\frac{1}{2}}2x_{i,j}}{r_{i,j}^2 r_{k,j}^2} \right) \\
&= \frac{-k_\theta (\cos(\theta) - \cos(\Theta))}{r_{i,j}r_{k,j}} \left( x_{k,j} - \frac{x_{i,j}(\mathbf{r}_{i,j} \cdot \mathbf{r}_{k,j})}{r_{i,j}^2} \right), \\
F_{k,j}^x &= \frac{-k_\theta (\cos(\theta) - \cos(\Theta))}{r_{i,j}r_{k,j}} \left( x_{i,j} - \frac{x_{k,j}(\mathbf{r}_{i,j} \cdot \mathbf{r}_{k,j})}{r_{i,j}^2} \right), \\
F_{i,j}^y &= \frac{-k_\theta (\cos(\theta) - \cos(\Theta))}{r_{i,j}r_{k,j}} \left( y_{k,j} - \frac{y_{i,j}(\mathbf{r}_{i,j} \cdot \mathbf{r}_{k,j})}{r_{i,j}^2} \right).
\end{aligned}$$

### 3.4.3 DIHEDRAL ANGLE POTENTIAL ENERGY

The dihedral angle energy between four nodes  $i$ ,  $j$ ,  $k$ , and  $l$  is computed from the angle between the planes formed by the first three nodes, and the last three nodes.  $k_\phi$  is the force constant,  $\delta$  is the phase shift.  $\phi$  is the angle formed between the normal vector of the plane formed by the  $i$ ,  $j$ ,  $k$  nodes, and the normal vector of the plane formed by the  $j$ ,  $k$ ,  $l$  nodes. The multiplicity,  $n$ , will be set to 1,

$$E_{i,j,k,l}^{\text{dih}} = k_\phi(1 - \cos(n\phi - \delta)). \quad (3.49)$$

A similar approach can be taken in finding the degrees of freedom. Initially four nodes with three coordinates each produce twelve degrees of freedom. three are the translation of the system. three are adjusting the magnitude of each vector without changing direction. Two more from rotation of the system. The last two arise if we fix  $\mathbf{r}_{j,k}$  and rotate  $\mathbf{r}_{i,j}$  and  $\mathbf{r}_{k,l}$  within their respective planes. So we are left with  $12 - 3 - 3 - 2 - 2 = 2$  degrees of freedom.

In order to find  $\phi$ , the equations for each plane must be computed. The equation of the  $i$ ,  $j$ ,  $k$  plane can be computed using the cross product of two vectors in the space,

$$\mathbf{r}_{i,j} \times \mathbf{r}_{j,k} = a_1 \hat{i} + b_1 \hat{j} + c_1 \hat{k}, \quad (3.50)$$

$$a_1 = y_1 z_2 - z_1 y_2, \quad (3.51)$$

$$b_1 = z_1 x_2 - x_1 z_2, \quad (3.52)$$

$$c_1 = x_1 y_2 - y_1 x_2, \quad (3.53)$$

where,

$$\begin{aligned}
x_1 &= x_i - x_j, & x_2 &= x_k - x_j, \\
y_1 &= y_i - y_j, & y_2 &= y_k - y_j, \\
z_1 &= z_i - z_j, & z_2 &= z_k - z_j.
\end{aligned}$$

The equation for the plane is given in Eq. (3.54),

$$0 = a_1x + b_1y + c_1z + d_1, \quad (3.54)$$

where  $d_1$  is a constant that we do not need to compute since we only wish to extract the normal vector to this plane,

$$n_1 = (a_1, b_1, c_1). \quad (3.55)$$

The equation of the normal vector to the plane formed by the  $j$ ,  $k$ , and  $l$  nodes is computed in a similar manner yielding Eq. (3.56),

$$n_2 = (a_2, b_2, c_2). \quad (3.56)$$

The angle between these two normal vectors is given in Eq. (3.57),

$$\begin{aligned}
\phi &= \arccos \left( \frac{n_1 \cdot n_2}{\|n_1\| \|n_2\|} \right) \\
&= \arccos \left( \frac{a_1a_2 + b_1b_2 + c_1c_2}{\|n_1\| \|n_2\|} \right).
\end{aligned} \quad (3.57)$$

This allows for the computation of the force,

$$F_{i,j}^x = -nk_\phi \sin(n\phi - \delta) \left( -\frac{1}{\sqrt{1 - \cos^2 \phi}} \right) \left[ - (y_{i,j}z_{k,j} - z_{i,j}y_{k,j})(x_{k,j}^2 + y_{k,j}^2 + z_{k,j}^2)(x_{l,k}y_{k,j}z_{i,j} - x_{k,j}y_{l,k}z_{i,j} - x_{l,k}y_{i,j}z_{k,j} + x_{i,j}y_{l,k}z_{k,j} + x_{k,j}y_{i,j}z_{l,k} - x_{i,j}y_{k,j}z_{l,k}) \right] / (\|n_1\|^3 \|n_2\|).$$

### 3.4.4 LENNARD-JONES POTENTIAL ENERGY

Unlike the bonded interactions, all particle pairs interact via a Lennard-Jones potential energy. However, to avoid unwanted noise, in this model we will only consider particles that are within a cutoff distance of  $r_{\text{cut}} = 1.2\text{nm}$ . Furthermore, the MARTINI Force Field excludes non bonded potentials with nearest neighbors [16]. In our simulation this was applied by ignoring Lennard Jones potentials between pairs of nodes that were already bonded. The force is once again computed by taking the derivative of the energy,

$$\begin{aligned} E_{i,j}^{\text{LJ}} &= 4\gamma_{i,j} \left[ \left( \frac{\mu_{i,j}}{r_{i,j}} \right)^{12} - \left( \frac{\mu_{i,j}}{r_{i,j}} \right)^6 \right] \\ &= 4\gamma_{i,j} \left[ \left( \frac{\mu_{i,j}}{\sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2}} \right)^{12} - \left( \frac{\mu_{i,j}}{\sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2}} \right)^6 \right] \\ &= 4\gamma_{i,j} \left[ \frac{\mu_{i,j}^{12}}{(x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^6} - \frac{\mu_{i,j}^6}{(x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^3} \right], \end{aligned} \quad (3.58)$$

$$\begin{aligned} F_{i,j}^x &= -4\gamma_{i,j} \left[ -6 \frac{\mu_{i,j}^{12}}{(x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^7} + 3 \frac{\mu_{i,j}^6}{(x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^4} \right] 2x_{i,j} \\ &= -24\gamma_{i,j} \left[ -2 \frac{\mu_{i,j}^{12}}{(x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^7} + \frac{\mu_{i,j}^6}{(x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^4} \right] x_{i,j}, \end{aligned} \quad (3.59)$$

where  $\gamma_{i,j}$  is the depth of the potential well, and  $\mu_{i,j}$  is the is the finite distance at which the inter-particle potential is zero.  $\gamma_{i,j}$  can be between 5.6 to 2.0 kJ/mol depending on the particle.  $\mu_{i,j} = 0.47$  nm for all normal particle types. For the special class of particles used for ringlike molecules, slightly reduced parameters are defined to model ringring interactions:

$\mu_{i,j} = 0.43$  nm and  $\gamma_{i,j}$  is scaled to 75% of the standard value [16].

### 3.4.5 COULOMBIC POTENTIAL ENERGY

The Coulombic Energy describes the energy resulting from the electromagnetic force of two particles. Once again a cutoff distance of  $r_{\text{cut}} = 1.2\text{nm}$  will be used,

$$E_{i,j}^{\text{el}} = \frac{q_i q_j}{4\pi\epsilon_{\text{rel}}\epsilon_0 r_{i,j}} = \frac{q_i q_j k_e}{\epsilon_{\text{rel}} \sqrt{x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2}}, \quad (3.60)$$

$$k_e = \frac{1}{4\pi\epsilon_0} = 8.9875517873681764 \times 10^9 \text{ N m}^2 \text{ C}^{-2}, \quad (3.61)$$

where  $\epsilon_0$  is the permittivity of free space, and  $k_e$  is Coulumb's constant.  $\epsilon_{\text{rel}}$  is the relative permittivity which will be fixed at  $\epsilon_{\text{rel}} = 15$ .  $q_i$  is the charge of node  $i$ . Only nodes within a cut-off distance of 1.2nm, that both contain a charge will be considered. Since the MARTINI Force Field only includes the charge of an amino acid within the last side-chain node, the exclusion of the non-bonded potential energy of the nearest neighbors does not apply. The  $x$ -component of the force will be,

$$F_{i,j}^x = -\frac{\partial E_{i,j}^{\text{el}}}{\partial x_{i,j}} = \frac{1}{2} \frac{q_i q_j k_e}{\epsilon_{\text{rel}}} (x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^{-\frac{3}{2}} 2x_{i,j}, \quad (3.62)$$

$$= \frac{q_i q_j k_e}{\epsilon_{\text{rel}}} (x_{i,j}^2 + y_{i,j}^2 + z_{i,j}^2)^{-\frac{3}{2}} x_{i,j}. \quad (3.63)$$

## 3.5 SIMULATION

The atomic structures of many proteins can be found freely online at the protein data bank. The vast majority of structures are computed using X-Ray crystallography. However, this technique is an estimation of the protein. In order to "smooth out" any irregularities within the protein data bank file (pdb), an atomic simulation was performed.

The atomic resolution coordinate pdb file obtained from the protein data bank was solvated in water using the software VMD. An Another atomic simulation was performed

where both the protein and water were allowed to move. This simulation had a minimization of five thousand time-steps and forty thousand time-steps of simulation. The resulting protein structure was coarse grained using the MARTINI Force Field. Each protein was pulled at a constant velocity ranging from  $1.0\text{e-}5$  to  $1.0\text{e-}6$  Angstroms per time-step. Each time-step is two femtoseconds. The simulation was run for four hundred thousand time-steps using the software NAMD without water. Different proteins were pulled at different speeds in an attempt to produce a smooth simulation, along with 1000 time-steps of minimization after every 1000 time-steps of simulation. All alpha helices were rotated to be parallel to the  $x$  axis and pulled in the  $x$  direction.

Some proteins required additional simulations to be performed. These consisted of fixing the solvated atomic protein, and allow the water to move. This “soaked” the water into the protein. Another possible simulation to take place was running a coarse-grained simulation with the protein solvated. Lastly some proteins required a simulation to be run on the coarse-grained structure, without water, where strategic nodes were fixed to prevent the protein from buckling onto itself.

## CHAPTER 4

### CONTINUUM ELASTIC PROPERTIES OF THE MATERIAL

#### 4.1 TRANSVERSELY ISOTROPIC MATERIAL

From the stress and strain tensors one can compute the anisotropic elasticity tensor,  $C$ , called the elastic modulus. This will be computed at each backbone node, which will describe the protein's response to loads [7].  $C$  can be computed using the generalized Hooke's law,

$$\sigma^{\alpha\beta} = \sum_{\gamma\delta} C^{\alpha\beta\gamma\delta} \epsilon^{\gamma\delta}. \quad (4.1)$$

As in computing the strain tensor, only backbone nodes will be considered since they represent their respective amino acid.  $\alpha, \beta, \gamma$ , and  $\delta$  represent the three cardinal directions.  $C^{\alpha\beta\gamma\delta}$  is a fourth-order tensor, meaning it will have  $3^4 = 81$  components. Since  $\sigma$  and  $\epsilon$  are symmetric, only thirty-six components of the elastic tensor are independent, shown below. This is referred to as minor symmetry.  $C$  also has major symmetry since  $C^{\alpha\beta\gamma\delta} = C^{\gamma\delta\alpha\beta}$  [10]. Therefore  $C$  has twenty-one independent components. This relation can be expressed in matrix form in Eq. (4.2),

$$\begin{pmatrix} \sigma_{x,x} \\ \sigma_{y,y} \\ \sigma_{z,z} \\ \sigma_{z,y} \\ \sigma_{z,x} \\ \sigma_{y,x} \end{pmatrix} = \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{pmatrix} \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ 2\epsilon_4 \\ 2\epsilon_5 \\ 2\epsilon_6 \end{pmatrix}. \quad (4.2)$$

These twenty-one independent components can be further reduced since the protein can be considered transversely isotropic. A material is transversely isotropic if at any point there is an axis of symmetry such that the elastic properties in any direction within a plane perpendicular to the axis are all the same [10]. Each backbone node is connected to two other backbone nodes (excluding end nodes of the chain). Therefore each node will only have two planes about which symmetry occurs. Each plane will be parallel to the vector connecting to the two bonded backbone nodes. This allows for a simplification of several of the components of the elastic modulus,

$$\begin{aligned}
C_{14} &= C_{15} = C_{16} = C_{24} = C_{25} = C_{26} = 0, \\
C_{34} &= C_{35} = C_{36} = C_{45} = C_{46} = C_{56} = 0, \\
C_{11} &= C_{22}, \\
C_{13} &= C_{23}, \\
C_{44} &= C_{55}, \\
C_{66} &= \frac{C_{11} - C_{12}}{2}.
\end{aligned} \tag{4.3}$$

Therefore, the elastic modulus can be greatly simplified to the system of Eqs. (4.4 - 4.9) using the properties in Eq. (4.3) [10],

$$\sigma_1 = C_{11}\epsilon_1 + C_{12}\epsilon_2 + C_{13}\epsilon_3, \tag{4.4}$$

$$\sigma_2 = C_{12}\epsilon_1 + C_{11}\epsilon_2 + C_{13}\epsilon_3, \tag{4.5}$$

$$\sigma_3 = C_{13}\epsilon_1 + C_{13}\epsilon_2 + C_{33}\epsilon_3, \tag{4.6}$$

$$\sigma_4 = 2C_{44}\epsilon_4, \tag{4.7}$$

$$\sigma_5 = 2C_{44}\epsilon_5, \tag{4.8}$$

$$\sigma_6 = (C_{11} - C_{12})\epsilon_6 = 2C_{66}\epsilon_6. \tag{4.9}$$



Unfortunately, these equations are not linearly independent, so more work is needed. The engineering constants will be found first. Instead of computing the elastic modulus directly from the Hooke's Generalized Law using the stress and strain tensors, the compliance matrix,  $S$ , will be computed using the engineering constants of the material,

$$\epsilon_1 = \frac{1}{\mathbf{E}}(\sigma_1 - \nu\sigma_2) - \frac{\nu'}{\mathbf{E}_0}\sigma_3, \quad \epsilon_4 = \frac{1}{2G}\sigma_4, \quad (4.10)$$

$$\epsilon_2 = \frac{1}{\mathbf{E}}(\sigma_2 - \nu\sigma_1) - \frac{\nu'}{\mathbf{E}_0}\sigma_3, \quad \epsilon_5 = \frac{1}{2G}\sigma_5, \quad (4.11)$$

$$\epsilon_3 = -\frac{\nu'}{\mathbf{E}_0}(\sigma_1 + \sigma_2) + \frac{1}{\mathbf{E}_0}\sigma_3, \quad \epsilon_6 = \frac{1}{2G}\sigma_6, \quad (4.12)$$

where  $\mathbf{E}$  and  $\mathbf{E}_0$  are the Young's moduli in the plane of isotropy and in the direction perpendicular to it respectively. The Poisson ratio,  $\nu$ , characterizes transverse contraction in the plane of isotropy due to applied tension in the orthogonal direction.  $\nu'$ , in a similar manner, describes this contraction due to the applied tension perpendicular to the plane of isotropy. This load is also perpendicular to the load applied for  $\nu$ .  $G$  is the shear modulus in the plane of isotropy, and  $G_0$  in any plane perpendicular to the plane of isotropy [14]. Together these engineering constants can be used to compute the compliance matrix  $S$ ,

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ 2\epsilon_4 \\ 2\epsilon_5 \\ 2\epsilon_6 \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & S_{13} & 0 & 0 & 0 \\ S_{12} & S_{11} & S_{13} & 0 & 0 & 0 \\ S_{13} & S_{13} & S_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & S_{66} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix},$$

$$\begin{aligned} S_{11} &= \frac{1}{\mathbf{E}}, & S_{33} &= \frac{1}{\mathbf{E}_0}, & S_{12} &= -\frac{\nu}{\mathbf{E}}, \\ S_{13} &= -\frac{\nu_0}{\mathbf{E}_0}, & S_{44} &= \frac{1}{G_0}, & S_{66} &= \frac{2(1+\nu)}{\mathbf{E}} = \frac{1}{G}. \end{aligned} \quad (4.13)$$

Since the inverse of  $\mathbf{C}$  is the compliance matrix,  $\mathbf{S}$ ,  $\mathbf{C}$  can be readily computed,

$$\begin{aligned}
S_{11} &= \frac{C_{11} - C_{13}^2/C_{33}}{C_{11} - C_{12}} S, & S_{12} &= -\frac{C_{12} - C_{13}^2/C_{33}}{C_{11} - C_{12}} S, \\
S_{13} &= -\frac{C_{13}}{C_{33}} S, & S_{33} &= \frac{C_{11} + C_{12}}{C_{33}} S, \\
S_{44} &= \frac{1}{C_{44}}, & S_{66} &= \frac{1}{C_{66}}, \\
S &= \frac{1}{C_{11} + C_{12} - 2C_{13}^2/C_{33}}. & & (4.14)
\end{aligned}$$

## 4.2 ISOTROPIC SIMPLIFICATION

In order to test the code in computing the elastic modulus, a simplified model was also incorporated. The material (protein) was assumed to be isotropic instead of transversely isotropic. This reduced the number of unknowns from six to three: the Young's modulus ( $\mathbf{E}$ ), Poisson ratio ( $\nu$ ), and Shear modulus ( $G$ ). The Shear modulus can be computed from the Young's modulus and Poisson ratio, reducing the system further to two unknowns [10],

$$G = \frac{\mathbf{E}}{2(1 + \nu)}. \quad (4.15)$$

The Young's modulus and Poisson ratio can be computed directly from the stress and strain tensors [10]:

$$\epsilon_1 = \frac{1}{\mathbf{E}}[\sigma_1 - \nu(\sigma_2 + \sigma_3)], \quad (4.16)$$

$$\epsilon_2 = \frac{1}{\mathbf{E}}[\sigma_2 - \nu(\sigma_3 + \sigma_1)], \quad (4.17)$$

$$\epsilon_3 = \frac{1}{\mathbf{E}}[\sigma_3 - \nu(\sigma_1 + \sigma_2)]. \quad (4.18)$$

## CHAPTER 5

### COMPUTATION SIMULATION USING TOY MODEL PROTEINS

#### 5.1 VERLET INTEGRATION OF MOLECULAR DYNAMIC SIMULATION

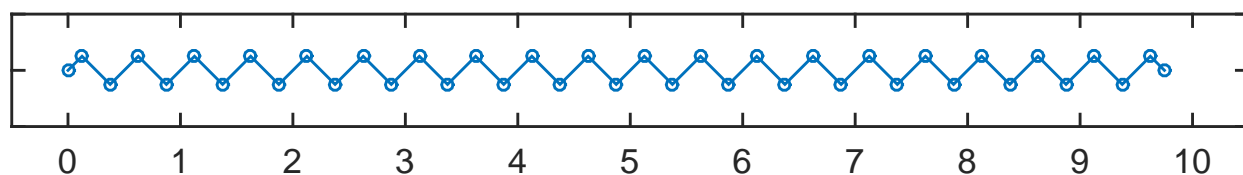


Figure 5.1: Picture of the spring with forty-one nodes in equilibrium. The units of distance is in nanometers.

Two toy model molecular dynamic simulations with constant velocity pulling were created in MATLAB. The elastic modulus was known to be uniform across the entire structure in order which allowed the testing of the validity of this procedure. The first model consisted of a two dimensional forty-one node system as seen in Figure 5.1. The second model contained eleven nodes in one dimension. Each node is connected to two neighboring nodes by a virtual “spring”. Initially, the two dimensional simulation was computed using Verlet velocity. The

$x$  and  $y$  components of this integration scheme is seen in Eqs. (5.1 - 5.6) [21] are,

$$a_{n+1}^x = \frac{F_{n+1}^x}{m}, \quad (5.1)$$

$$a_{n+1}^y = \frac{F_{n+1}^y}{m}, \quad (5.2)$$

$$x_{n+1} = x_n + v_n^x \Delta t + a_n^x (\Delta t)^2, \quad (5.3)$$

$$y_{n+1} = y_n + v_n^y \Delta t + a_n^y (\Delta t)^2, \quad (5.4)$$

$$v_{n+1}^x = v_n^x + \frac{1}{2}(a_n^x + a_{n+1}^x) \Delta t, \quad (5.5)$$

$$v_{n+1}^y = v_n^y + \frac{1}{2}(a_n^y + a_{n+1}^y) \Delta t, \quad (5.6)$$

where  $a$  is the acceleration,  $v$  is the velocity,  $m$  is the mass, and  $x, y$  are the  $x$  and  $y$  positions of each node. The force,  $F$ , is computed from only the bond and angle force in this simplified model.  $n$  is the current time-step.

Unfortunately, this numerical method was not stable with our parameters. However, the second attempt using Verlet integration was successful. The integration scheme is seen in Eq. (5.7) and Eq. (5.8) for each dimension.

$$x_{n+1} = 2x_n - x_{n-1} + a_n^x (\Delta t)^2, \quad (5.7)$$

$$y_{n+1} = 2y_n - y_{n-1} + a_n^y (\Delta t)^2. \quad (5.8)$$

The one dimensional Verlet scheme can be directly seen from its two dimensional counterpart by only including the  $x$  equations.

This scheme is a fourth-order accurate in position. The derivation of the Verlet algorithm using a Taylor series expansion can be used to show this. The acceleration  $a(t)$  will be

computed using Newton's second law in Eq. (5.1),

$$\begin{aligned} x(t + \Delta t) &= x(t) + \Delta t x'(t) + \frac{\Delta t^2}{2} x''(t) + \frac{\Delta t^3}{6} x'''(t) + \mathcal{O}(\Delta t^4), \\ &= x(t) + \Delta t v(t) + \frac{\Delta t^2}{2} a(t) + \frac{\Delta t^3}{6} x'''(t) + \mathcal{O}(\Delta t^4), \end{aligned} \quad (5.9)$$

$$x(t - \Delta t) = x(t) - \Delta t v(t) + \frac{\Delta t^2}{2} a(t) - \frac{\Delta t^3}{6} x'''(t) + \mathcal{O}(\Delta t^4). \quad (5.10)$$

Adding together Eq. (5.9) and Eq. (5.10) cancels some of the terms producing our Verlet integration scheme,

$$x(t + \Delta t) + x(t - \Delta t) = 2x(t) + \Delta t^2 a(t) + \mathcal{O}(\Delta t^4). \quad (5.11)$$

## 5.2 ONE DIMENSIONAL TOY MODEL PROTEIN

A simple one dimensional toy model was created. Here there are eleven nodes as seen in Figure 5.2. The first node is fixed, while the last node is pulled at a constant velocity of 0.02 nm/fs. The simulation is run for 50,000 fs meaning the last node is pulled 1 nanometers producing twenty percent strain.

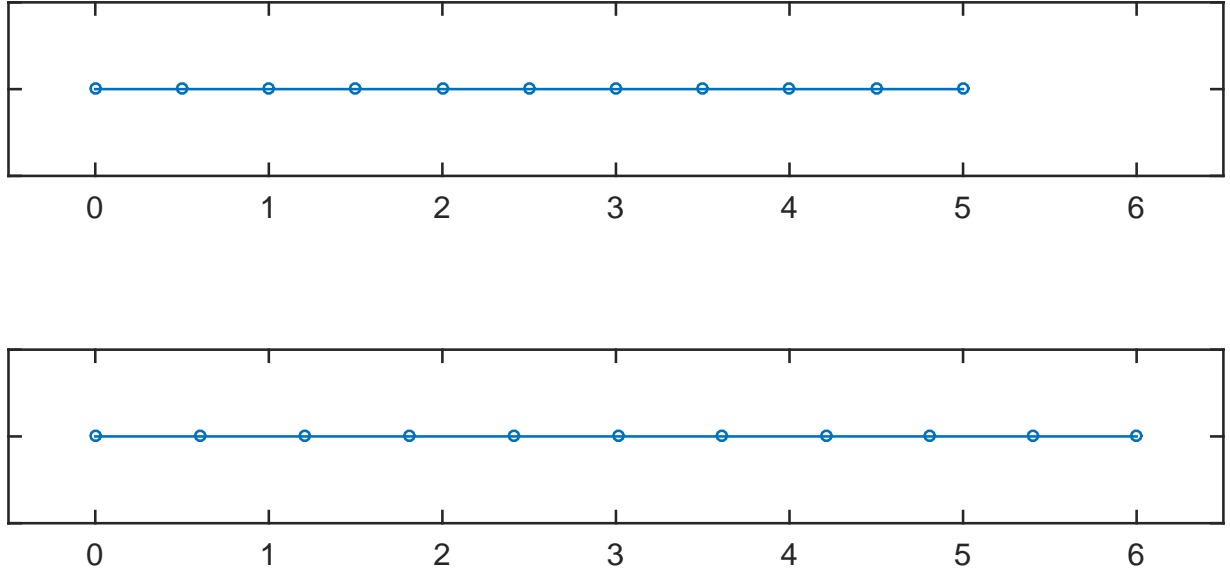


Figure 5.2: Top: One dimensional spring in equilibrium. Bottom: One dimensional spring after simulation. The units are nanometers

The theoretical Young's Modulus,  $\mathbf{E}$ , can be computed from the spring constant, which was given as  $1250 \text{ kJ}/(\text{nm}^2 \text{ mol})$ ,

$$\begin{aligned}
 \mathbf{E} &= k \frac{r_0}{A_0} = k \frac{r_0}{\frac{V_0}{r_0}} = k \frac{r_0^2}{V_0} = 1250 \frac{\text{kJ}}{\text{nm}^2 \text{ mol}} \frac{.5 \text{nm}^2}{\frac{128 \times 10^{-3} \text{m}^3}{1170 \text{mol}}} \\
 &= \frac{1250 \cdot (.5)^2 \cdot 1170 \cdot 10^3}{128} \frac{\text{kJ}}{\text{m}^3} = 2856445.3125 \times 10^3 \frac{\text{J}}{\text{m}^3} \\
 &= 2856445312.5 \text{ Pa} = 2.8564453125 \text{ GPa}.
 \end{aligned} \tag{5.12}$$

This is in great agreement with the Young's Modulus computed from the simulation as seen in Figure 5.3. In one dimension, the Young's Modulus can be found by the stress tensor divided by the strain tensor,

$$\mathbf{E} = \frac{\sigma}{\epsilon}. \tag{5.13}$$

The stress tensor is computed as in Eq. (3.37), however the strain tensor is computed using

the engineering strain rather than the Green-Lagrangian strain in Eq. (2.6).

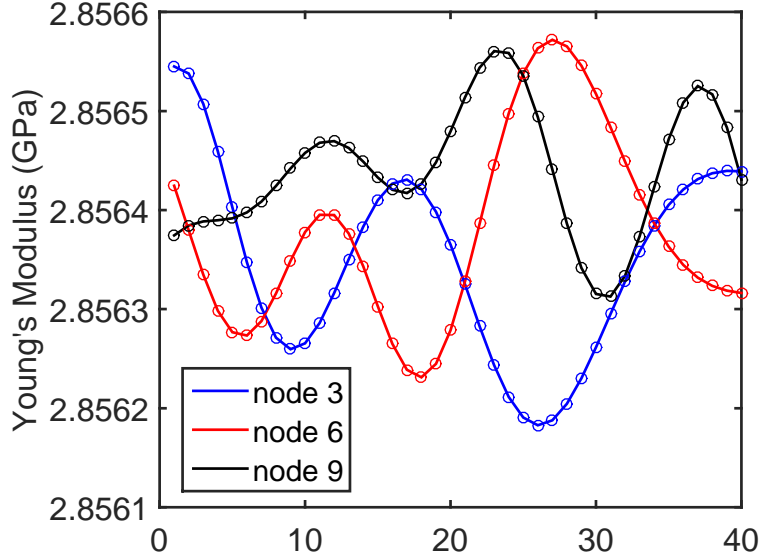


Figure 5.3: Computed Young's Modulus of the one dimensional system. The  $y$ -axis is the Young's Modulus with units of GPa. Each point on the  $x$ -axis is the average Young's Modulus of 100 time-steps over the last 4,000 fs of the simulation. Each time-step is a femtosecond.

To further check the validity of this toy model, the energy added to the system was computed directly from the definition of the bond energy, and by integrating the power of the last node. Due to the small constant velocity at which the system is pulled, it is essentially in equilibrium at every time-step. This means that the force at each node is approximately zero at every node other than the end nodes.

In one dimension, the bond energy is given by Eq. (5.14),

$$E = \frac{1}{2}k_b(r - r_0)^2, . \quad (5.14)$$

This is compared by integrating the power at each time-step  $t$  as seen in Eq. (5.15),

$$E = \int_0^t Fv \, dt. \quad (5.15)$$

In the numerical simulation this can be approximated with the sum in Eq. (5.16),

$$E = \sum_i F_i v. \quad (5.16)$$

The results are in very good agreement.

### 5.3 ELASTIC MODULUS IN TWO DIMENSIONS

The stress and strain tensors are computed in a similar manner to the earlier computations. However, since our model is two dimensional instead of three dimensional, some computations can be simplified. Both the stress and strain tensor now have 4 components instead of 9, which reduces to 3 independent components due to symmetry. Furthermore, the system is isotropic instead of anisotropic. So only the Young's Modulus,  $E$ , Poisson Ratio,  $v$ , and Shear Modulus,  $G$ , need be computed,

$$\epsilon_1 = \frac{1}{\mathbf{E}}(\sigma_1 - v\sigma_2), \quad (5.17)$$

$$\epsilon_2 = \frac{1}{\mathbf{E}}(\sigma_2 - v\sigma_1), \quad (5.18)$$

$$\mathbf{E} = \frac{\sigma_2 - v\sigma_1}{\epsilon_2}, \quad (5.19)$$

$$\epsilon_1 \frac{\sigma_2 - v\sigma_1}{\epsilon_2} = (\sigma_1 - v\sigma_2), \quad (5.20)$$

$$\epsilon_1 \sigma_2 - \epsilon_1 \sigma_1 v = \epsilon_2 \sigma_1 - \epsilon_2 \sigma_2 v, \quad (5.21)$$

$$\epsilon_1 \sigma_2 - \epsilon_2 \sigma_1 = \epsilon_1 \sigma_1 v - \epsilon_2 \sigma_2 v, \quad (5.22)$$

$$v = \frac{\epsilon_1 \sigma_2 - \epsilon_2 \sigma_1}{\epsilon_1 \sigma_1 - \epsilon_2 \sigma_2}. \quad (5.23)$$

Within the computations of strain tensor Eqs. (2.10), (2.11), (2.13), and (2.14) were used. However, due to the symmetry of the strain tensor Eqs. (2.11) and (2.14) were averaged together to form a three dimensional system of equations. There was no significant difference between this approach and Least Squares Method. However, this system of equations



produced an ill-conditioned matrix. This was due to the fact that the bond vector between the nodes are very close to being linearly dependent where the only significant difference is the sign of the y-component. In order to address this issue Eq. (2.11) and Eq. (2.13) were changed,

$$(r_{i,i+2})^2 - (r_{0,i,i+2})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i,i+2}^\alpha)(\mathbf{r}_{0,i,i+2}^\beta), \quad (5.24)$$

$$(r_{i-2,i})^2 - (r_{0,i-2,i})^2 = \sum_{\alpha,\beta} 2\epsilon^{\alpha,\beta}(\mathbf{r}_{0,i-2,i}^\alpha)(\mathbf{r}_{0,i-2,i}^\beta). \quad (5.25)$$

The bond vectors y-component is approximately zero, making them linearly independent from equations (2.10) and (2.13).

#### 5.4 TWO-DIMENSIONAL TOY MODEL

The simulation is run for 50,000 fs meaning that the last node is pulled 1 nm producing approximately ten percent strain. The mass of each node is roughly the mass of Lysine, 0.128 kg/mol. The bond constant and angle constant were given to be  $k_b = 1250$  kJ/(nm<sup>2</sup>mol), and  $k_\theta = 700$  kJ/(nm<sup>2</sup>mol) respectively, representing a helical secondary structure in the MARTINI Force Field. From Figure 5.4 it is seen that the Young's Modulus converges nicely to a value of approximately 1.3 GPa. This is representing a protein of Lysine with helical secondary structure. This is on a similar order for the Young's Modulus calculated for F-actin which is 2.7 GPa [2].

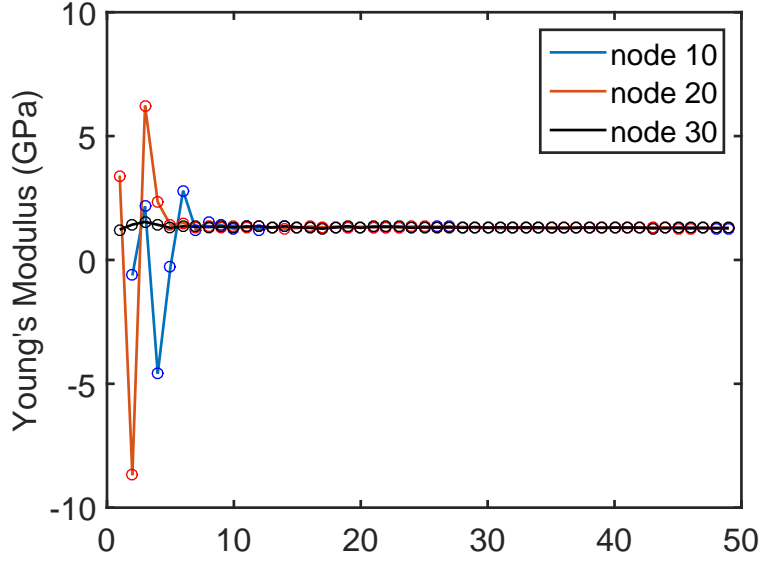


Figure 5.4: Computed Young's Modulus of the two dimensional system. The  $y$ -axis is the Young's Modulus with units of GPa. Each point on the  $x$ -axis is the average Young's Modulus of 1000 time-steps over 50,000 femtoseconds of simulation. Each time-step is a femtosecond.

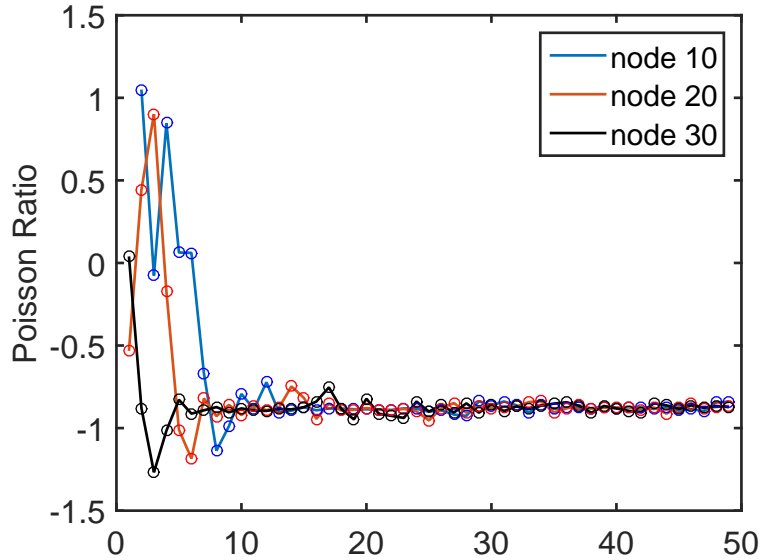


Figure 5.5: Poisson Ratio of the two dimensional system. The  $y$ -axis is the Poisson Ratio which is unitless. Each point on the  $x$ -axis is the average Poisson Ratio of 1000 time-steps over 50,000 femtoseconds of simulation. Each time-step is a femtosecond.

Unfortunately, the Poisson ratio was not as well behaved. When pulling a spring, the springs length should shrink in the the direction orthogonal to the load. However, as seen

in Figure 5.6, this is not the case. The magnitude of the bond lengths in the orthogonal direction are increasing. This is the cause of for the Poisson ratio to be negative as seen in Figure 5.5. The origin of this unrealistic behavior is the MARTINI Force Field. Such a large angle constant is preventing the spring from shrinking in the  $y$  direction. If this angle constant is lowered to 10 kJ/(nm<sup>2</sup>mol), a more realistic Poisson ratio of approximately 0.6 is observed for this model.

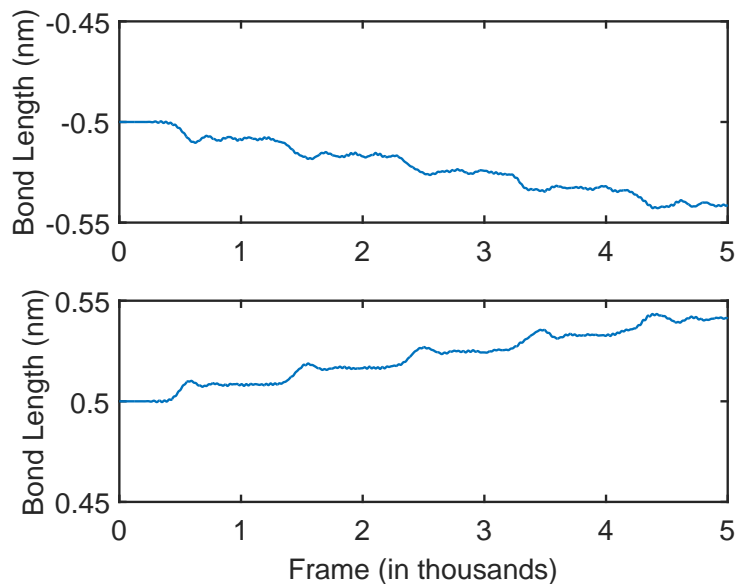


Figure 5.6: Bond length of the two bonds of the node tested within the spring.

## CHAPTER 6

### COMPUTATION SIMULATION WITH PROTEINS

#### 6.1 TOY PROTEIN WITH ALL BACKBONE NODES

A toy protein was created to further test the validity of code. The protein with the PDB ID of 1MV4 had its side-chain nodes removed, and each backbone node was considered Glycine. This created 37 nodes in all. The first node was fixed while the last node was pulled at a constant velocity of  $3.75\text{e-}5$  angstroms per time step where each time-step is two femtoseconds. The pulling was done for 1000 time-steps, followed by 1000 time-steps of minimization. This was iterated 400 times.

Frames ten through three hundred of the simulation were used for these results. The shear modulus was computed directly from the stress and strain tensors in Eqs. (4.10 - 4.12). Twenty frames were used within a least squares approach to solve the overdetermined system in order to compute the Young's modulus and Poisson ratio with the assumption that the material is isotropic. The results that follow are the median of the solutions all of the solutions of the overdetermined systems.

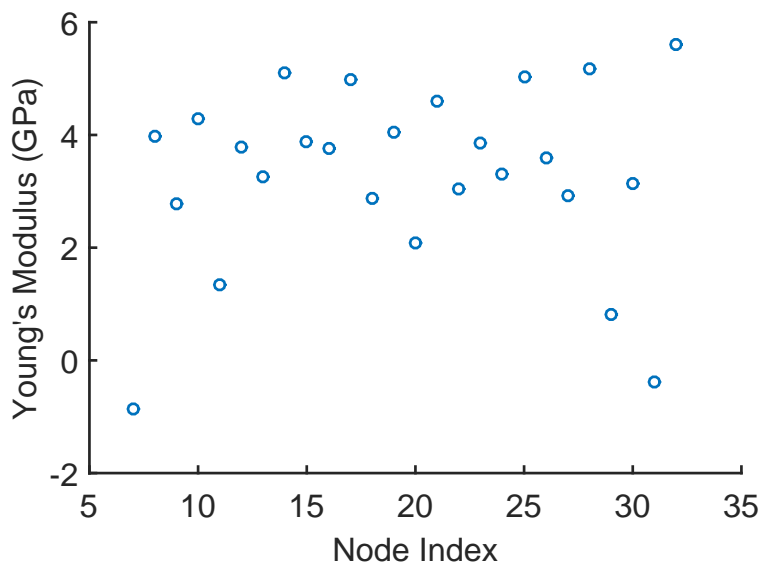


Figure 6.1: Computed Young's Moduli of the toy alpha helix comprising of only Glycine backbone nodes. The  $y$ -axis is the Young's Modulus with units of GPa.

The data is clustered quite nicely except for a few nodes near the ends of the alpha helix. Since our approach takes some averaging of nearby nodes, specifically in the strain tensor computation, nodes near the ends will have inaccurate results. Therefore, the two nodes with negative Young's Modulus can be removed as outliers. The mean of the remaining data results in a value of 3.63 GPa, while the median is 3.77 GPa.

Unfortunately, the Poisson ratio results proved to be poor. More than half of the results were negative. A negative Poisson ratio is described physically by an expansion of the material in the direction orthogonal to the direction being pulled. Since the alpha helix was translated to be parallel with the  $x$ -axis and pulled only in the  $x$ -direction, the orthogonal directions would be in the  $y$  and  $z$  direction. This is precisely what is seen in the simulation. An example is shown in Figure 6.3.

However, this is unrealistic for a spring like structure such as our alpha helix. It would be expected that the Poisson ratio would be positive, and on the order of 0.5. An attempt of reducing the angle bond constant,  $k_\theta$ , as was done with the two dimensional toy model, within the NAMD simulation was made. While the longterm behavior was a decrease of

the bond lengths of the backbone nodes in the y and z direction, initially they increased in magnitude causing a poor Poisson ratio once again. This shortcoming of the MARTINI Force-Field makes our approach to compute the Poisson ratio, and Shear modulus impossible at this juncture. If the material is considered to be anisotropic instead of isotropic, the results of the elastic moduli are similarly poor to the Poisson ratio.

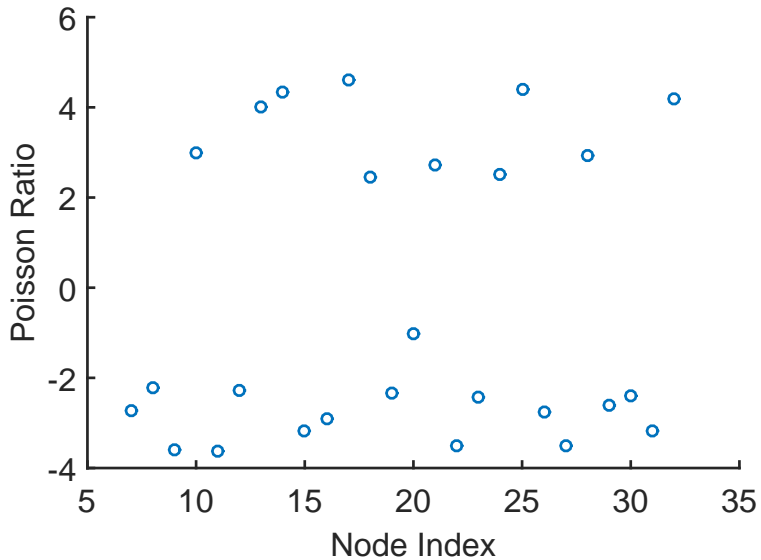


Figure 6.2: Computed Poisson ratio of the fictitious alpha helix comprising of only Glycine backbone nodes.

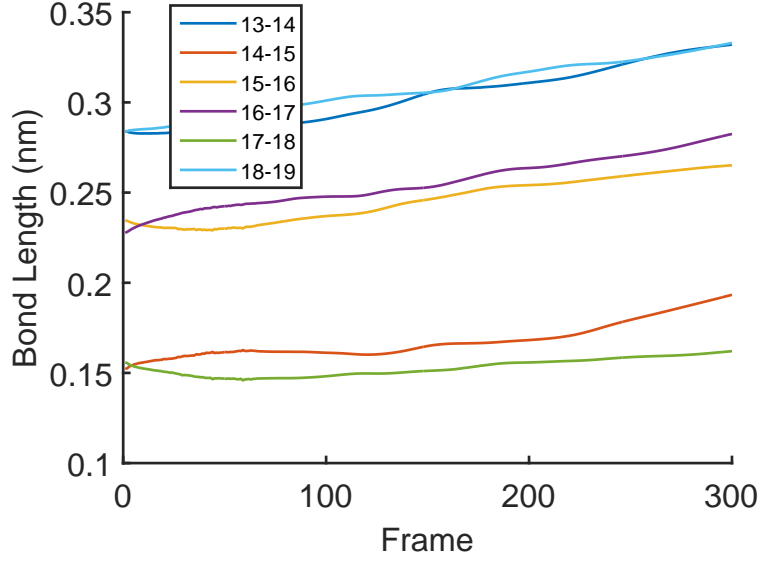


Figure 6.3:  $y$ -component of the bond length of several interior nodes of the fictitious toy alpha helix comprising of only Glycine backbone nodes. Each point on the  $x$ -axis is a frame within the simulation.

## 6.2 ALPHA HELICES

Twenty alpha helix proteins were selected. Some included several alpha helices within their structure. In this case only one alpha helix was used within the simulation, while the remaining ones were deleted. The helices were considered isotropic material and only the Young's moduli is reported. The other engineering constants and the elastic moduli for transversely isotropic material were all poor due to unrealistic simulations as described in the previous section.

The presented data in Figures 6.4 - 6.5 is truncated, only considering computed Young's Modulus between 0 and 10 GPa. Negative Young's Modulus is obviously unrealistic, and 10+ GPa is a magnitude larger than the reported Young's Modulus of 2.7 GPa [2].

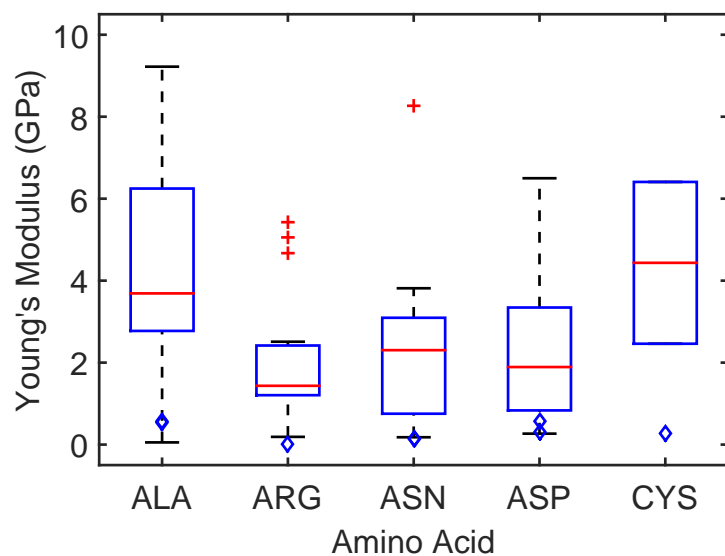


Figure 6.4: Young's moduli of amino acids within a tolerance of 0 to 10. Unit is GPa.

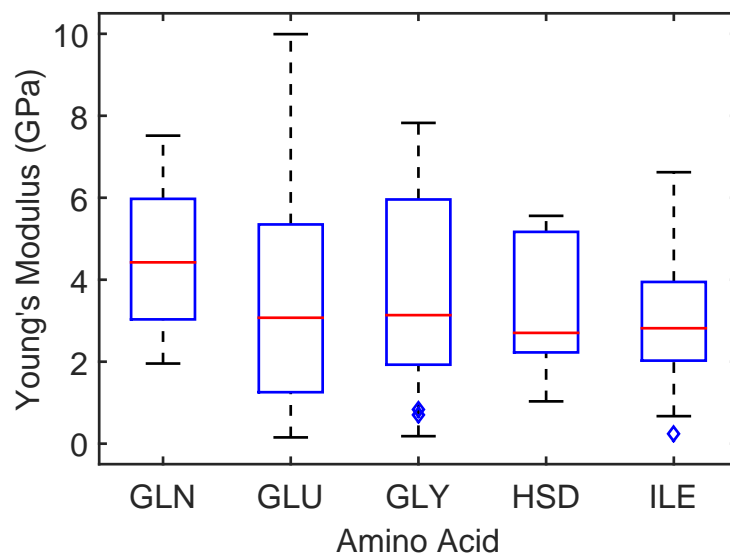


Figure 6.5: Young's moduli of amino acids within a tolerance of 0 to 10. Unit is GPa.



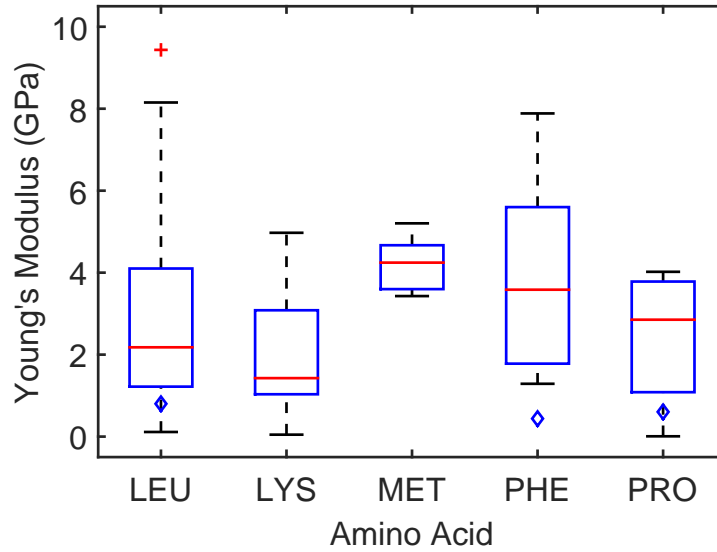


Figure 6.6: Young's moduli of amino acids within a tolerance of 0 to 10. Unit is GPa.

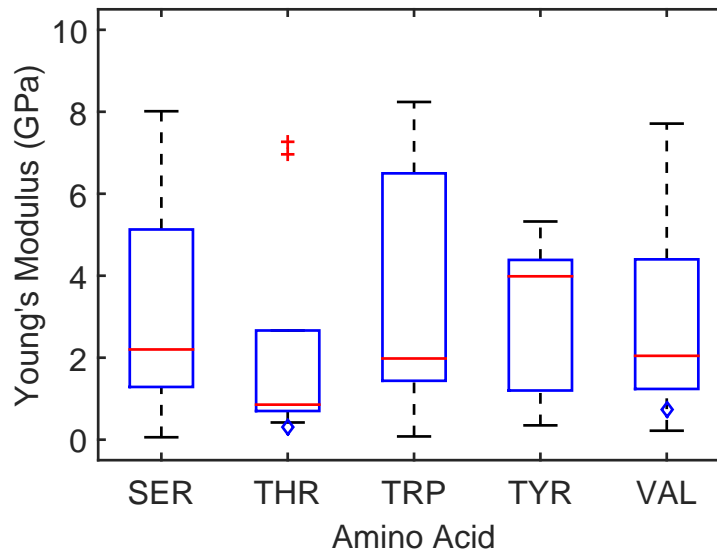


Figure 6.7: Young's moduli of amino acids within a tolerance of 0 to 10. Unit is GPa.

As seen in Figures 6.4 - 6.7, the median of the truncated data for the Young's Modulus for each amino acid lies within a range of 1 to 5 GPa. This is of the same order as F-actin which is 2.7 GPa [2]. The amino acids with small sample size within the truncated area (5 or less samples) are CYS, MET and PRO. Pro has very nice results while CYS and MET have slightly larger Young's Modulus than expected, but still reasonable.

### 6.3 PROTEIN 1B6Q

Alpha helices are well behaved compared to other secondary protein structures due to their spring like nature. However, our model must be tested on other secondary structures which proved to be difficult. The protein 1B6Q was chosen since it consisted of two alpha helices connected by a small coil, two amino acids long. An all atom simulation was performed as described in Section 3.5. A steered molecular dynamic simulation was attempted in a similar manner to Section 3.5 however the alpha helices were attracted to each other producing an unrealistic result. In a protein's natural environment, it is contained in things like membranes or solutions. In this state the protein's conformation is similar to the one depicted in Figure 6.8. However, in a vacuum, as in the simulation, the alpha helices are attracted to each other causing the protein to buckle upon itself.

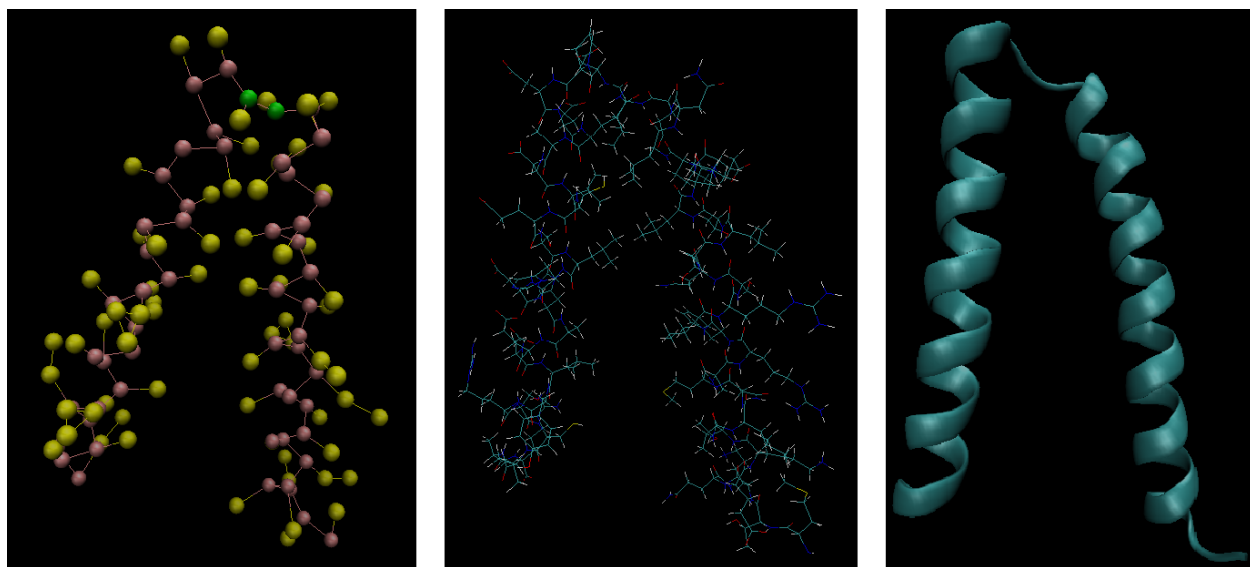


Figure 6.8: 1B6Q consists of two alpha helices connected by a coil in the middle of the protein. Left: Coarse grained 1B6Q. The coil backbone nodes are indicated in green. Pink indicates backbone nodes, and yellow indicates side-chain nodes. Middle: Atomic resolution of 1B6Q displayed in line method. Right: Atomic resolution of 1B6Q displayed in new cartoon method. These images were created using VMD.

To overcome this obstacle, many nodes were fixed. Nodes 1-35, 51, and 90-120 were fixed. 1-35 and 90-120 are the tail ends of each alpha helix. Node 51 lies opposite of the pulled

Table 6.1: Young’s Moduli of coil and nearby alpha helix nodes of 1B6Q.

Backbone Node	Young’s Modulus
58	+7.7331e-01
60	+3.1726e-01
62 (coil)	+2.0544e-01
64 (coil)	-3.8041e-01
66	+1.1907e-01
68	+3.9367e-01

node, 75, to prevent the helices near the coil from sticking together. The bond constant,  $k_b$ , and angle constant  $k_\theta$  are much larger for the alpha helix compared to the coil [16]. Therefore, the expected Young’s modulus of the coil should be smaller than the previously computed Young’s modulus of the alpha helix.

The two coil backbone nodes 62 has a positive Young’s Modulus which is at least an order less than that which was computed for the alpha helices. This is realistic since the bond constants for alpha helices are much larger than that of the coils. For example, the bond constant between two backbone nodes that are within a alpha helix is 1250 kJ/nm<sup>2</sup>mol, while coils are 200 kJ/nm<sup>2</sup>mol [16]. Unfortunately node 64 has a negative Young’s Modulus which can be considered an outlier, but it is on a similar order. The nearby backbone nodes also have a smaller Young’s Modulus despite being alpha helices. This is due to the fact that the weaker force constant is used for the backbone bonded parameter when there is more than one type of secondary structure present. Furthermore, the strain tensor uses an average among nearby backbone bonds. It is for this reason that these smaller magnitude Young’s Modulus for the alpha helix backbone nodes is to be expected.

## 6.4 COILS

Ten coils were selected for the application of the method. A few initial proteins were chosen which had longer coils. Unfortunately, these proteins gave poor results with Young’s moduli over 10 GPa. One possibility is the fact that these “long” coils were still short

Table 6.2: The Computed Young's Moduli of All Coils Examined.

Amino Acid	Mean	Median	Max	Min	Sample Size
ALA	+0.5546	+0.5546	+0.5555	+0.5536	2
ARG	-0.0283	-0.0283	+0.0002	-0.0569	2
ASN	+0.1327	+0.1327	+0.1373	+0.1281	2
ASP	+0.4346	+0.4346	+0.5598	+0.3094	2
CYS	+0.2587	+0.2587	+0.2587	+0.2587	1
GLY	+0.7645	+0.7645	+0.8319	+0.6972	2
ILE	-0.1476	-0.1476	+0.2555	-0.5508	2
LEU	+0.7896	+0.7896	+0.7896	+0.7896	1
PHE	+0.1384	+0.1384	+0.4466	-0.1699	2
PRO	+0.0202	+0.0202	+0.6106	-0.5703	2
SER	+1.3046	+1.3046	+2.6416	-0.0323	2
THR	+0.0729	+0.0729	+0.3135	-0.1677	2
TYR	-0.4533	-0.4533	-0.4533	-0.4533	1
VAL	+0.7514	+0.7514	+0.7514	+0.7514	1

compared to the smaller alpha helices. All alpha helices contained at least twenty amino acids, while these coils contained roughly ten amino acids. It was observed that amino acids nearby the fixed node, and nearby the pulled node gave worse results than those closer to the middle of the section tested.

It was for this reason that small coils in between two alpha helices were chosen. This is realistic due to the fact that the weaker bond constant, in this case the coil, is used for bonded interactions of mixed secondary structure. Through strategic fixing of nodes as done with 1B6Q, all 10 proteins were tested with fairly reasonable results.

Despite the small sample size, most of the Young's Moduli computed were within the range of 0 to 1.5 GPa. This is of smaller order than the alpha helix, which is expected. From this small sample size, a good estimate for the Young's Modulus of each amino acid is difficult to determine, but these results do provide evidence toward the validity to the method.

Table 6.3: The Computed Young's Moduli of Coils that are within 0 and 1.5 GPa

Amino Acid	Mean	Median	Max	Min	Sample Size
ALA	+0.5546	+0.5546	+0.5555	+0.5536	2
ARG	+0.0002	+0.0002	+0.0002	+0.0002	1
ASN	+0.1327	+0.1327	+0.1373	+0.1281	2
ASP	+0.4346	+0.4346	+0.5598	+0.3094	2
CYS	+0.2587	+0.2587	+0.2587	+0.2587	1
GLY	+0.7645	+0.7645	+0.8319	+0.6972	2
ILE	+0.2555	+0.2555	+0.2555	+0.2555	1
LEU	+0.7896	+0.7896	+0.7896	+0.7896	1
PHE	+0.4466	+0.4466	+0.4466	+0.4466	1
PRO	+0.6106	+0.6106	+0.6106	+0.6106	1
SER	NaN	NaN			0
THR	+0.3135	+0.3135	+0.3135	+0.3135	1
TYR	NaN	NaN			0
VAL	+0.7514	+0.7514	+0.7514	+0.7514	1

## CHAPTER 7

### CONCLUDING REMARKS

#### 7.1 CONCLUSION

This thesis begins by presenting the motivation for the current work describing how a protein's deformations play an important role in their function, along with some biological background. Derivations for both the stress and strain tensors are shown, as they play a key role in finding a protein's elastic properties. There is supportive evidence for the validity of the method when the protein is considered isotropic. This includes a one and two dimensional spring that were both tested in MATLAB. These springs were comprised of only Lysine backbone nodes giving the toy model a simplified structure. Both springs provided well behaved results. The anisotropic elastic properties were not obtained due to unrealistic simulations.

Furthermore, the alpha helix protein 1MV4 had its side chains removed to help minimize noise. Again, the results were well behaved for the Young's Modulus when the protein is considered isotropic. An unrealistic Poisson Ratio was found due the fact that the radius of the alpha helix was increasing when being pulled rather than decreasing. Due to this data, only the Young's Modulus could be found at this time. A large sample size was used to estimate the Young's Modulus of most of the alpha helix amino acids. Due to of difficulties when testing coils, only a small sample size was used to test the validity of the method. Overall, there is a large body of evidence showing that the implementation of the method will provide successful results.

## 7.2 FUTURE WORK

The obvious next step in the research is to apply this approach in computing the Young's Modulus for more proteins. The goal is to have a database of the Young's Modulus of each amino acid for each secondary structure. Most of the amino acids with alpha helix structures have a large enough sample size, however, there were a few where this was not the case. For example, the amino acid CYS only had one sample, and it was considered an outlier. While enough data was collected to show the method's success in computing the Young's Modulus of coil amino acids, the sample sizes are too small to provide good estimations. Furthermore, there are other secondary structures; extended, turn, and bend, which were not investigated as of yet.

The initial goal of modeling the deformation of proteins that are considered anisotropic was not accomplished. In future work the obstacle of unrealistic simulations will attempted to be overcome. Once the elastic moduli of each amino acid are found, the first step in turning the protein from a discrete system into a continuum is defining the volume. Through the program, MSMS [20], a solvent-excluded surface (SES) is used to represent the protein. This surface is found by first letting each node be considered a sphere of some radius. This is called the Van der Waals surface. A probe sphere is rolled around this Van der Waals surface. Where the probe contacts this surface and the reentrant surface will be considered the solvent-excluded surface. The reentrant surface is defined as a series of patches defined by the interior-facing part of the probe when it is simultaneously in contact with more than one node [19].

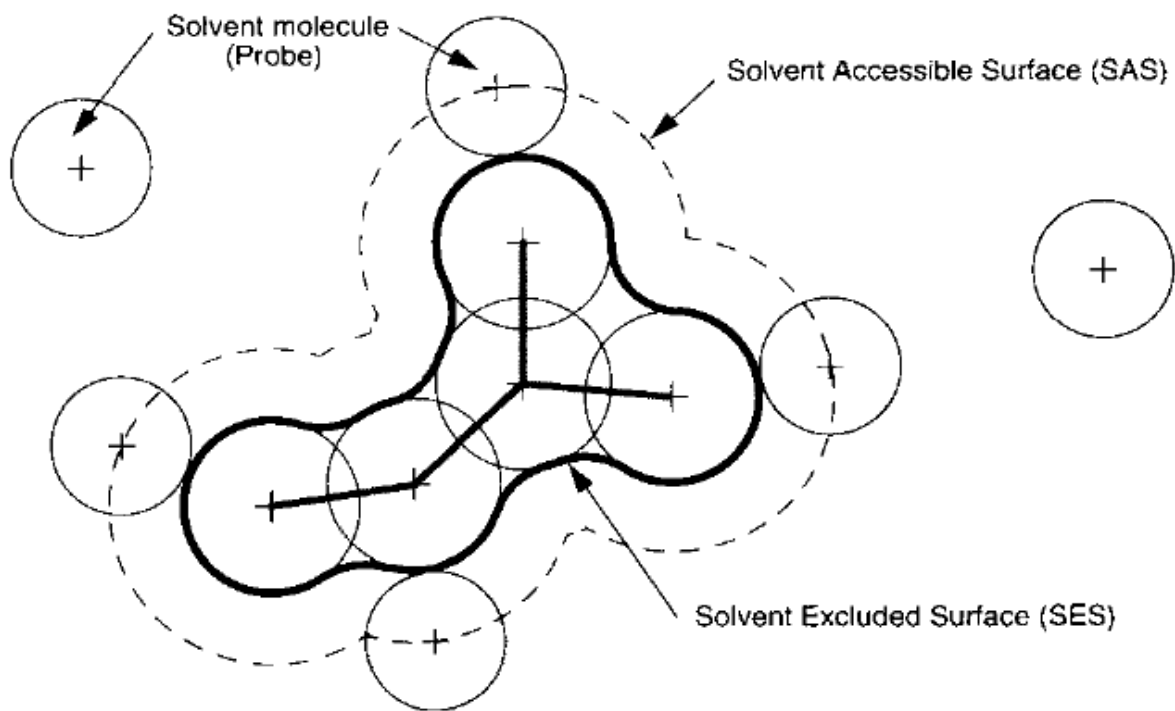


Figure 7.1: Probe being rolled over a van der Waals surface [20]



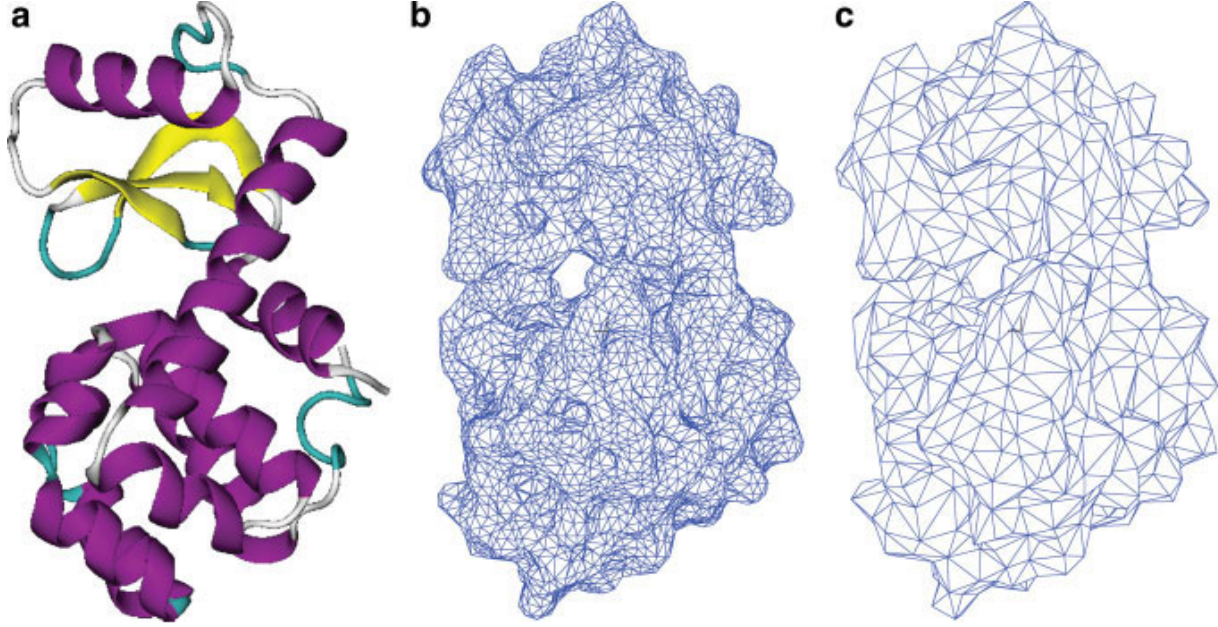


Figure 7.2: T4 lysozyme. Left: Schematic representation of the energy-minimized molecular structure analyzed. Middle: Original high-resolution triangulated discretization of the SES. Right: Simplified discretization of the SES used to define the FE model. Molecular structure is rendered using VMD version 1.8.5 and surface mesh models are rendered using ADINA version 8.4 [2]

Once a surface is defined, the elastic moduli along with a Finite Element Method [13], will be used to find the properties of a protein in a continuous sense. The elements within the FEM scheme will be tetrahedrals. The FEM uses a numerical volum-integration to derive a system of equations that is linear with respect to the displacement  $\mathbf{d}$  [2],

$$\mathbf{M}\mathbf{d} + \mathbf{K}\mathbf{d} = \mathbf{r} \quad (7.1)$$

where  $\mathbf{K}$  is the stiffness matrix,  $\mathbf{M}$  is the mass matrix, and  $\mathbf{r}$  is the forcing vector resulting from (force-based) boundary conditions [2]. When there is no external force,  $\mathbf{r} = 0$ , this will produce the generalized eigenvalue problem,

$$\mathbf{K}\mathbf{y} = \lambda\mathbf{M}\mathbf{y}, \quad (7.2)$$

where  $\mathbf{y}$  and  $\boldsymbol{\lambda}$  are found from substituting the oscillatory solution for the displacement  $\mathbf{d}$ ,

$$\mathbf{d} = \mathbf{y} \cos(\omega t + \gamma). \quad (7.3)$$

The solutions to Eq. (7.3), gives the eigenmodes of the protein. Both the solutions to Eqs. (7.1) and (7.3) will describe the natural deformations of the protein, and the final goal of the next step for the research. These deformations will provide insight into a proteins functions.

## BIBLIOGRAPHY

- [1] Nikhil Chandra Admal and E.B. Tadmor. A unified interpretation of stress in molecular systems. *Journal of Elasticity*, 100(1-2):63–143, 2010.
- [2] Mark Bathe. A finite element framework for computation of protein normal modes and mechanical response. *Proteins: Structure, Function, and Bioinformatics*, 70(4):1595–1609, 2008.
- [3] Markus J. Buehler. Atomistic and continuum modeling of mechanical properties of collagen: Elasticity, fracture, and self-assembly. *Journal of Materials Research*, 21(8):1947–1961, 008 2006.
- [4] R. Clausius. Xvi. on a mechanical theorem applicable to heat. *Philosophical Magazine Series 4*, 40(265):122–127, 1870.
- [5] Yi Fang, Yu-Shen Liu, and Karthik Ramani. Atomistic and continuum modeling of mechanical properties of collagen: Elasticity, fracture, and self-assembly. <https://engineering.purdue.edu/PRECISE/LDD>.
- [6] J. Clerk Maxwell F.R.S. Xlv. on reciprocal figures and diagrams of forces. *Philosophical Magazine Series 4*, 27(182):250–261, 1864.
- [7] Michael Griebel and Jan Hamaekers. Molecular dynamics simulations of the elastic moduli of polymercarbon nanotube composites. *Computer Methods in Applied Mechanics and Engineering*, 193(1720):1773 – 1788, 2004.

- [8] P M Gullett, M F Horstemeyer, M I Baskes, and H Fang. A deformation gradient tensor and strain tensors for atomistic simulations. *Modelling and Simulation in Materials Science and Engineering*, 16(1):015001, 2008.
- [9] J.C. Hansen, R. Skalak, S. Chien, and A. Hoger. An elastic network model based on the structure of the red blood cell membrane skeleton. *Biophysical Journal*, 70(1):146 – 166, 1996.
- [10] L. Zhang Haojiang Ding, Weiqiu Chen. Basic equations of anisotropic elasticity. In *Elasticity of Transversely Isotropic Materials*, volume 126 of *Solid Mechanics and Its Applications*, pages 1–27. Springer Netherlands, 2006.
- [11] Robert J. Hardy. Formulas for determining local properties in molecular dynamics simulations: Shock waves. *The Journal of Chemical Physics*, 76(1):622–628, 1982.
- [12] J. H. Irving and John G. Kirkwood. The statistical mechanical theory of transport processes. iv. the equations of hydrodynamics. *The Journal of Chemical Physics*, 18(6):817–829, 1950.
- [13] Daryl L. Logan. *A First Course in the Finite Element Method Using Algor*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2nd edition, 2000.
- [14] V. A. Lubarda and M. C. Chen. On the elastic moduli and compliances of transversely isotropic and orthotropic materials. *Journal of Mechanics of Material and Structures*, 3(1):153–171, 2008.
- [15] Joel David Marquez. Elastic network & finite element model to study actin protein mechanics & its molecular elasticity. 2010.
- [16] Luca Monticelli, Senthil K. Kandasamy, Xavier Periole, Ronald G. Larson, D. Peter Tieleman, and Siewert-Jan Marrink. The martini coarse-grained force field: Extension to proteins. *Journal of Chemical Theory and Computation*, 4(5):819–834, 2008.

- [17] A. I. Murdoch and D. Bedeaux. Continuum equations of balance via weighted averages of microscopic quantities. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 445(1923):157–179, 1994.
- [18] Walter Noll. Thoughts on the concept of stress. *Journal of Elasticity*, 100(1-2):25–32, 2010.
- [19] F M Richards. Areas, Volumes, Packing, and Protein Structure. *Annual Review of Biophysics and Bioengineering*, 6:151–176, 1977.
- [20] Michel F. Sanner, Arthur J. Olson, and Jean-Claude Spehner. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996.
- [21] Tamar Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer Publishing Company, Incorporated, 2nd edition, 2010.
- [22] Jerry Zhijian Yang, Xiaojie Wu, and Xiantao Li. A generalized irvingkirkwood formula for the calculation of stress in molecular dynamics models. *The Journal of Chemical Physics*, 137(13), 2012.
- [23] J. Zhijian Yang and S. Du. Calculation of Cauchy stress tensor in molecular dynamics system with a generalized Irving-Kirkwood formulism. *ArXiv e-prints*, November 2014.