MULTISCALE CONTINUUM MODELING OF PROTEIN DYNAMICS

A Thesis
Presented to
The Academic Faculty

by

Kyle N. Karlson

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology
May 2012

MULTISCALE CONTINUUM MODELING OF PROTEIN DYNAMICS

Approved by:

Dr. Michael J. Leamy, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Aldo Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Harvey Lipkin
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: 26 March 2012

Dedicated to my father, Neil Karlson.

ACKNOWLEDGEMENTS

I would like to thank Dr. Michael Leamy for his invaluable guidance and advice while working on this project. Without his support and patience, I would not have been successful in this endeavor. I also appreciate his advice and support throughout the project, which helped me grow intellectually and gain valuable skills as an engineer. In addition, I would like to thank Dr. Aldo Ferri and Dr. Harvey Lipkin for taking the time to serve on my thesis committee, and for providing feedback on the thesis itself.

I would also like to acknowledge the support of friends, who gave me encouragement and advice when needed. Kevin Manktelow was especially helpful when I encountered technical problems during research and needed someone discuss ideas with. My other lab mates; Jason Kulpe, Elsa Johnson, Farzad Sadeghi, Dekun Pei, John Arata and Farhad Farzbod; also provided a nice break from research when it became overwhelming or frustrating. Additionally, all the coffee breaks, lunch hours, and workouts taken with John Taphouse greatly improved my productivity and morale through the research and thesis writing process.

Support from my family was also instrumental in keeping me motivated and getting me to this point in my education. My mother encouraged me to learn and stay motivated from a young age, which influenced me to obtain my graduate education. My grandfather gave me encouragement and relief from thesis related work in our frequent conversations, and my sister gave me advice when I became overwhelmed with graduate student life. Additionally, I would like to thank Debbie Fossum for putting up with the long hours and late nights spent working on my research and thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$\mathbf{B}_i$	Local intrinsic coordinate system
BVP	Boundary value problem
C	The β-carbon on the protein backbone
CHARMM	Chemistry at HARvard Molecular Mechanics is a program for macromolecular simulations and defines an all-atom force field.
d	Distance along the helical axis between two C_α atoms
$\mathbf{D}_{eq}^{D_f}$	Final averaged diagonal equivalent stiffness matrix to model proteins with intrinsic beams.
$\mathbf{D}_{eq}^{f}$	Final averaged symmetric equivalent stiffness matrix to model proteins with intrinsic beams.
$\mathbf{D}_{eq}$	Equivalent stiffness matrix to model proteins with intrinsic beams.
$\mathbf{D}_K$	Intrinsic curvature stiffness matrix for elastic beams
FEA	Finite element analysis
FFT	Fast Fourier Transform
$\mathbf{F}_{V_i}$	Follower loads applied in local basis $\mathbf{V}_i$ in LAMMPS MD
IVP	Initial value problem
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator is an all atom molecular dynamics simulator developed by Sandia National Labs.
L_{Pep}	Length of the helix through a peptide plane
l^{rve}	Length of the rve
MD	Molecular Dynamics
N	Nitrogen atom on the protein back bone
ODE	Ordinary differential equation
QM	Quantum Mechanics
RMSD	Root mean squared deviation
rve	Representative Volume Element
$\mathbf{D}$	Intrinsic stiffness matrix for elastic beams

F	Internal force per until length vector in the local intrinsic basis
f	External force per until length vector in the local intrinsic basis
H	Angular momentum in the local intrinsic basis
M	Internal moment per until length vector in the local intrinsic basis
m	External moment per until length vector in the local intrinsic basis
P	Linear momentum in the local intrinsic basis
V	Velocity vector in the local intrinsic basis
U_s^{rve}	Simplified backbone model rve potential energy
V_i	Orthonormal basis for calculating follower loads in LAMMPS MD
XYZ	Global coordinate system
xyz	Helix coordinate system
a_2	Material point parameter specifying a distance in the $\mathbf{B}_2$ direction at x_1.
a_3	Material point parameter specifying a distance in the $\mathbf{B}_3$ direction at x_1.
α	Direction of follower load in XY plane before deformation
b_{ij}	Bond length
b_2	Material point parameter specifying a distance in the $\mathbf{B}_2$ direction at $x_1 + dx_1$.
b_3	Material point parameter specifying a distance in the $\mathbf{B}_3$ direction at $x_1 + dx_1$.
β	Direction of follower load in XY plane after deformation
F_{X_m}	m^{th} axial buckling load of a beam
C_α	The α-carbon on the protein backbone that is bonded to the residue side chain.
dx_1	Material point parameter denoting the distance down the centerline to a point away from the x_1 location.
η	Buckling load factor for axial buckling of beams
η^d	Dihedral inner displacement

$\boldsymbol{\gamma}$	Intrinsic strains of the beam
$\widehat{\boldsymbol{\gamma}}$	Net intrinsic strains of the beam solved for during shooting.
i_2	Mass moment of inertia per unit length about the $\mathbf{B}_2$ axis
i_3	Mass moment of inertia per unit length about the $\mathbf{B}_3$ axis
K	Intrinsic curvature of the beam
κ	Frenet-Serret curvature
$\widehat{\mathsf{K}}$	Net intrinsic curvatures of the beam solved for during shooting.
μ	Mass per unit length of the intrinsic beam
ω	The protein dihedral angle between the C and N atoms on the protein backbone.
Ω_f	Deformed configuration of the beam
$\boldsymbol{\Omega}$	Angular velocity vector in the local intrinsic basis
Ω_0	Initial configuration of the beam
Ω_{ref}	Reference configuration for the beam
ϕ	The protein dihedral angle between the C_α and N atoms on the protein backbone.
P_i^N	Global coordinates of the back bone atoms
ψ	The protein dihedral angle between the C_α and C atoms on the protein backbone.
$\mathsf{R}(x_1)$	Centerline position of the beam in global coordinate system
ρ	Radius of helix fit through C_α atoms
$\mathsf{R}^*(x_1 + dx_1, x_2, x_3)$	Position of material point off the center line of the beam
τ	Frenet-Serret twist
θ_{ijk}	Bond angle
θ	Rotation about the helical axis between two C_α atoms
u	Global X displacement
u_e	Internal energy per unit length of the continuum
v	Global Y displacement
w	Global Z displacement
x_1	Position along the intrinsic beam centerline

SUMMARY

Two multiscale continuum models for simulating protein dynamics are developed which allow for resolution of protein peptide planes in a beam-like finite element. A curvature and strain-based finite element formulation is utilized. This formulation is advantageous in simulating proteins since amino acid chains may be described by a single element, even when the protein segment considered exhibits large curvature and twist such as in the α-helical shapes prominent in many proteins. Concurrent and hierarchical multiscale models are developed for the curvature and strain-based beam formulation. The hierarchical multiscale continuum model utilizes a novel shooting method to calculate the deformed configuration of the protein. An optimization algorithm determines the requisite stiffness parameters by varying the beam stiffness used in the shooting method until deformed configurations of test cases correspond to those produced by the LAMMPS molecular dynamics software. Additionally, a concurrent multiscale method is detailed for evaluating protein inter-atomic potential parameters from the curvature and strain degrees of freedom employed in the model. This allows internal forces and moments to be calculated using nonlinear protein potentials. Proof of concept testing and model verification for both models includes comparing the multiscale techniques to all-atom molecular dynamics solutions. Specifically, the models are verified by simulating a polypeptide in a vacuum and comparing the predicted results to those computed using LAMMPS.

CHAPTER I

INTRODUCTION

1.1 Motivation

Proteins play a role in every major biological function from immune response to hormone regulation to cell division. Specifically, proteins provide the mechanisms and structure necessary for cellular functions and are therefore considered the building blocks of life. The ability of proteins to perform their cellular function is critical, and the protein's three-dimensional "native" shape determines how a protein functions [69]. It is understood that a protein's atomic structure and environment govern the protein's native shape, and that this shape is located at a global minimum of the protein's potential energy landscape [2]. If a cell cannot produce the correct atomic structure or an external stimuli disturbs the cellular system, the possibility exists where the protein will misfold and not attain its correct three-dimensional shape. These misfolded proteins lead to debilitating diseases and disorders that currently have few cures such as Alzheimer's disease, spongiform encephalopathies (prions disease), Parkinson's disease, Type II diabetes, multiple myeloma and several types of cancer ([15],[16]). The underlying mechanisms of the misfolding causing these diseases include gene mutations causing a change in the atomic structure of the molecule [47], failure of molecular chaperones to prevent misfolding [6], and environmental influences such as a change in pH [28] or increased shear stress due to restricted blood flow ([8], [35]). Since protein folding is critical to biological functions, and misfolding is a primary cause for serious illnesses, modeling protein kinetics accurately and quickly is of academic and medical interest. The ability to accurately and efficiently model protein dynamics and the protein folding process will aid in the design of new therapeutic drugs to treat incurable diseases [14].

Aside from therapeutics, the utility of designing novel proteins for the development of molecular machines provides further motivation for solving the protein folding and dynamics problem. In biological life, proteins are molecular machines manufactured by the cell to carry out cellular functions, which these proteins perform extremely efficiently. For example, plants use molecular machines to convert solar energy into synthesized organic compounds on a scale humans are unable to emulate [30]. Also, the genetic make-up of life is stored in the genetic code of DNA, which is transfered using molecular machines during cell division. During this process, protein molecular machines transfer and process data much more efficiently than silicon based computers ([17], [30]). In addition, bacteria contain molecular machinery that performs chemical synthesis and conversion. Examples of such molecular machines are already used in the processing of biofuels [25] and the bioremediation of chemical spills [50]. Gaining an understanding of the underlying processes at work in these molecular machines will aid in the modification of existing proteins and the design of novel molecules for use as molecular machines.

Currently, biologists possess the ability to synthesize proteins, but cannot predict their final conformation or how they will dynamically interact in a biological system. Several reasons contribute to the difficulty of modeling proteins efficiently and accurately. Firstly, proteins exhibit complex structure and a high number of degrees of freedom that make predicting native configurations computationally expensive. In addition, the energy landscape that governs the folding process and drives dynamics is “rugged.” This rugged energy landscape contains a large amount of local minimums, which makes determining the global minimum difficult. As a result, a simulation may not find the absolute global energy conformation. Lastly, quantum mechanics via the Schrödinger equation govern the atomic interactions causing the protein folding process. Implementing quantum mechanical theory in a protein simulation is exceedingly prohibitive due to computational expense; however, less accurate empirical potential fields employed in simulations demonstrate utility.

Recent advances in protein simulation tools demonstrate current techniques suffice for small protein problems; however, improvement is still needed. For example, the supercomputer Anton simulated the millisecond protein folding processes of 12 small proteins containing 10 - 80 amino acid residues [34] and the Rosetta@home research group modeled the 1.5 millisecond folding process of a 39 residue protein [67]. These simulations used empirical potential fields to describe atomic interactions and determined accurate tertiary protein structures. Although these simulations represent a major breakthrough in protein simulations, the goal of simulating complex proteins with simulation times on the order of seconds remains unattainable even with specialized supercomputers like Anton or distributed simulations like Rosetta@home. In order to realize simulations on this scale, new methods for simulating protein dynamics are required.

1.2 Protein Structure

Biological proteins are polymeric chains built from amino acid monomers. These amino acids contain 5 chemical components: a central α carbon (C_α), an α proton (H), an amino functional group (-NH), a carboxylic acid functional group (-COOH), and a side chain group (R). The residual side chain group differentiates the 20 common biological amino acids, which are listed in Table 1.1. These amino acids combine to become proteins through an energy-driven condensation reaction as a result of cellular function. A condensation reaction results when an amino functional group and a carboxylic acid functional group of two amino acids form a covalent bond between carbon and nitrogen as shown in Fig. 1.1. This results in the creation of a peptide bond between the two amino acids, and repeating the process creates a polypeptide containing several peptide bonds. Notably, these peptide bonds behave like a partial double bond, which have restricted rotation about the bond. This restricted rotation results in a stable peptide plane. Figure 1.2 depicts a polypeptide consisting of four peptide planes. These peptide planes are repeating units that exhibit constant structure in the protein and reduce the number of degrees of freedom in

Table 1.1: List of primary amino acids found in proteins.

Alanine	ALA	Arginine	ARG
Asparagine	ASN	Aspartic Acid	ASP
Cysteine	CYS	Glutamine	GLN
Glutamic Acid	GLU	Glycine	GLY
Histidine	HIS	Isoleucine	ILE
Leucine	LEU	Lysine	LYS
Methionine	MET	Phenylalanine	PHE
Proline	PRO	Serine	SER
Threonine	THR	Tryptophan	TRP
Tyrosine	TYR	Valine	VAL

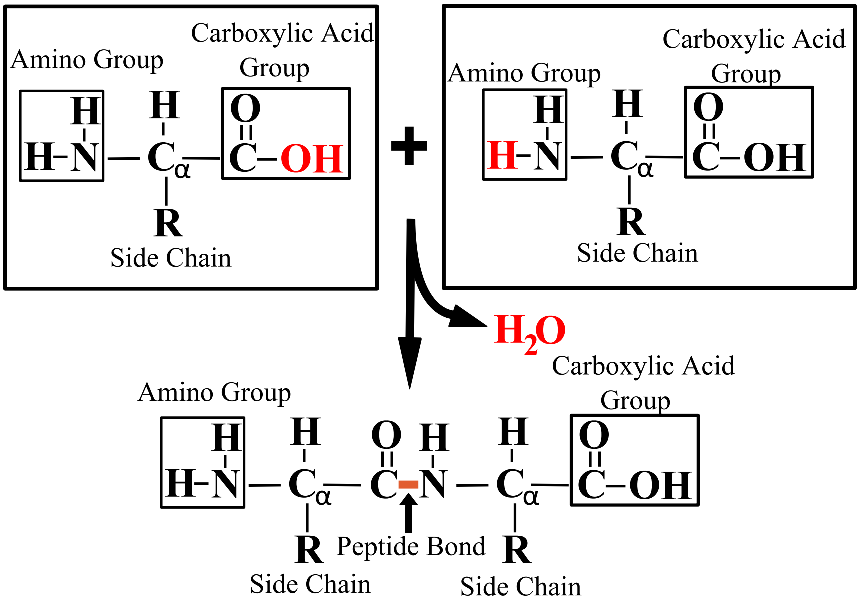


Figure 1.1: The amino acid consists of five chemical components: the C_α, the α proton, an amino group, carboxylic acid group, and a side chain group. Amino acids combine in a condensation reaction to form polypeptides.

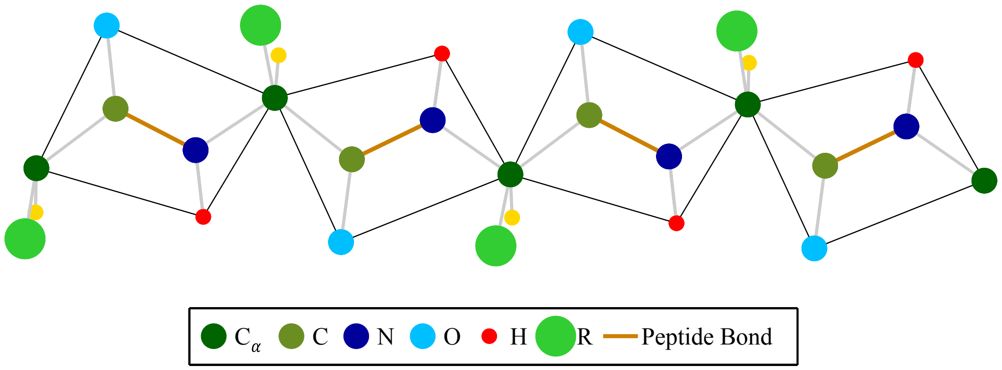


Figure 1.2: The polypeptide is a polymer chain of amino acids bonded with the peptide bond.

the protein.

Geometric relationships involving atoms in the polypeptide fully define the three-dimensional protein structure. The relationships consist of bond lengths, bond angles, dihedral angles and improper dihedral angles. Figure 1.3 illustrates these parameters and depicts important dihedral angles in the peptide plane. The primary contributors from these parameters, which determine overall polypeptide structure, are the dihedral angles ϕ and ψ defined in Fig. 1.3. Typically, the peptide plane remains relatively rigid during protein dynamics such

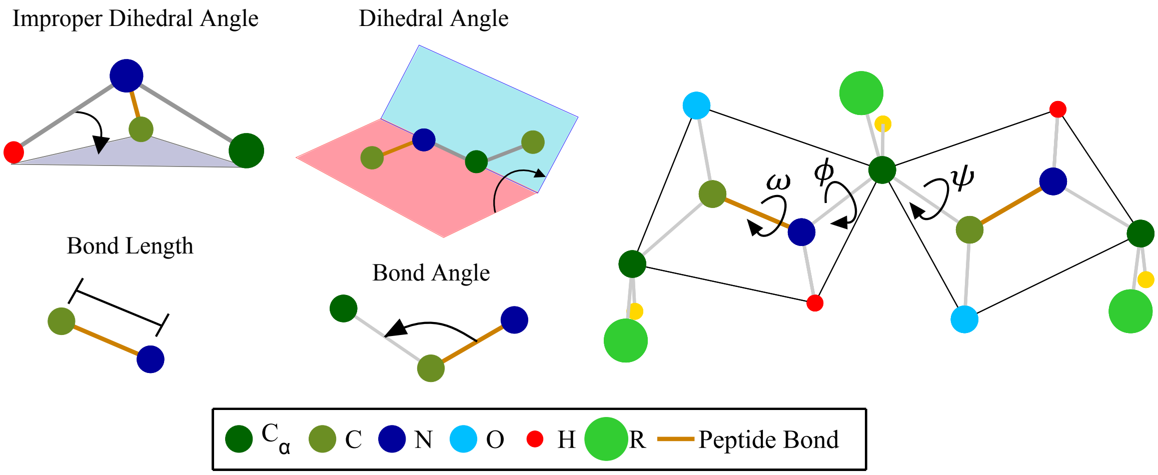


Figure 1.3: Bond lengths, bond angles, dihedral angles and improper dihedral angles give rise to protein structure. The dihedral angles ϕ, ψ and ω are located along the protein backbone.

that the bond lengths, bond angles and dihedral angle ω of the plane remain constant. As a result, the ϕ and ψ dihedral angles are the essential degrees of freedom that dictate the position of the polypeptide backbone atoms (C_α, C, N).

Four categories denote the different structural levels of a protein: primary, secondary, tertiary and quaternary structure. The primary structure consists of the amino acid sequence, which gives the list of amino acids found in the protein. For example, the primary structure for a protein consisting of four alanine amino acids is given by ALA-ALA-ALA-ALA. Secondary protein structure refers to commonly recurring geometric configurations in proteins such as the α-helix and β-sheets shown in Fig. 1.4. Furthermore, the ϕ and ψ dihedral angles define the secondary structures. Dihedral angles around $\phi = -64^\circ \pm 7^\circ$ and $\psi = -48^\circ \pm 7^\circ$ create an α-helix [5], and dihedrals angles near $\phi = -140^\circ$ and $\psi = 135^\circ$ result in an anti-parallel β-sheet [22]. Hydrogen bonding and steric restrictions make these secondary structures favorable [47]. Steric restrictions arise from the fact that atom electron clouds cannot overlap. This causes certain dihedral angle sets to be energetically unfavorable. Tertiary structure refers to the three-dimensional shape of a single polypeptide chain, while quaternary structure denotes the three-dimensional shape of two or more polypeptide chains interacting as a single biological molecule. Hemoglobin (see Fig. 1.5a) is an example of quaternary structure and consists of four polypeptide chain subunits bound together by

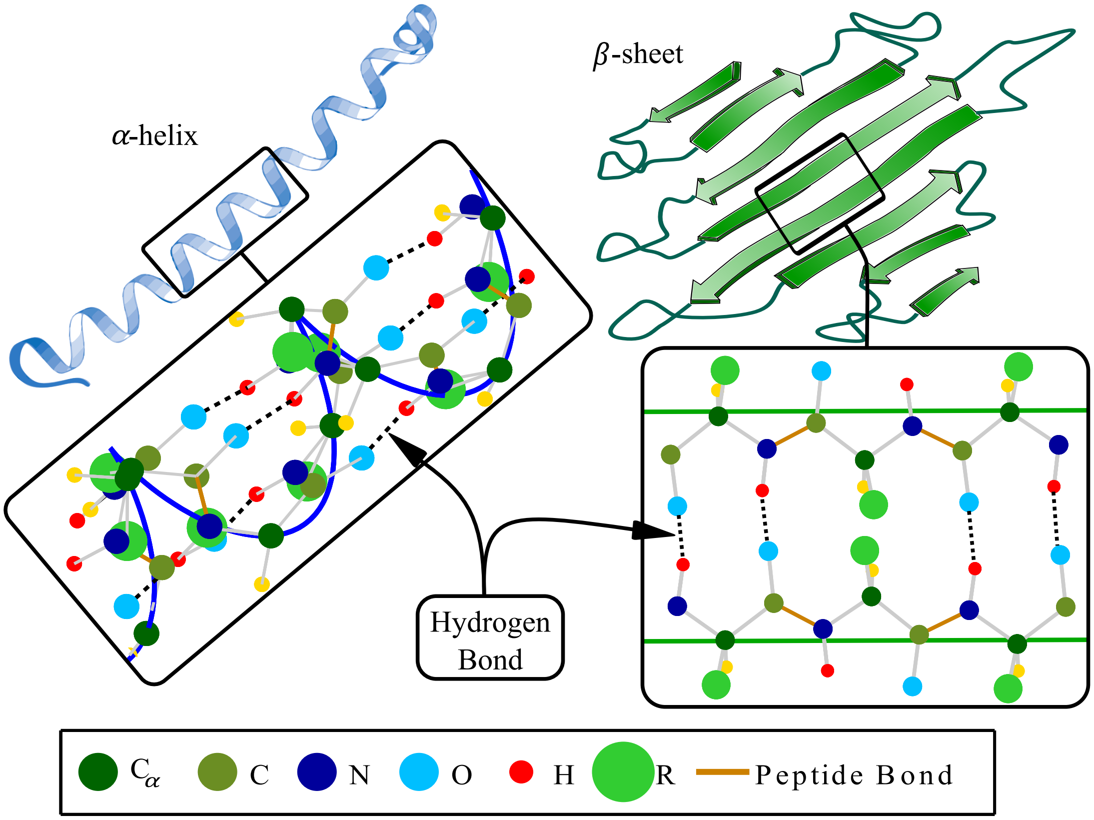


Figure 1.4: Common secondary structures appearing in proteins are the α-helix (blue) and the anti-parallel β-sheet (green). These secondary structures maintain stability due to hydrogen bonding.

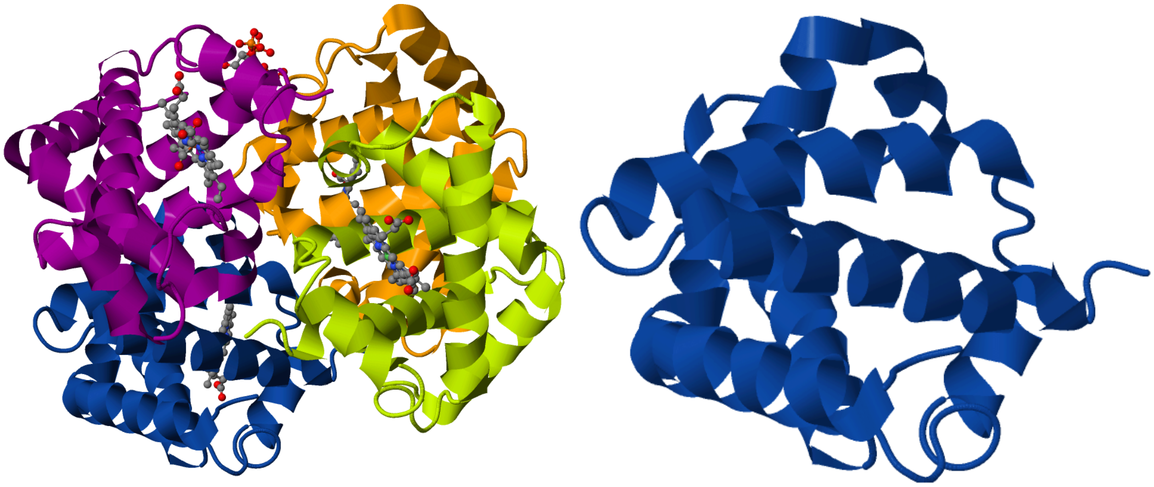
salt bridges, hydrogen bonds and the hydrophobic effect. The tertiary structure of one of Hemoglobin's molecular subunits is shown in Fig. 1.5b.

Proteins with known structures in the Protein Data Bank on average contain 633 amino acids, while the largest protein with identified structure consists of 57792 residues with hundreds of thousands of degrees of freedom [48]. Therefore, the large scale of the protein structure, along with the complex geometry and multiple atomic interactions, make simulating protein dynamics and folding extremely computationally expensive.

1.3 Literature Review

1.3.1 Traditional Protein Simulation Tools

Simulation tools developed for studying proteins fall into two broad categories: simulating protein kinetics and determining the native static shape of the protein. Simulating protein



(a) Hemoglobin quaternary structure (b) Hemoglobin subunit β-globin

Figure 1.5: (a) Hemoglobin has a quaternary structure containing four polypeptides (shown here in blue, yellow, orange, and purple) and four inorganic molecules (shown as gray and red atoms). (b) The tertiary structure of β-globin primarily exhibits α-helix secondary structures.

kinetics concerns studying the dynamics of the system when the time history of a biological process is desired. The primary dynamic simulation tool is molecular dynamics software (MD), which simulates all atoms in a protein over a given number of time steps. Molecular dynamics software packages also simulate reduced degree of freedom representations of proteins referred to as coarse-grained models where a single unit represents several atoms. Several molecular dynamics simulation packages exist such as LAMMPS, CHARMM, and Amber ([51], [10], [13]). In contrast, determination of the protein's native shape may not depend on the dynamic time history of the protein structure. Since the protein native shape exists at a global energy minimum, optimization algorithms are commonly used to determine protein structure. Examples of available protein folding programs that utilize optimization algorithms include CHARMM, Tasser, RaptorX, and Rosetta ([55], [49], [57]).

1.3.1.1 Molecular Dynamics

Molecular dynamics packages solve the differential equations governing the dynamics of the multi-body protein system. The multi-body problem can represent all atoms in the protein, or simplified coarse-grained models with groups of atoms represented by a large particle.

The governing equation for the dynamics of the system is

$$M\ddot{x} + C\dot{x} + \nabla V(x) = D\dot{W}(t), \tag{1.1}$$

where M denotes the mass matrix for the system, C denotes a damping matrix, and $V(x)$ denotes the protein potential function [41]. The forcing term on the right-hand side represents random forces imposed on the protein due to collisions with the surroundings that dissipate energy. The forcing term consists of normalized white noise $\dot{W}(t)$ and the matrix D, termed the diffusion matrix. The diffusion matrix is related to damping using

$$DD^T = 2k_BTC \tag{1.2}$$

where T denotes the temperature and k_B denotes the Boltzmann constant. The damping matrix and diffusion matrix represent coupling between the protein and its environment, and usually contains small terms. The use of these matrices simplifies this coupling, which is fully defined using quantum mechanics. Since implementation of the Shrödinger equation significantly increases computational expense, C is generally defined as a small scaling of the mass matrix γM, where the damping coefficient γ is determined from a heuristic method. The potential field is usually determined from empirical data, and V is defined as a function of the protein structural parameters and distances between nonbonded atoms that interact. The stiffnesses in potential terms are generally high and for a large protein the dimensional space of the problem is extremely large. As a result, the fastest molecular dynamics software running on specialized supercomputers simulate dynamics on the order of microseconds per day for average sized proteins [3]. As a result, modeling protein dynamics for seconds in simulation time would require an impractical amount of time and computational resources.

1.3.1.2 Energy Landscape Optimization

Several molecular modeling packages exist that utilize optimization algorithms to locate the energy minimum for the protein's energy landscape. Two methods exist for the optimization process: physics-based techniques and *de Novo* techniques. The physics-based processes run common optimization algorithms (e.g., Monte Carlo minimization and Newton-Based Methods) on the protein potential field in an attempt to find the native state [58]. Analytical methods like Newton-Based methods involve following the steepest gradient to the energy minimum. The gradient for the potential field is costly to calculate, and analytical methods tend to converge to local minimums away from global minimums. Statistical Monte Carlo minimization techniques apply random changes to the protein configuration and choose the lowest energy solution. This is repeated until a minimum is reached; however, the computational cost is prohibitive due to the sampling of many random configurations. Furthermore, modified algorithms are used which combine the analytical and statistical methods. In these modified algorithms, statistical methods sample the entire potential domain and identify low points in the domain. Analytical techniques then determine the true minimums located at these points. Then, the process is repeated on the lowest energy configuration found in the previous step until a minimum is determined [59]. As stated earlier, the energy function occupies a large dimensional space, which makes such an optimization task nearly impossible for most proteins. As a result, *de Novo*, or knowledge based methods, create folded configurations based on known protein structures and then optimization finds the local minimum. These *de Novo* methods include comparative modeling and fold recognition algorithms [21]. Comparative modeling matches residue sequences from the unknown protein to known proteins in the protein data bank and assumes that similar sequences imply similar structural characteristics. Fold recognition algorithms attempt to fit a specified sequence section to known structure in the protein data bank which may exhibit much different amino acid structure. These *de Novo* modeling methods demonstrate superiority in determining folded protein shapes over present day physics-based methods [58].

1.3.2 Force Fields

The force fields (also known as potential fields or energy functions) used in molecular modeling ultimately determine the final shape of protein. Two popular force fields are the CHARMM force field [36] and the Amber force field [52]. These force fields yield the energy of the protein as a function of the atomic positions. Specifically, the force fields consist of bonded terms and nonbonded terms. Bonded terms include energy terms based on bond lengths, bond angles, dihedral angles and improper dihedral angles. The bonded terms behave similar to linear or nonlinear springs depending on the parameter being modeled. The stiffnesses and values used in the spring-like terms depend on the chosen force field. Nonbonded terms capture weak interactions and electrostatic forces occurring between atoms in close proximity to each other. Equation (1.3) gives the CHARMM potential function based on atomic positions and protein structure parameters [37].

$$\begin{aligned} U(\mathbf{X}_p) = & \sum_{bonds} K_b(b-b_0)^2 + \sum_{UB}(S-S_0)^2 + \sum_{angles} K_\theta(\theta-\theta_0)^2 \\ & + \sum_{dihedrals} K_\chi(1+cos(n\chi-\delta)) + \sum_{impropers} K_\lambda(\lambda-\lambda_0)^2 \\ & + \sum_{nonbond} \left\{ \epsilon \left[\left(\frac{R_{min_{ij}}}{r_{p_{ij}}} \right)^{12} - \left(\frac{R_{min_{ij}}}{r_{p_{ij}}} \right)^{6} \right] + \frac{q_i q_j}{\epsilon_1 r_{p_{ij}}} \right\}. \end{aligned} \tag{1.3}$$

The stiffnesses for the bond lengths b, bond angles θ, dihedral angles χ and improper dihedral angles λ, are given by K_b, K_θ, K_χ and K_λ, respectively. The Urey-Bradley (UB) term represents additional energy due to bond angles where S denotes the UB length. The UB length is the distant between the first and third atom in a bond angle. The 0 subscript represents equilibrium values for each potential term. The nonbonded terms consist of the Lennard-Jones potential and the Coulomb potential where ϵ denotes the 6-12 Lennard-Jones well depth, R_{min} denotes the distance at the Lennard-Jones minimum, q_i denotes the partial atomic charge of atom i, ϵ_1 denotes the effective dielectric constant, and $r_{p_{ij}}$ denotes the distance between the i^{th} and j^{th} atom. The vector $\mathbf{X}_p$ denotes the position of all atoms in the protein from which all protein structure parameters and necessary distances can be calculated using geometric relationships [41].

1.3.3 Multiscale and Continuum Models

Continuum modeling represents a discrete system as a continuous body or fluid, and provides the starting point for multiscale modeling. For example, a continuous system with appropriate material properties and characteristics represents the discrete nano-structures involved with the protein folding process [44]. When modeling protein dynamics and the protein environment, continuum models typically represent solvents, cell membranes and the diffusion dynamics of the proteins within the cell [65]. Water is the primary solvent in which protein folding and dynamics occur, and the continuum representation of water involves the local ion concentration, which is represented by Poisson-Boltzmann theory or generalized Born methods. In cell membrane deformation, elastic material models represent the deformation of the membrane in response to external stimuli. The continuum representation of protein diffusion is governed by protein densities and mass transport equations. Although less common, several authors have developed continuum models which represent protein segments. To develop these models, force-displacement curves determined from MD or empirical data are used to calculate equivalent continuum material properties such as Young's Modulus or bending stiffnesses. For example, beam segments replace collagen fibers [11] or elastic membranes replace protein membranes [56]. For example, Goyal et. al. represent DNA loops containing hundreds of atoms as continuous beams and obtain dynamic response results efficiently [24]. When the material properties are developed using MD simulations, the resulting continuum models are *hierarchical* multiscale models, since data gathered from higher accuracy models parameterize the continuum models. Continuum models like these greatly reduce the degrees of freedom required to model the protein of interest; however, they lose atomistic resolution which governs more complex protein-protein interactions.

Concurrent multiscale methods link models at different scales such that the models are used simultaneously during simulations. These methods reduce the degrees of freedom for the overall model, but maintain atomistic resolution at areas of interest. One such method models crack growth in crystal structures at the atomistic level using a continuum representation. In this method, a finite element like mesh locates 'representative' atoms at each node, and linearly interpolates locations of atoms internal to mesh elements such that the full atomistic energy potential can be evaluated [38]. Near crack growth, mesh refinement allows atomistic resolution since the 'representative' atoms correspond to actual atoms. Such concurrent multiscale continuum models have not been developed for modeling protein polypeptide chains. Recent concurrent multiscale models for proteins link quantum mechanical modeling with molecular dynamics all-atom modeling. Segments of the polypeptide chain undergoing enzymatic reactions are represented by quantum mechanical modeling while the remaining segments are represented with molecular dynamics modeling [60].

1.4 Contribution and Organization

This work presents two multiscale continuum models for modeling protein polypeptide chains as continuous beams. The first method is a hierarchical method developed with an intrinsic beam formulation [27]. The developed hierarchical method uses a zero-iteration shooting method to optimize the intrinsic beam stiffnesses such that the beam correctly models protein characteristics calculated using MD. The concurrent multiscale method, on the other hand, embeds atomistic unit cells in the continuum model such that atomistic potentials determine beam internal forces and moments. Chapter 2 of this works covers the development of the zero-iteration shooting method for the intrinsic beam formulation, which is used to determine the beam stiffnesses required for the hierarchical model. The zero-iteration method is compared to similar shooting methods and validated with several test cases. Chapter 3 presents the development of the hierarchical multiscale continuum method which maintains multiscale resolution such that the position of C_α atoms and peptide planes are known. Results from this method obtained under various loading conditions are compared to values determined from MD. Chapter 4 discusses a preliminary concurrent multiscale method developed for modeling polypeptide chains using a finite element discretization. Comparisons between the preliminary concurrent multiscale method and MD are made at both the atomistic scale and the continuum level. Chapter 5 provides conclusions and areas of future work for both models.

CHAPTER II

THREE-DIMENSIONAL EQUILIBRIA AND STABILITY OF NONLINEAR CURVED BEAMS USING INTRINSIC EQUATIONS AND SHOOTING

This chapter covers the development of a novel shooting method used for calculating the hierarchical continuum model stiffness for proteins. The shooting method calculates the deformed configuration of helical beams subjected to follower loads with zero iterations. Since this method requires zero iterations to solve a follower load cantilever beam problem, it allows for computationally efficient calculation of equivalent protein stiffness inside of an optimization process. Chapter 3 describes in detail the optimization process which varies the intrinsic formulation stiffness terms until the deformed beam configuration matches the deformed configuration of a protein modeled with MD. This shooting method can be applied to any beam system, and as such this chapter presents the technique as a general method for obtaining three-dimensional beam equilibria and buckled shapes.

2.1 Introduction

Beams are a common element in mechanical systems used in modeling a variety of machine elements. As a result, studying the nonlinear deformation of beams is of interest to several fields of engineering such as aerospace, mechanical, biomedical, and civil engineering. Several authors, including Timoshenko[63]; Barten[7]; Bisshopp and Drucker[9]; Argyris[4]; and Pai and Lee[45], have performed in-depth studies on the nonlinear deformation of beams. In particular, beam problems with nonconservative follower loads are a subset which have received renewed interest. The follower load problem was first introduced as a study of elastic stability by Nikolai ([42], [43]) and was expanded upon by Ziegler[71]; however, this research was mainly viewed as a pure theoretical endeavor[26]. With advances in technology and material science, this category of beam problems was recognized to have practical engineering applications. The primary examples are in the medical and aerospace industries. Tools and equipment in these fields have become dependent on large flexible cables and beams, whether they be large robotic arms for spacecraft or endoscopic surgery tools. In robotic arms, follower moments exist at joints where servos control motion. With endoscopic tools, robotic catheters are in development which utilize shape memory alloys to control the motion of the catheter by applying distributed follower loads to the catheter body[66]. A commonly observed follower load phenomena is the flutter instability experienced by airplane wings due to aeroelastic effects. Gaining a greater insight into the nonlinear deformation of beams subjected to follower loads will aid in the design and control of these devices.

Solutions to large deformation beam problems have been investigated using several methods such as nonlinear finite element methods[4], iterative shooting methods[68], the finite difference method ([1], [23]), and less general analytical methods such as the elliptic integral formulation[63]. Numerical solutions to the large deformation of beams are of particular interest since these solution methods are applicable to general problems. In this subset of solution methods, traditional finite element methods, finite difference methods and shooting methods require multiple iterations in order to solve the beam boundary value problem (BVP). The finite element method must update load and stiffness matrices as the geometry varies until the final load case is reached, which increases computational expense and simulation time[29]. Finite difference methods require convergence of a set of residuals with the use of a Newton-Raphson technique or may employ iterative shooting methods in order to solve a nonlinear BVP[54]. Also, finite element and finite difference methods require a pre-processing step where the domain is discretized. For a beam defined by one independent variable, iterative shooting methods are useful for solving nonlinear BVP's.

Iterative shooting methods are numerical techniques for solving BVP's posed with one independent variable. These methods solve systems of first-order ordinary differential equations (ODE's) over an interval $[x_1^0, x_1^f]$ where x_1 is the independent variable. For the case of a beam with parameters and static equilibrium equations dependent on x_1, a set of fourth-order ODE's is reduced to a set of first-order ODE's and a set of boundary conditions is defined at both x_1^0 and x_1^f. With iterative shooting, this set of ODE's is treated as an initial value problem (IVP) in which multiple initial condition iterations at x_1^0 are performed in order to correctly arrive at the final conditions at x_1^f. In general, one cannot define a complete set of boundary conditions at x_1^0 such that the requisite boundary conditions at x_1^f are also satisfied. As a result, an initial guess of one or more boundary conditions at x_1^0 is required, followed by a minimization routine that varies the initial values at x_1^0 until all requisite boundary conditions at x_1^f are satisfied[54]. The varying of the initial values gives a different solution trajectory for each "shot," until the requisite boundary conditions are "hit" by the correct trajectory.

In contrast to traditional iterative shooting techniques for beam BVP's, a method was recently introduced by Shvartsman to solve planar cantilever beam problems subjected to follower loads by a direct shooting method[61]. This method uses variable substitution to arrive at a set of first-order ODE's which describe the equilibrium configuration of a beam undergoing large deflections. Some limitations of this direct shooting method are that it utilizes Euler-Bernoulli beam theory and has not been generalized for three-dimensional cases. Shvartsman's method also requires post-processing of data in order to calculate the deformed configuration of the loaded beam since the deformed configuration's governing equations are not part of the first-order ODE set[39]. In the present paper, a more general three-dimensional shooting method is introduced based on the intrinsic beam formulation[27].

The method proposed herein is a general shooting method to solve the large deflection problem of pre-curved, non-prismatic beams in three-dimensions. With this method, intrinsic beam theory with first-order shear modeling is used, allowing all relevant differential equations to be presented in first-order form. After the specification of a constitutive law, an ODE solver is used to find the deformed configuration of a loaded beam. For cantilever beam BVP's with nonconservative circulatory loading, this method gives an accurate solution with no iteration. This is possible for follower loads since the system equations are defined in a local coordinate system that rotates and translates with the beam, akin to the follower loads. In addition, only one boundary condition or initial value is needed since the equations are defined in first-order form. The additional boundary condition at the fixed end is automatically satisfied by the system equations since they enforce equilibrium.

For beam BVP's with conservative loading and general boundary conditions, iterative shooting is still required for the presented method. These general beam BVP's are modeled as a series of cantilever beams where reaction loads and external loads are both modeled as applied loads on the cantilever. The reaction loads and the direction of the external loads are varied until the known boundary conditions of the problem are satisfied on each end. The entire process is explained in detail in the methods section. A unique feature of this method is apparent when studying buckling beam problems. *All* stable and unstable solutions can be found for conservative-load BVP's as a result of the Picard-Lindelöf theorem guaranteeing the uniqueness of the solutions to IVP's with requisite smoothness. Due this theorem, the follower load beam BVP defined by the intrinsic formulation gives unique solutions for unique sets of initial conditions. As such, conservative beam BVP's are posed as follower load BVP's that satisfy the conservative BVP's final loading and boundary conditions. Many follower load solutions may exist that satisfy a given conservative load beam BVP. By sampling the entire initial condition subspace for the follower load BVP that models the conservative load BVP, all solutions for the conservative load problem are guaranteed to be found.

This paper contributes the following to analyzing the deformation of beams:

- a general procedure for finding the solutions to the nonlinear deformation of three-dimensional pre-curved cantilever beams subjected to follower loads without the need for iteration;

- a general procedure for determining *all* solutions to beam BVP's with conservative loading;

- a shooting method incorporating first-order shear modeling, which is particularly useful for the study of thick beams.

2.2 Methods

This section presents the intrinsic formulation's governing equations utilized in the shooting method. In addition, the constitutive law will be discussed, followed by a description for implementing shooting on the equilibrium equations.

2.2.1 Kinematics

There are three configurations, as shown in Fig. 2.1, required to describe the shape of a general three-dimensional beam for use with the intrinsic equilibrium equations. The reference configuration, Ω_{ref}, represents a straight beam with zero curvature or strain. The distance along the center line is defined by x_1 and a material point in the cross section at x_1 is defined by the triplet (x_1, x_2, x_3), which hold true in any of the configuration states. These material points are defined relative to the local basis of the beam in the Ω_{ref} configuration such that a material point is located at (x_1, x_2, x_3) in the global coordinate system XYZ. The initial configuration, Ω_0, represents the beam in an unstressed configuration which may exhibit initial curvature and strain, $\mathbf{K}_0$ and $\boldsymbol{\gamma}_0$. The curvature vector has three components which describe the twist and bending curvatures relative to the intrinsic basis of the beam, $\mathbf{B}_{0_1}, \mathbf{B}_{0_2}, \mathbf{B}_{0_3}$. The strain vector also has three components termed the axial and cross-sectional shear strains (analogous to the shear strain in Timoshenko beam theory). The deformed configuration, Ω_f, represents the beam in a stressed state where internal moments and forces are present. The final curvature and strain vectors, $\mathbf{K}_f$ and $\boldsymbol{\gamma}_f$, denote the change in curvature and strain for Ω_f relative to Ω_{ref}, while the net change in curvature and strain, $\widehat{\mathbf{K}}$ and $\widehat{\boldsymbol{\gamma}}$, denote the change from Ω_0 to Ω_f.

The intrinsic continuum model uses a unit basis to describe the beam kinematics in each configuration. For the Ω_{ref} configuration, the Cartesian unit vectors $[\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3]$ are used as the basis, while the two sets of basis vectors, $[\mathbf{B}_{0_1}, \mathbf{B}_{0_2}, \mathbf{B}_{0_3}]$ and $[\mathbf{B}_{f_1}, \mathbf{B}_{f_2}, \mathbf{B}_{f_3}]$, are used for Ω_0 and Ω_f, respectively. The basis vectors in Ω_0 and Ω_f follow the cross-section of the beam and are defined such that $\mathbf{B}_{0_2}$ and $\mathbf{B}_{f_2}$ initially correspond with $\mathbf{I}_2$ in Ω_{ref}, while $\mathbf{B}_{0_3}$ and $\mathbf{B}_{f_3}$ align with $\mathbf{I}_3$ in Ω_{ref}. The basis vectors $\mathbf{B}_{0_1}$ and $\mathbf{B}_{f_1}$ are defined by $\mathbf{B}_{0_2} \times \mathbf{B}_{0_3}$ and $\mathbf{B}_{f_2} \times \mathbf{B}_{f_3}$, completing the orthonormal set. Figure 2.2 shows these basis vectors, where it

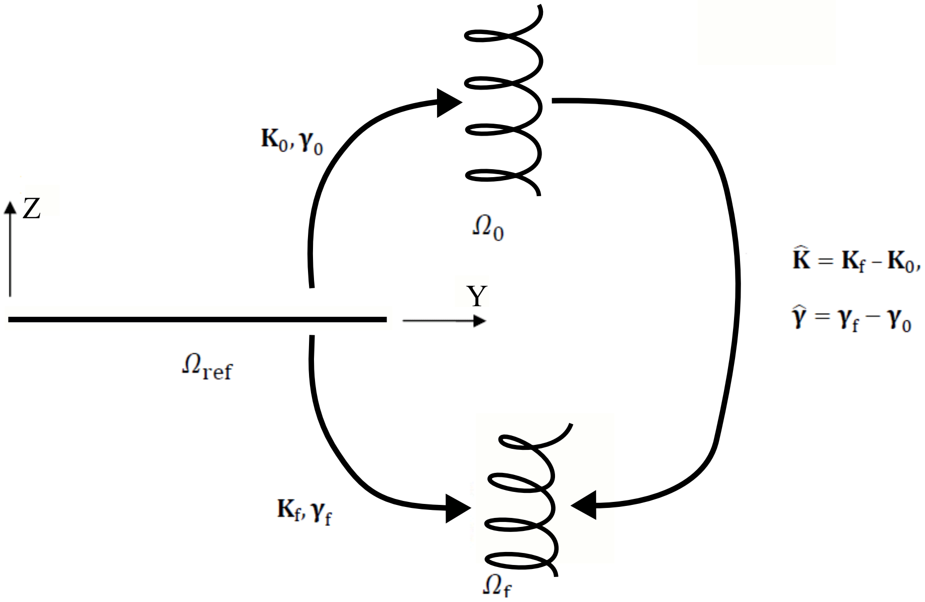


Figure 2.1: Three configurations are utilized in the intrinsic beam model. An initially straight configuration Ω_{ref} is used to define material points in the continuum. Initial curvature and strain, $\mathbf{K}_0$ and $\boldsymbol{\gamma}_0$, are used to map the material points to the initial configuration Ω_0. The deformed configuration Ω_f is achieved through a similar mapping from Ω_0 involving the net curvature and strain, $\widehat{\mathbf{K}}$ and $\widehat{\boldsymbol{\gamma}}$.

is evident that $\mathbf{B}_{0_1}$ and $\mathbf{B}_{f_1}$ are not necessarily tangent to the centerline $\mathbf{R}(x_1)$ in Ω_0 and Ω_f due to the presence of strain. The spatial rate of change of the centerline position and

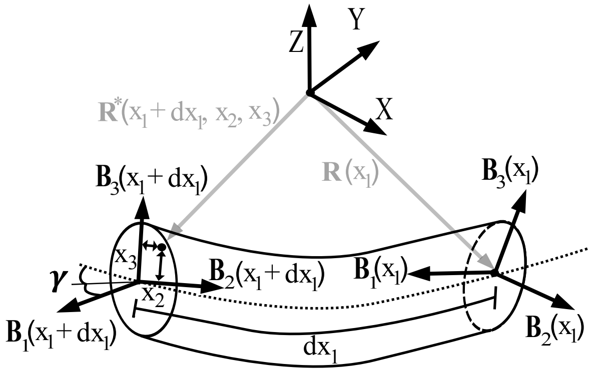


Figure 2.2: The basis vectors and material point position vectors used in the intrinsic beam development. These vectors can be assigned to Ω_0 or Ω_f utilizing the subscripts 0 or f, respectively.

the basis vectors implicitly define the strains and curvatures of the beam in both Ω_0 and

Ω_f[70] as shown in Eqs. (2.1) and (2.2):

$$\begin{aligned}\mathbf{R}_0' &= (1+\gamma_{0_{11}})\mathbf{B}_{0_1} + 2\gamma_{0_{12}}\mathbf{B}_{0_2} + 2\gamma_{0_{13}}\mathbf{B}_{0_3}, \\ \mathbf{R}_f' &= (1+\gamma_{f_{11}})\mathbf{B}_{f_1} + 2\gamma_{f_{12}}\mathbf{B}_{f_2} + 2\gamma_{f_{13}}\mathbf{B}_{f_3},\end{aligned} \tag{2.1}$$

$$\mathbf{B}_{0_i}' = \mathbf{K}_0 \times \mathbf{B}_{0_i},\ \mathbf{B}_{f_i}' = \mathbf{K}_f \times \mathbf{B}_{f_i}. \tag{2.2}$$

The spatial derivative with respect to x_1 is denoted by a prime.

2.2.2 Equilibrium Equations

An intrinsic beam formulation, developed by Hodges[27], defines the three-dimensional equations of motion governing the temporal and spatial changes of the beam's velocity, angular velocity, curvature and strain:

$$\frac{\partial \mathbf{F}}{\partial x_1} + \mathbf{K}_f \times \mathbf{F} + \mathbf{f} = \frac{\partial \mathbf{P}}{\partial t} + \boldsymbol{\Omega} \times \mathbf{P}, \tag{2.3}$$

and

$$\begin{aligned}\frac{\partial \mathbf{M}}{\partial x_1} + \mathbf{K}_f \times \mathbf{M} + (\mathbf{e}_1 + \boldsymbol{\gamma}_f) \times \mathbf{F} + \mathbf{m} = \\ \frac{\partial \mathbf{H}}{\partial t} + \boldsymbol{\Omega} \times \mathbf{H} + \mathbf{V} \times \mathbf{P},\end{aligned} \tag{2.4}$$

where **F** and **M** denote internal forces and moments; **f** and **m** denote external forces and moments per unit length; **P** and **H** denote the linear and angular momentum per unit length corresponding to linear velocity **V** and angular velocity $\boldsymbol{\Omega}$; and $\mathbf{e}_1$ denotes a unit vector in the $\mathbf{B}_{1_f}$ direction ($[1\ 0\ 0]^T$). The following two equations are necessary constraint equations to complete the set of four equations for the four field variables ($\mathbf{K}_f$, $\boldsymbol{\gamma}_f$, $\mathbf{V}$, $\boldsymbol{\Omega}$):

$$\frac{\partial \boldsymbol{\Omega}}{\partial x_1} + \mathbf{K}_f \times \boldsymbol{\Omega} = \frac{\partial \widehat{\mathbf{K}}}{\partial t} \tag{2.5}$$

and

$$\frac{\partial \mathbf{V}}{\partial x_1} + \mathbf{K}_f \times \mathbf{V} + (\mathbf{e}_1 + \boldsymbol{\gamma}_f) \times \boldsymbol{\Omega} = \frac{\partial \boldsymbol{\gamma}_f}{\partial t}. \tag{2.6}$$

The momenta and velocities of the beam are related using the mass per unit length μ; cross-sectional mass moments and product of inertia i_2, i_3 and i_{23}; and the centroidal offsets from the center line $\overline{x}_2$ and $\overline{x}_3$ as given by Eq. (2.7):

$$\begin{Bmatrix} P_1 \\ P_2 \\ P_3 \\ H_1 \\ H_2 \\ H_3 \end{Bmatrix} = \begin{bmatrix} \mu & 0 & 0 & 0 & \mu\overline{x}_3 & -\mu\overline{x}_2 \\ 0 & \mu & 0 & -\mu\overline{x}_3 & 0 & 0 \\ 0 & 0 & \mu & \mu\overline{x}_2 & 0 & 0 \\ 0 & -\mu\overline{x}_3 & \mu\overline{x}_2 & i_2+i_3 & 0 & 0 \\ \mu\overline{x}_3 & 0 & 0 & 0 & i_2 & i_{23} \\ -\mu\overline{x}_2 & 0 & 0 & 0 & i_{23} & i_3 \end{bmatrix} \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{Bmatrix}. \tag{2.7}$$

For the static case, these equations simplify greatly by removing time dependent terms and derivatives. As a result, the equilibrium equations are given by

$$\frac{\partial \mathbf{F}}{\partial x_1} + \mathbf{K}_f \times \mathbf{F} + \mathbf{f} = \mathbf{0} \tag{2.8}$$

and

$$\frac{\partial \mathbf{M}}{\partial x_1} + \mathbf{K}_f \times \mathbf{M} + (\mathbf{e}_1 + \boldsymbol{\gamma}_f) \times \mathbf{F} + \mathbf{m} = \mathbf{0}. \tag{2.9}$$

A constitutive law relating the internal forces and moments to curvature and strain completes the formulation. For all beams considered herein, the initial curvatures are small-enough to warrant a decoupled constitutive model[32] of the form,

$$\begin{Bmatrix} \mathbf{F} \\ \mathbf{M} \end{Bmatrix} = [\mathbf{D}] \begin{Bmatrix} \widehat{\gamma} \\ \widehat{\mathbf{K}} \end{Bmatrix}, \tag{2.10}$$

where

$$\mathbf{D} = \begin{bmatrix} AE & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2AG}{k} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2AG}{k} & 0 & 0 & 0 \\ 0 & 0 & 0 & (i_2+i_3)G & 0 & 0 \\ 0 & 0 & 0 & 0 & i_2E & 0 \\ 0 & 0 & 0 & 0 & 0 & i_3E \end{bmatrix}. \tag{2.11}$$

In Eq. (2.11), E denotes the material elastic modulus, G denotes the shear modulus defined by $G = E/(2(1+\nu))$, ν denotes Poisson's ratio, $\mathbf{D}$ denotes the intrinsic stiffness and k denotes a shear correction factor based on the cross section shape. This constitutive law is for a continuum with a symmetric cross-section and an isotropic material. For use in this work the constitutive law is linearized for small $\widehat{\mathbf{K}}$ and $\widehat{\boldsymbol{\gamma}}$; however, the formulation is general such that a nonlinear constitutive law could be used. In the case where strain is negligible, only the curvature-moment relations are required from Eq. (2.10). As a result, the constitutive law becomes

$$\{\mathbf{M}\} = [\mathbf{D}_K] \left\{\widehat{\mathbf{K}}\right\} \tag{2.12}$$

where the curvature stiffness $\mathbf{D}_K$ is

$$\mathbf{D}_K = \begin{bmatrix} (i_2+i_3)G & 0 & 0 \\ 0 & i_2E & 0 \\ 0 & 0 & i_3E \end{bmatrix}. \tag{2.13}$$

2.3 Solving the intrinsic equilibrium equations using a shooting method

In order to use a shooting method on the equilibrium equations, the ODE set is treated as an IVP and the equations must be rearranged into the form

$$\mathbf{y}'(x_1) = \mathbf{f}(x_1, \mathbf{y}(x_1)). \tag{2.14}$$

Equations (2.1), (2.2), (2.8), and (2.9) are algebraically manipulated into the required form using Eqs. (2.10) and (2.11). To facilitate initial curvatures and remove initial strains, $\mathbf{K}_0+\widehat{\mathbf{K}}$ replaces $\mathbf{K}_f$ and $\widehat{\boldsymbol{\gamma}}$ replaces $\boldsymbol{\gamma}_f$ in the aforementioned equations. As a result, eighteen scalar first-order nonlinear differential equations are obtained for the spatial derivatives of $\widehat{\mathbf{K}}$, $\widehat{\boldsymbol{\gamma}}$, $\mathbf{B}_i$, and $\mathbf{R}$. There are three scalar equations each for $\widehat{\mathbf{K}}$, $\widehat{\boldsymbol{\gamma}}$, and $\mathbf{R}$, while nine scalar equations are required for the basis $\mathbf{B}_i$. These equations express $\widehat{\mathbf{K}}'$ and $\widehat{\boldsymbol{\gamma}}'$ in the $\mathbf{B}_i$ basis, while $\mathbf{B}_i'$ and $\mathbf{R}'$ are expressed with respect to the global reference basis. To express all quantities in a common basis, Eq. (2.2) is modified as follows:

$$\mathbf{B}_i' = \mathbf{K}_G \times \mathbf{B}_i \tag{2.15}$$

where

$$\mathbf{K}_G = \sum (\mathbf{K}_0 + \widehat{\mathbf{K}})_j \mathbf{B}_j. \tag{2.16}$$

Appendix A presents the set of equations in full detail using the material constitutive relationship from Eq. (2.11). Treating this ODE set as an IVP requires the specification of eighteen initial conditions. The $\mathbf{R}$ and $\mathbf{B}_i$ initial conditions only affect the final configuration orientation in the global basis, and do not influence the solution for $\widehat{\mathbf{K}}$ and $\widehat{\boldsymbol{\gamma}}$. Therefore, the initial conditions for $\mathbf{R}$ and $\mathbf{B}_i$ are set to arbitrary values and the $\mathbf{B}_i$ values must obey the orthonormal constraint required for the basis. The beam BVP loading and boundary conditions dictate the $\widehat{\mathbf{K}}$ and $\widehat{\boldsymbol{\gamma}}$ initial conditions. After specification of the initial values, a numerical ODE solver calculates the solution to the IVP (e.g., in this work Matlab's ODE45 solver is used).

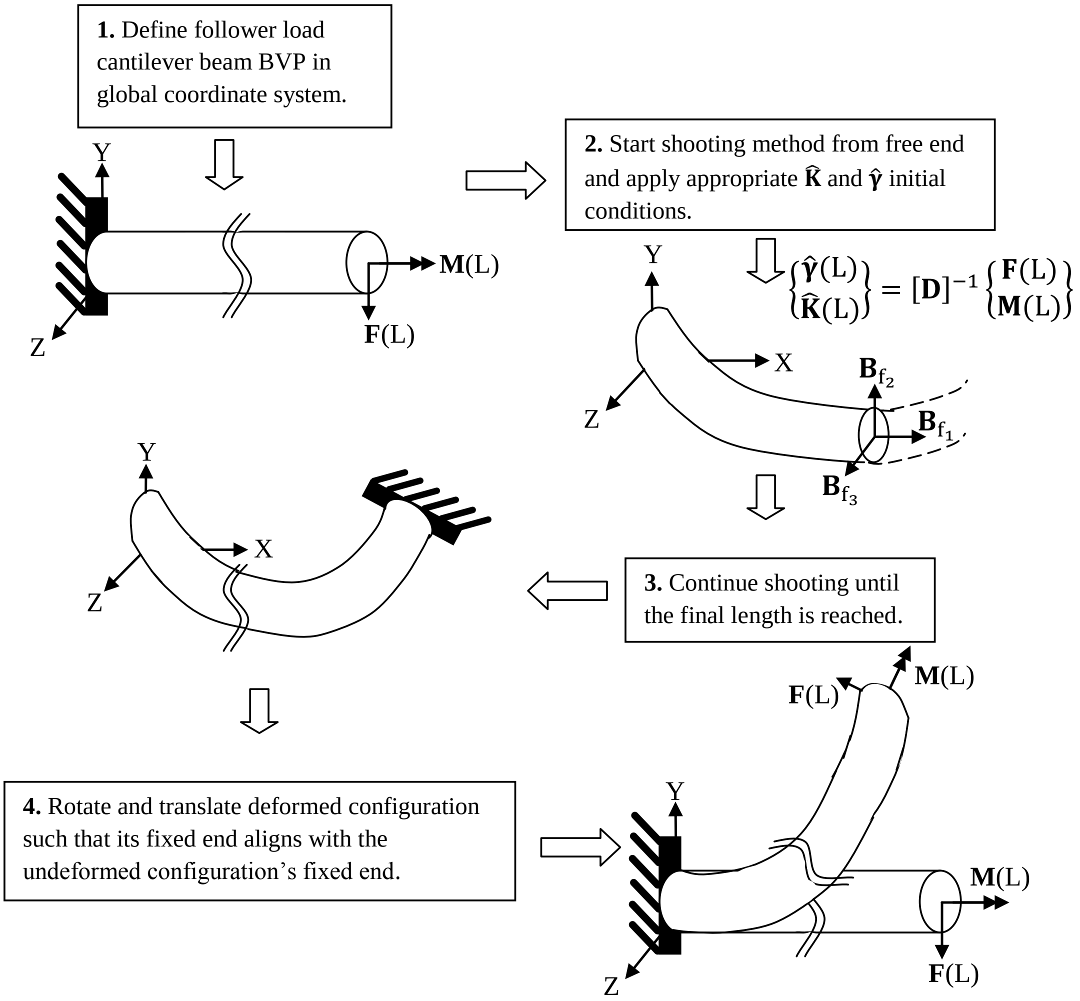


Figure 2.3: The shooting method for cantilever beams subjected to follower loads has four main steps: define problem, initiate shooting from the free end with $\mathbf{R}(L) = \mathbf{0}$ and $\mathbf{B}_{1_f}$ aligned with X, integrate the ODE set until the desired length of the beam is reached, rotate and translate the deformed configuration such that the fixed end is aligned with the undeformed configuration.

For a cantilever beam subjected to follower loads, shooting is initiated from the free end of the beam and requires no iteration. As a result, a single "shot" determines the nonlinear deformation of the beam. This is only valid for beam BVP's with follower loads since the basis in which the equilibrium equations are expressed translates and rotates with the beam deformation. The equilibrium equations allow the application of point loads at the free boundary and distributed loads along the beam. The $\widehat{\mathbf{K}}$ and $\widehat{\gamma}$ initial conditions amount to applying the point follower loads to the end of the cantilever. An algebraic manipulation of the constitutive law defines these initial conditions as a function of the end load using

$$\left\{ \begin{array}{c} \widehat{\gamma}(L) \\ \widehat{\mathbf{K}}(L) \end{array} \right\} = \left[\mathbf{D}^{-1}\right] \left\{ \begin{array}{c} \mathbf{F}(L) \\ \mathbf{M}(L) \end{array} \right\}. \tag{2.17}$$

The appropriate values for $\mathbf{m}$ and $\mathbf{f}$ in Eqs. (2.8) and (2.9) apply distributed follower loads to the beam. These distributed loads can be constant or defined as functions of x_1. Furthermore, non-prismatic cantilever beams are easily modeled when the material properties and beam dimensions vary as a function of x_1. The appropriate functions defining these beam parameters are implemented in the constitutive law given by Eq. (2.11). The equilibrium equations automatically enforce the fixed boundary condition at the end of the cantilever. After numerical integration of the ODE set, a rotation and displacement applied to the deformed configuration relocates the deformed configuration in the global basis. The displacements of the beam are calculated with

$$\left\{ \begin{array}{c} u \\ v \\ w \end{array} \right\} = \mathbf{R}_f - \mathbf{R}_0 \tag{2.18}$$

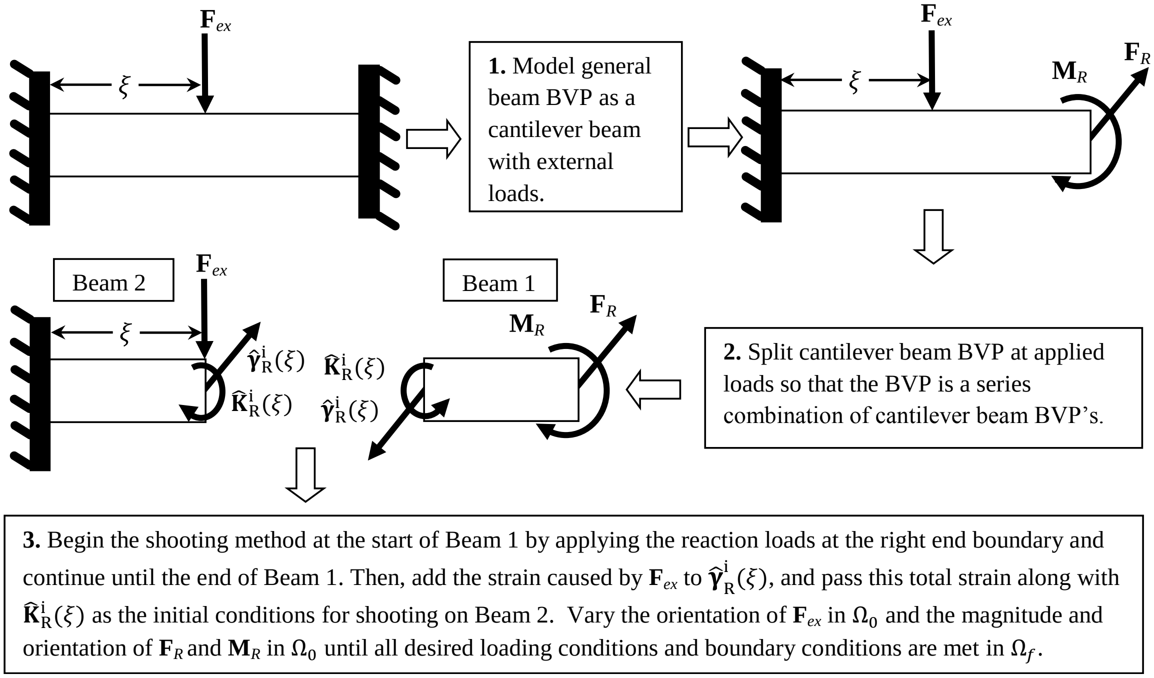
where u is the global X-displacement, v is the global Y-displacement and w is the global Z-displacement. The orientation of the follower loads in the global basis is given by

$$\left\{ \begin{array}{c} L_X \\ L_Y \\ L_Z \end{array} \right\} = \left[\begin{array}{ccc} \mathbf{B}_{f_1} & \mathbf{B}_{f_2} & \mathbf{B}_{f_3} \end{array} \right] \left\{ \begin{array}{c} L_1 \\ L_2 \\ L_3 \end{array} \right\} \tag{2.19}$$

where $[L_X, L_Y, L_Z]^T$ denotes the global force or moment vector and $[L_1, L_2, L_3]^T$ denotes the applied follower force or moment vector. The flow chart in Fig. 2.3 clearly outlines this procedure for modeling cantilever beams with follower loading.

This shooting method also generates solutions to general beam BVP's with any combination of boundary conditions, conservative loads and nonconservative loads; however, iterative shooting must be utilized. Figure 2.4 illustrates the procedure for solving general problems. First, the problem is converted into a series combination of cantilever beams. For general BVP's, the reaction loads caused by imposed boundary conditions and loading are modeled as external loads on a cantilever beam. For example, Fig. 2.4 depicts a fixed-fixed beam transversely loaded with conservative force $\mathbf{F}_{ex}$ located at $x_1 = \xi$. This BVP is modeled as a cantilever beam with a reaction force $\mathbf{F}_R$ and moment $\mathbf{M}_R$ on the free end in addition to $\mathbf{F}_{ex}$ at $x_1 = \xi$. The next step involves decomposing the original BVP into the two cantilever beam problems denoted by Beam 1 and Beam 2. The initial values for Beam 1 are an initial guess for $\mathbf{F}_R$ and $\mathbf{M}_R$, while the initial values for Beam 2 include the arrived-at values from Beam 1, namely $\widehat{\mathbf{K}}^i_R(\xi)$ and $\widehat{\gamma}^i_R(\xi)$, along with an initial guess for the orientation of $\mathbf{F}_{ex}$. An initial guess is required for $\mathbf{F}_{ex}$ since conservative loads are modeled as follower loads that are rotated in Ω_0 until the follower load in Ω_f is oriented in the direction of the desired conservative load. An optimization algorithm then determines the values for $\mathbf{F}_R$ and $\mathbf{M}_R$, and the correct orientation of $\mathbf{F}_{ex}$ on Beam 2, which satisfy the original beam BVP. This optimization minimizes a cost function involving the deformed configuration's adherence to boundary conditions and the Ω_f-orientation of the applied external forces. The actual form of the cost function used in the optimization depends on the imposed boundary conditions and loads. Modification of this cost function allows modeling of any general beam BVP.

This iterative method for general beam BVP's possesses the unique ability to determine all solutions to the beam problem of interest. The Picard-Lindelöf theorem guarantees the existence and uniqueness of the solutions to an ODE set with specified initial conditions[20]. For the equilibrium beam equations utilized in this work, the function $\mathbf{f}(x_1, \mathbf{y}(x_1))$ appearing in Eq. (2.14) is Lipschitz continuous in $\mathbf{y}$ and continuous in x_1. Therefore, a unique solution exists for any initial condition set specified for the follower load beam BVP posed



1. Model general beam BVP as a cantilever beam with external loads.

2. Split cantilever beam BVP at applied loads so that the BVP is a series combination of cantilever beam BVP's.

3. Begin the shooting method at the start of Beam 1 by applying the reaction loads at the right end boundary and continue until the end of Beam 1. Then, add the strain caused by $\mathbf{F}_{ex}$ to $\hat{\boldsymbol{\gamma}}^{\mathrm{i}}_{\mathrm{R}}(\xi)$, and pass this total strain along with $\hat{\boldsymbol{\kappa}}^{\mathrm{i}}_{\mathrm{R}}(\xi)$ as the initial conditions for shooting on Beam 2. Vary the orientation of $\mathbf{F}_{ex}$ in Ω_0 and the magnitude and orientation of $\mathbf{F}_R$ and $\mathbf{M}_R$ in Ω_0 until all desired loading conditions and boundary conditions are met in Ω_f.

Figure 2.4: The shooting method requires that general beam BVP's be split into a series combination of cantilever beams. The external loads and reaction loads are then varied until applied loading conditions and boundary conditions are satisfied.

by the intrinsic equilibrium equations. As a result, varying the force and moment initial conditions for a follower load beam BVP over their entire range (e.g., by rotating the applied loads through all possible directions) and compiling the solutions that satisfy the desired conservative load beam BVP yields *all* solutions for the conservative beam BVP. Specifically for the two-dimensional Euler buckling of a fixed-free beam loaded by a *conservative* axial force, rotating a *follower* load over the range $[0, 2\pi]$ in Ω_0 and determining all solutions that satisfy the correct deformed load orientation in Ω_f gives all buckled solutions to this BVP. A later section presents results and discussion for the Euler buckling problem in detail.

After determining all solutions for a BVP, the stability of each solution can be assessed using analytical linearization techniques on the dynamic equations[18] or a dynamic numerical solver. For this work, a dynamic finite element code written specifically for the intrinsic beam equations of motion[32] makes determining system stability trivial. First, the general shooting method calculates the Ω_f configuration for a specific load case. Then, an automated process ports the curvatures and strains from this configuration, along with the necessary boundary conditions and loads, into the dynamic finite element code. Since the finite element program accepts curvatures and strains as the nodal degrees of freedom, the deformed geometry from shooting transfers to the finite element code with ease. Next, the system configuration is perturbed slightly and the dynamic finite element code simulates system response for an extended period of time. An exponential growth of the system response over time demonstrates system instability, while a stable system experiences oscillations proportional to the disturbance.

In situations where neglecting strain is warranted, only the curvature-moment relations from the constitutive law in Eqs. (2.12) and (2.13) need to be substituted into the equilibrium equations. The equations neglecting strain will be utilized in later sections where quantitative results are compared with literature. When neglecting strain, the set of field variables now include **F** instead of $\widehat{\gamma}$, which makes the application of point force loads at the end of the beam straight forward. These equations are presented in their entirety in Appendix A.

2.4 Results

The presented shooting method exhibits its strongest advantages when applied to cantilever beams subjected to nonconservative follower loads - these cases do not require iteration. The following test cases validate the method using both literature comparisons ([46], [4]) and a unique comparison of a three-dimensional system with a commercial finite element code.

2.4.1 Follower Load Validation

2.4.1.1 Cantilever Beam Subjected to Moment Load.

The first investigation involves a point moment load on the end of a two dimensional, initially straight beam. The results given for this case are parameterized by the beam properties; therefore, any arbitrary beam parameters that define a beam with a symmetric cross section and isotropic material properties will yield the presented response. Figure 2.5 illustrates the beam BVP loading and boundary conditions, and shows the deformed configurations calculated using zero-iteration shooting. The exact solution for the curvature for this loading case is

$$\widehat{K}_3 = \frac{M_Z}{EI_{ZZ}}. \tag{2.20}$$

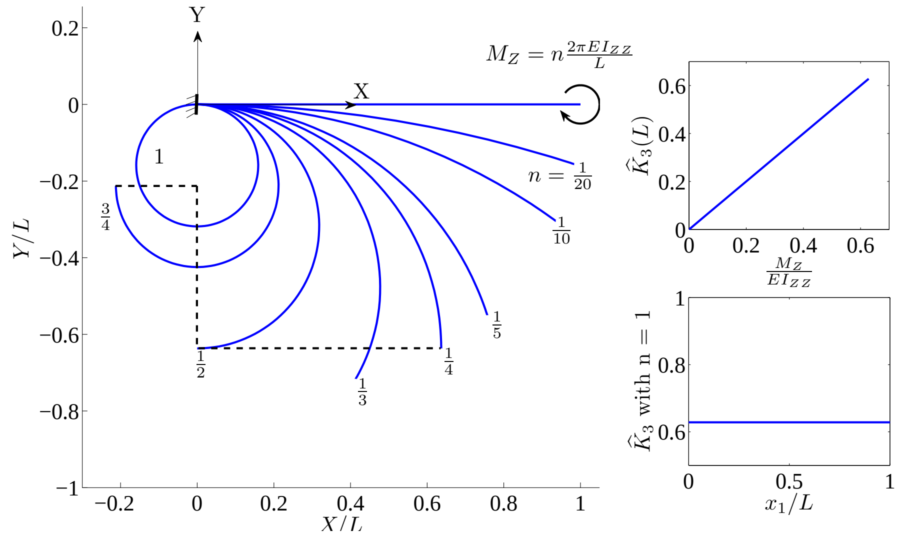


Figure 2.5: The deformed configurations of a beam subjected to an increasing point moment located at the free end calculated using the developed shooting method. The two subfigures illustrate that $\widehat{K}_3$ increases linearly with M_Z and that $\widehat{K}_3$ is constant along the beam arc length as predicted by theory.

Table 2.1: Beam dimensions and material properties for straight beams used for method validation.

Length$[cm]$	100.00
Area Moment of Inertia $I_{ZZ}[cm^4]$	1.67
Area$[cm^2]$	20.00
Young's Modulus$[\frac{N}{cm^2}]$	2.10×10^7
Poison's Ratio	0.30

The subfigures in Fig. 2.5 validate the shooting solution since the predicted $\widehat{K}_3$ is exactly proportional to $\frac{M_Z}{EI_{ZZ}}$, and $\widehat{K}_3$ is constant throughout the length of the beam, as required.

2.4.1.2 Straight and Pre-Curved Cantilever Beams Subjected to Point and Distributed Forces.

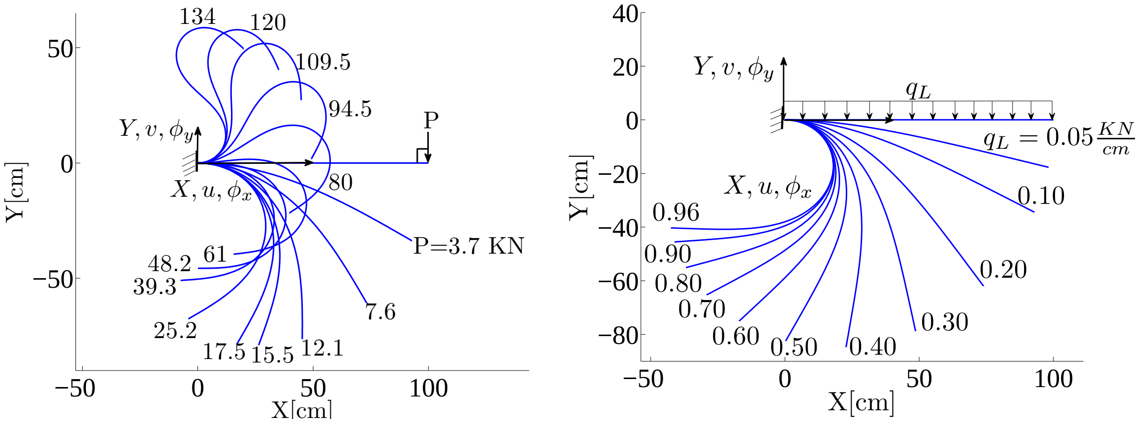
The next validation results are for straight and pre-curved cantilever beams loaded by point follower forces, and a straight cantilever beam loaded by a follower distributed force. Tables 2.1 and 2.2 present the material properties for each case and match those from [4].

Figure 2.6a presents the deformed configurations of an initially straight beam subjected to a perpendicular, point, follower force at the free end for multiple load magnitudes. Figure

Table 2.2: Beam dimensions and material properties for pre-curved beams used for method validation.

Length$[cm]$	157.08
Area Moment of Inertia $I_{ZZ}[cm^4]$	0.50
Area$[cm^2]$	1.00
$K_{0_3}[cm^{-1}]$	0.02
Young's Modulus$[\frac{N}{cm^2}]$	7.20×10^6
Poison's Ratio	0.30

2.6b shows the deformed configurations of an initially straight beam subjected to a perpendicular, distributed, follower force across the length of the beam for various magnitudes.

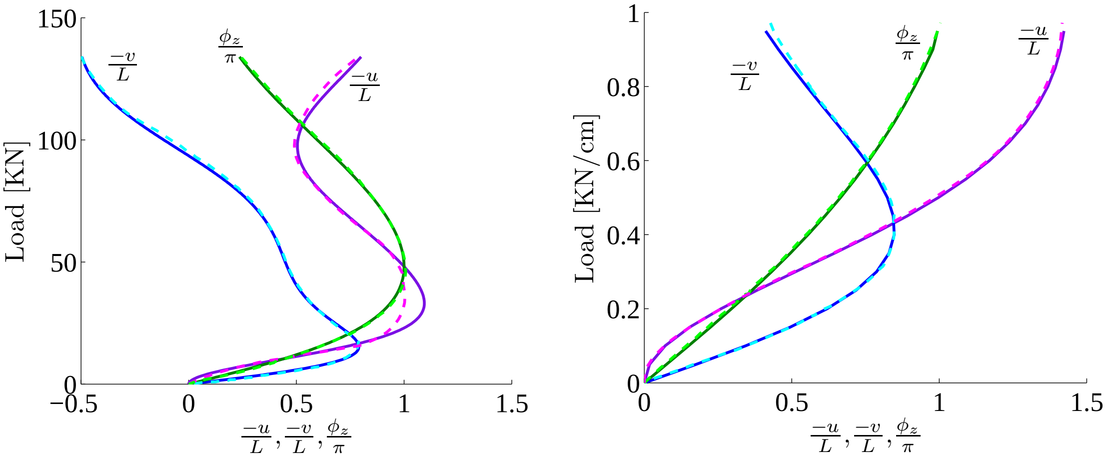


(a) Perpendicular Point Loading

(b) Perpendicular Distributed Loading

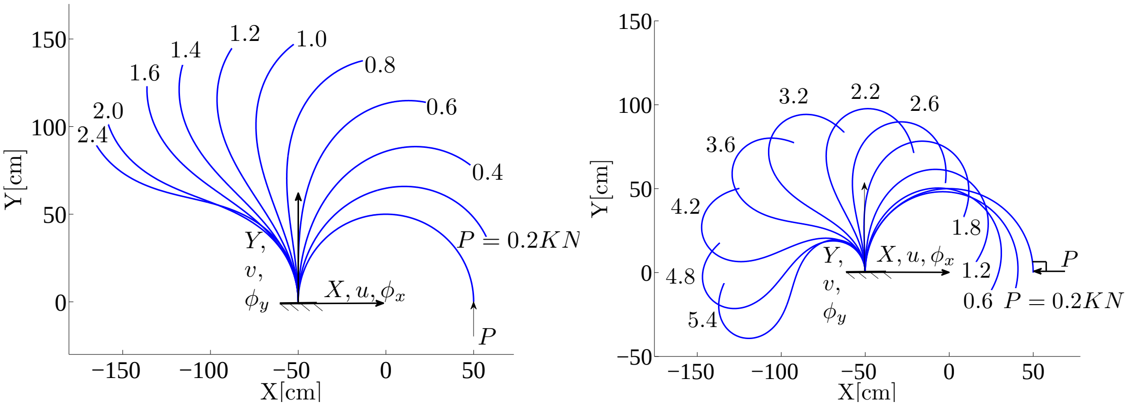
Figure 2.6: The deformed configurations of straight beams with (a) perpendicular point and (b) perpendicular distributed follower loads calculated using the presented shooting method.

For both of these cases, the results obtained with zero-iteration shooting compare favorably to the results obtained from the finite element analysis (FEA) work done by Argyris [4] The quantitative results used for comparison (see Fig. 2.7) comprise of the normalized displacements and the rotation ϕ_z about the Z axis for the end of the beam. It is evident from this figure that results generated using the presented method compare well with the FEA results presented by Argyris, other than minor disagreement in Fig. 2.7a for $\frac{-u}{L} = 1$. These quantitative comparisons validate the method's ability to model the nonlinear deformation of straight beams subjected to point and distributed follower forces.



(a) Perpendicular Point Loading (b) Perpendicular Distributed Loading

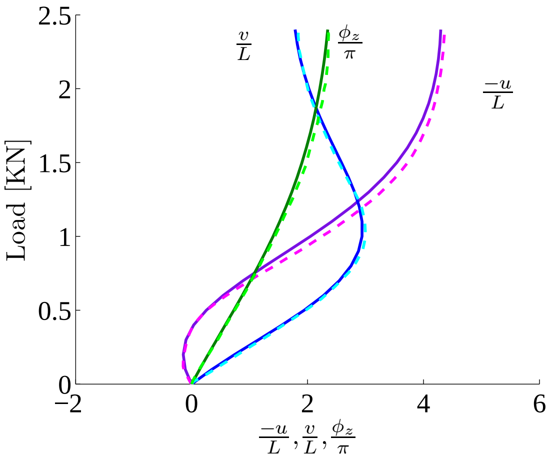
Figure 2.7: The normalized displacements and rotations for the straight beams subjected to (a) perpendicular point and (b) perpendicular distributed follower loads compared with the finite element results presented by Argyris [4]. The dashed lines are data from [4] while the solid lines are results from the method presented herein.



(a) Pre-Curved Tangent Loading (b) Pre-Curved Perpendicular Loading

Figure 2.8: The deformed configurations of pre-curved beams subjected to (a) tangential and (b) radial point follower loads calculated using the method presented herein.

Furthermore, Figs. 2.8 and 2.9 validate the ability of the shooting method to model pre-curved beams without iteration. Two cases are considered: a follower point force tangent to the pre-curved beam at the free end, and a follower point force perpendicular to the pre-curved beam at the free end. As before, Fig. 2.8 depicts the final configurations of the beam for several different load magnitudes and Fig. 2.9 compares the normalized beam tip displacements and rotations from zero-iteration shooting to numerical results from



(a) Pre-Curved Tangent Loading

(b) Pre-Curved Perpendicular Loading

Figure 2.9: The normalized displacements and rotations for the pre-curved beams subjected to (a) tangential and (b) radial point follower loads compared with the finite element results presented by Argyris[4].The dashed lines are data from [4] while the solid lines are results from the method presented herein.

Table 2.3: Helix properties given by Pai and Lee[33].

Length	$2R\pi n_c/cos(\psi_h)$
Helix Radius$[m]$	0.02
Helix Pitch Angle ψ_h	10.00°
Cross Section Radius $r[m]$	0.0010
Young's Modulus$[\frac{N}{m^2}]$	200.00×10^9
Poison's Ratio	0.32

Argyris[4]. Once again, the results show strong agreement with those presented in literature, validating the method's ability to model pre-curved beams subjected to follower loads.

2.4.1.3 Helical Cantilever Beam Subjected to Large Follower Load.

This final follower load case study illustrates the method's ability to solve three-dimensional, pre-curved, cantilever beam BVP's without iteration. A helix is loaded on the free end with a follower force that is tangent to the centerline with a magnitude of $-4.0N$. Table 2.3 gives the helix properties for this validation case. Pai and Lee used these same values in [33], where they presented a shooting method that modeled the extension and compression of a spring. In Table 2.3, r denotes the circular cross sectional radius of the beam, n_c denotes the number of complete turns in the helix, ψ_h denotes the pitch angle of the helix, and R

denotes the radius of the projection of the helix onto the XY plane. The presented shooting method requires that these helix parameters be converted into initial curvatures. Pai and Lee[33] defined the initial curvatures as functions of these parameters using

$$K_{0_1} = \frac{1}{R} cos(\psi_h) sin(\psi_h), \tag{2.21}$$

$$K_{0_2} = \psi_h', \tag{2.22}$$

$$K_{0_3} = \frac{1}{R} cos^2(\psi_h), \tag{2.23}$$

where for a helix $\psi_h' = 0$. Table 2.4 gives the numerical values used for the initial curvature and length of the beam.

Table 2.4: Beam dimensions and material properties for the helical beam used for method validation.

Length$[m]$	0.77
$K_{0_1}(m^{-1})$	8.55
$K_{0_2}(m^{-1})$	0.00
$K_{0_3}(m^{-1})$	48.49

Figures 2.10 and 2.11 display the results for the example cantilever helix subjected to the prescribed follower load. Figure 2.10 displays the deformed and undeformed configurations of the helix found using shooting with no iteration. Figure 2.11 presents quantitative results comparing the shooting method technique to a finite element solution obtained using a nonlinear analysis option available in the commercial software Abaqus. Converged results were obtained using 500 B32 elements. The results show that the curvatures, strains and displacements calculated using the shooting method are in good agreement with the finite element code; however, a small discrepancy exists between the two methods for the X-displacement. This displacement component is an order of magnitude less than the other displacement components. As a result, the discrepancy has little effect on the overall displacement magnitude comparison between the two methods. This X-displacement error results from an apparent strain coupling present in Abaqus that is not reproduced by the constitutive law used in this work. Figure 2.11b compares the strain components from the shooting method results to results obtained from Abaqus. These results show that Abaqus

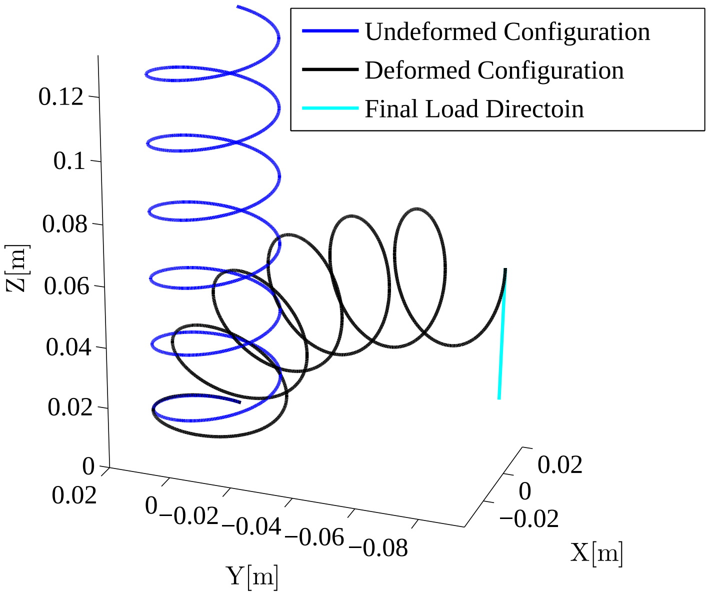
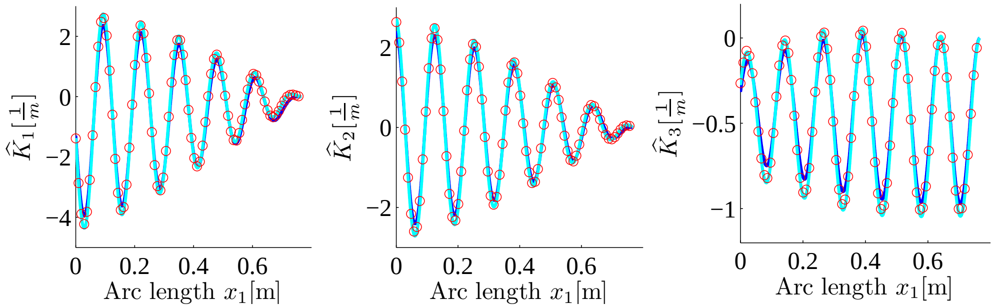


Figure 2.10: The shooting method outlined in this work can accurately calculate the nonlinear deformation of pre-curved beams in three-dimensions with only one iteration as shown by this deformed helix.

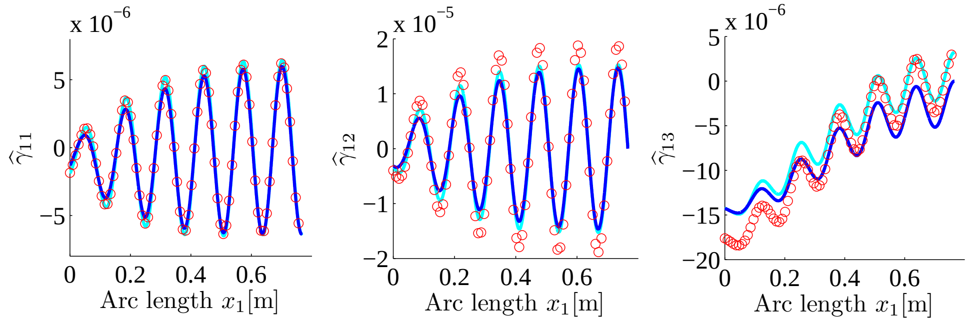
has three non-zero components of strain at the forced end of the helix, while the presented constitutive law only predicts non-zero axial strain at this end. This suggests the use of a coupled constitutive law would improve results. Note further that, if the shear strain values predicted by Abaqus at the forced end of the helix are used as the initial conditions for the shooting method, the shooting method produces notably better results - see light blue vice dark blue lines in Figure 2.11.

2.4.2 Non-Follower Load Validation

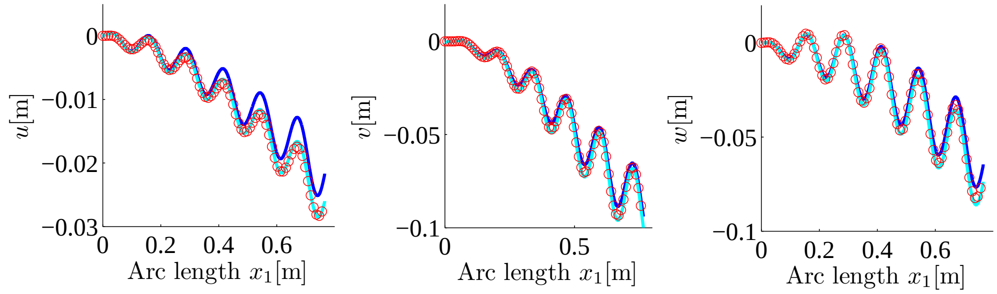
To demonstrate that the presented method does not have an apparent disadvantage when compared to traditional shooting methods, the presented shooting is used to solve standard conservative load beam BVP's. In addition, the ability of the method to obtain all solutions for a load case is demonstrated, and the stability of the post-buckled shapes are studied using a dynamic finite element method.



(a) $\widehat{\mathbf{K}}$ Component Comparison



(b) $\widehat{\boldsymbol{\gamma}}$ Component Comparison



(c) Displacement Component Comparison

Figure 2.11: The $\widehat{\mathbf{K}}$, $\widehat{\boldsymbol{\gamma}}$ and displacement components are compared to the same values from the finite element code Abaqus and show good agreement. The dark blue lines are the shooting method results, the light blue lines are the corrected shooting results and the Abaqus results are represented by the red circles.

2.4.2.1 Post-Buckling Deformation of a Straight Axially Loaded Cantilever Beam.

Studying the post-buckling behavior of a straight cantilever beam shows that the presented shooting method is able to solve beam problems with conventional non-follower loads. The results for this case are parameterized by beam dimensions and material properties. Using linear Euler-Bernoulli beam theory, the buckling loads for an axially-loaded fixed-free beam are well known:

$$F_m = \frac{(2m-1)^2\pi^2 EI_{zz}}{4L^2} \tag{2.24}$$

where $m = 1, 2, 3, ...$ denotes the buckling load number of the beam[63]. Since Euler-Bernoulli beam theory is used for these buckling cases, the intrinsic equations neglecting strain are used for sake of comparison. Figures 2.12 and 2.13 illustrate how buckled

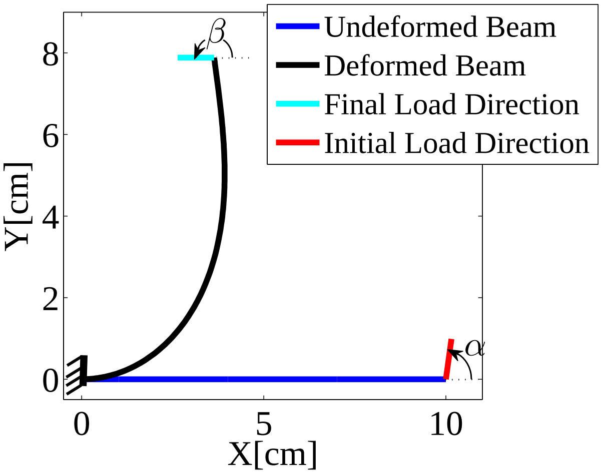
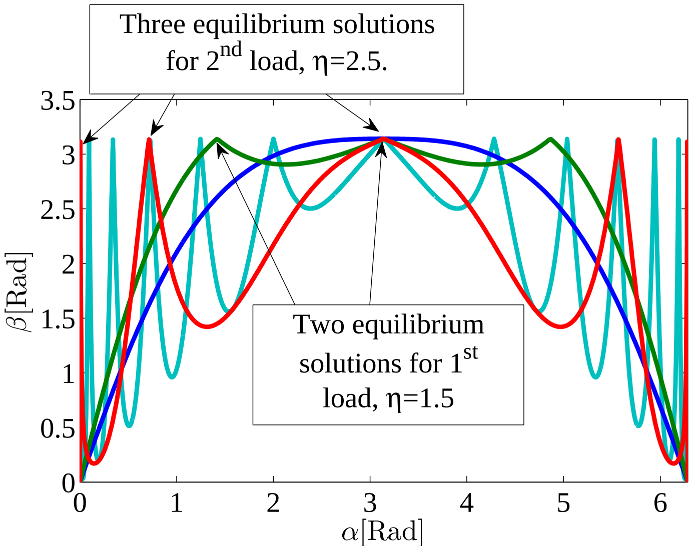


Figure 2.12: The buckled shapes are determined by varying α until the appropriate β in the deformed configuration is achieved.



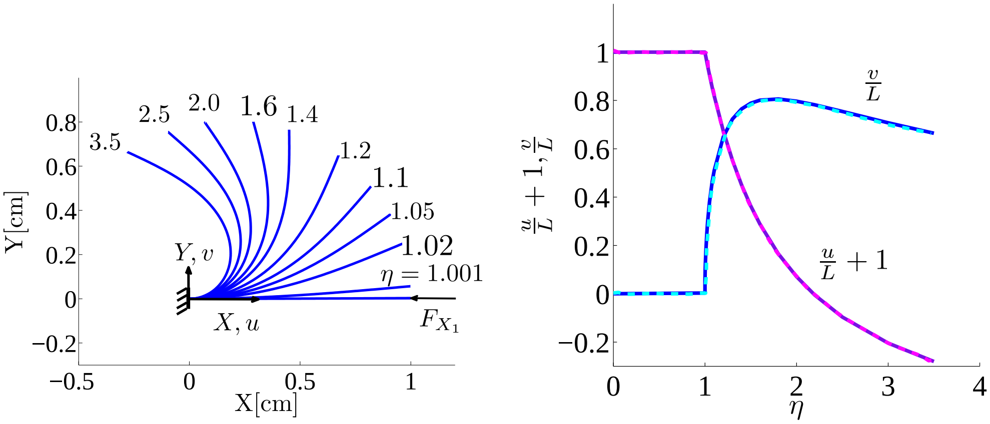
Three equilibrium solutions for 2nd load, η=2.5.

Two equilibrium solutions for 1st load, η=1.5

Figure 2.13: The deformed load angle β is plotted against the follower load angle α for three load cases showing that multiple solutions exist for $\eta > 1$. The blue, green, red and cyan lines depict when $m = 1$ and $\eta = .99$; $m = 1$ and $\eta = 1.5$; $m = 2$ and $\eta = 2.5$; and $m = 7$ and $\eta = 1.18$, respectively.

solutions are determined using the proposed shooting method. For planar buckling, the angle of the follower load relative to the local $\mathbf{B}_1$ vector is α, while β is the angle of the follower load relative to the global X-direction in the deformed configuration. The goal is to determine the post-buckled deformed configuration when $\beta = \pi$. These solutions are sought since they give the solution where the load in Ω_f is oriented in the negative X-direction, which causes axial buckling for conservative loading BVP's. Figure 2.13 presents the resultant β as a function of α for four different axial buckling loads. Also labeled are the necessary α's such that $\beta = \pi$ in Ω_f. Figure 2.13 indicates only one solution exists if $\eta < 1$, where η denotes a multiplication factor for the buckling load such that the applied axial load is $F_{X_m} = -\eta F_m$. This solution is for $\alpha = \pi$, resulting in axial compression of the beam. Furthermore, Fig. 2.13 depicts multiple solutions when $\eta > 1$ as a result of beam buckling. Note that all of the peaks in Fig. 2.13 represent a solution where the load in the deformed configuration is oriented as desired; however, all of these solutions do not necessarily yield unique configurations. The solutions with $\alpha > \pi$ give the same deformed configuration shape as the solutions found with $\alpha < \pi$, but have negative displacements. This is due to symmetry in the problem about the X axis. In fact, in the full three-dimensional problem, infinite solutions exist since the beam could buckle at any angle in the YZ plane. However, we are only concerned with the planar buckled configuration, so these solutions are suppressed.

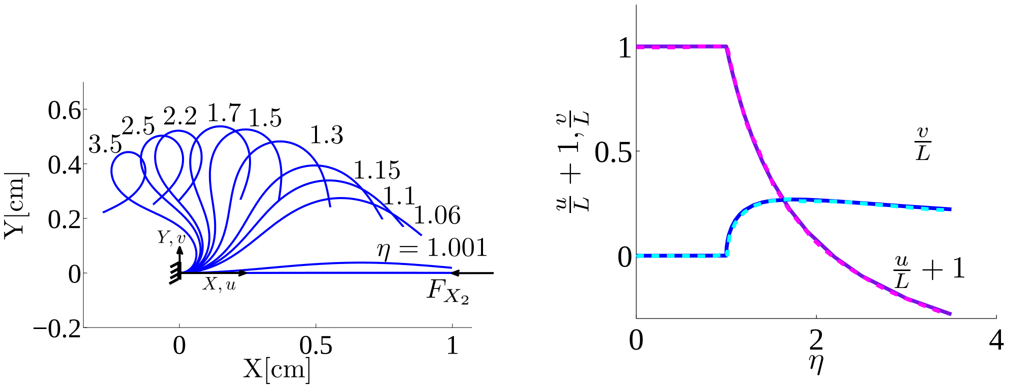
In order to investigate the post-buckling behavior of a cantilever beam, the deflection of the beam and the deformed post-buckled configurations of the beam are plotted as a function of η. Figure 2.14 shows the results from this study for F_{X_1}, and compares the shooting results with those obtained by Pai and Palazotto in [46]. As the load increases past F_1, or past the critical buckling load, more than one possible equilibrium solution satisfies the BVP. For the first buckling load, this includes the unstable straight configuration and a stable buckled shape. Figure 2.13 illustrates that for each buckling load, $m + 1$ solutions exist, which consist of m buckled mode shapes and the axially loaded unstable configuration. Figures 2.15a and 2.16a show a set of possible post-buckled configurations for F_{X_2} and F_{X_3} calculated using shooting, which correspond to the 2^{nd} and 3^{rd} modes. Figures 2.15b and



(a) Post-Buckled Deformation for $m = 1$

(b) Displacements versus Load Factor

Figure 2.14: The (a) post-buckled shape of the beam is shown with $m = 1$ and increasing load factor η along with (b) the normalized displacements results compared with work done by Pai [46].

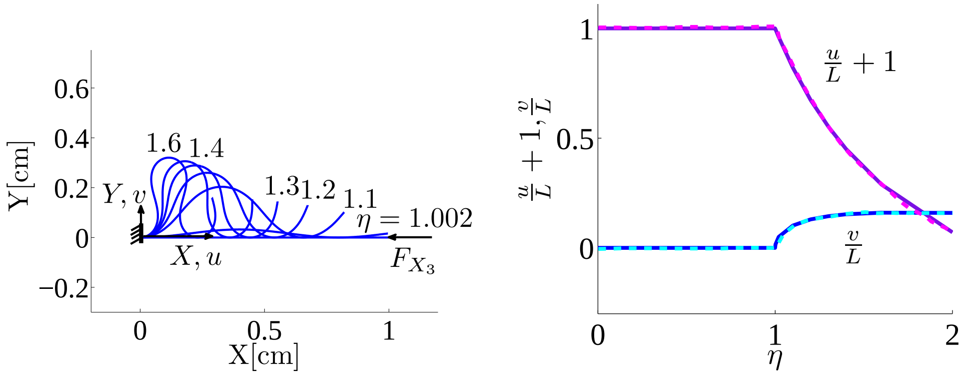


(a) Post Buckling Deformation for $m = 2$

(b) Displacements versus Load Factor

Figure 2.15: The (a) post-buckled shape for the 2^{nd} mode shape is studied with increasing load factor η and (b) the normalized displacements compare well to those presented by [46]. The solid lines are results from shooting and the dashed lines are results from [46].

2.16b show the normalized displacements of the beam end and compare the shooting results with results available in [46]. The studied post-buckled configurations compare favorably to literature results. The two deformed shapes shown in Figs. 2.15a and 2.16a are only a single possible buckling mode for the 2^{nd} and 3^{rd} buckling loads. The multiple solutions for the second buckling load are investigated further. Figure 2.17 depicts another equilibrium solution for the second buckling load other than the one depicted in Fig. 2.15a. This alternate equilibrium solution past the 2^{nd} buckling load with $\eta = 2.5$ is found when $\alpha =$



(a) Post Buckling Deformation for $m = 3$ (b) Displacements versus Load Factor

Figure 2.16: The (a) post-buckled shape for the 3^{rd} mode shape is studied with increasing load factor η and (b) the normalized displacements compare well to those presented by [46]. The solid lines are results from shooting and the dashed lines are results from [46].

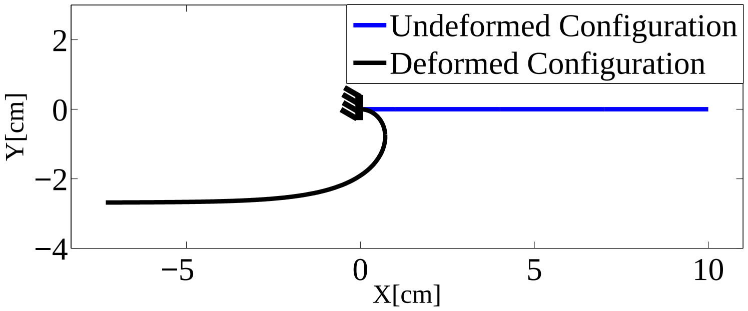


Figure 2.17: An alternate equilibrium solution for the 2^{nd} buckling load with $\eta = 2.5$ is found when $\alpha = 6.28$ radians and has a much different shape than the solution when $\alpha = 5.57$ radians.

6.28 radians. This shape differs from the solution illustrated in Fig. 2.15, which is found at $\alpha = 5.57$ radians. The configuration depicted in Fig. 2.17 is actually the first buckled mode shape with larger deflection due to the increased load. It is of practical interest which, if any, of these solutions is stable.

As described previously, the stability of the second buckling load mode shapes is tested with the use of a special-purpose explicit finite element code [32] developed from the same intrinsic equations. The two equilibrium positions and loads are ported to the finite element code and allowed to simulate for an extended period of time after the application of a small perturbation. Figure 2.18 presents the displacement of the beam end as a function of time

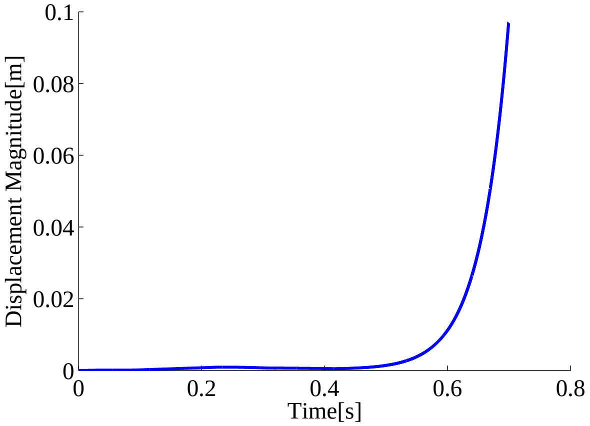


Figure 2.18: The instability of the 2^{nd} post-buckled mode shape is demonstrated by the exponential growth of the forced-end displacement magnitude with respect to time. This post-buckled configuration is for $m = 2$, $\eta = 2.5$ and $\alpha = 5.57$ radians.

for the mode shape shown in Fig. 2.15. These results illustrate instability as is evident by the exponential growth in the end displacement of the beam.

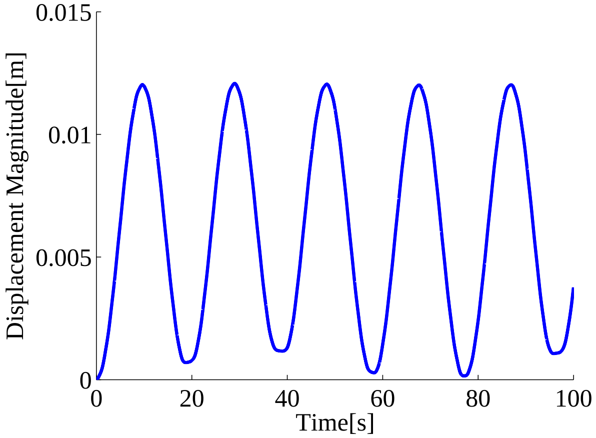
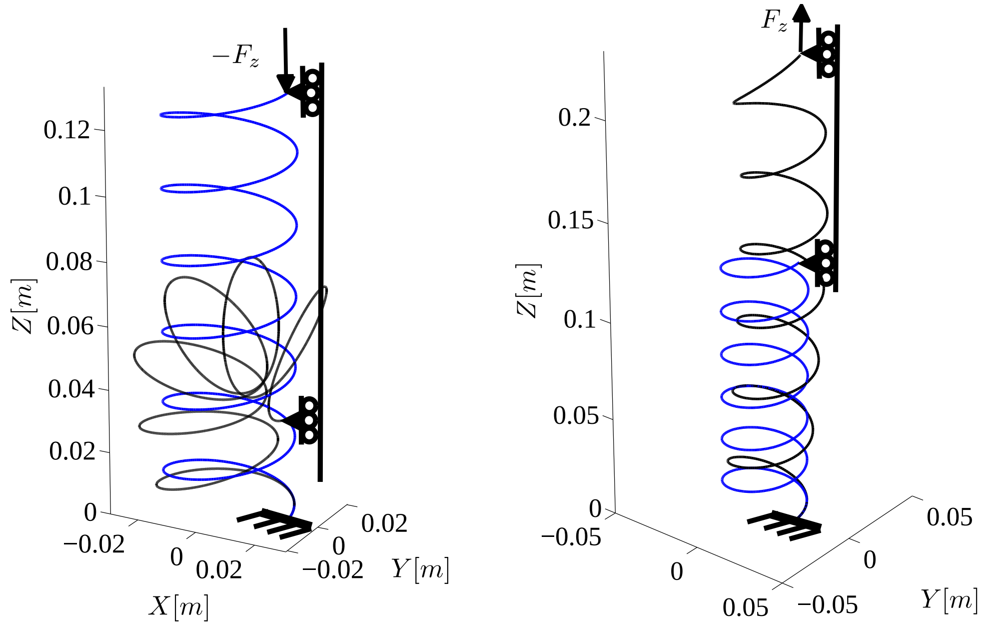


Figure 2.19: The stability of a post-buckled equilibrium solution is demonstrated by the steady-state oscillation of the forced-end displacement magnitude with respect to time. The post-buckle configuration is for $m = 2$, $\eta = 2.5$ and $\alpha = 6.28$ radians.

In contrast to this unstable solution, the buckled mode shape presented in Fig. 2.17 is a stable solution as verified by Fig. 2.19. Figure 2.19 displays the displacement of the end of this case away from the buckled solution as a function of time. Unlike the unstable case discussed earlier, the displacement oscillates around the stable equilibrium value at a magnitude on the order of the applied perturbation. This oscillation is constant for a long period of time, indicating dynamic stability around the equilibrium point.

2.4.2.2 Deformation of Helical Beam Subjected to Compression and Extension Loads.

The shooting method is also capable of modeling more complex loading configurations with three-dimensional geometry. In the validation case presented next, a prescribed Z-displacement applied to the free end of a fixed-free helical spring allows for the creation of compression and extension force-displacement curves. Tables 2.3 and 2.4 define the helix properties and dimensions, which are the same values used for the follower loaded helix. Pai and Lee[45] also studied this spring compression and extension case; however, a different solution is found with the developed method. The presented method's results are compared to results obtained from an Abaqus model using nonlinear analysis and 500 B32 beam elements. The force-displacement curves are created with the implementation of a multivariable minimization algorithm that uses shooting to find the correct load magnitude and direction at the end of the helix that results in the desired displacement. For this helix case with a prescribed Z-displacement, only the three force components at the helix free



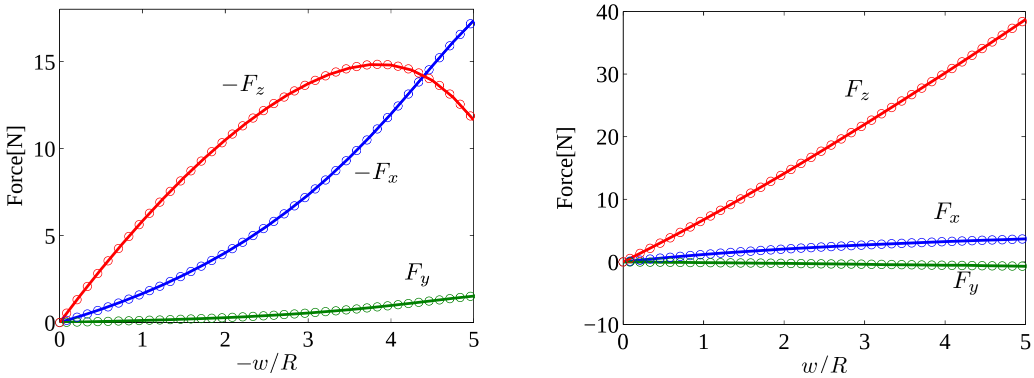
(a) $w = -5R$

(b) $w = 5R$

Figure 2.20: The deformed (black lines) and undeformed (blue lines) configurations of the helical spring for the largest displacements in (a) compression and (b) extension.

end govern the possible system response. Once the minimization algorithm determines the correct values for these forces, they are projected onto the global basis to give the global reaction forces and the force in the Z direction necessary to compress or extend the spring to the desired displacement. Displacements up to $5R$ and $-5R$ are imposed on the spring.

Fig. 2.20 illustrates the resulting deformed configurations along with the undeformed configurations for the largest imposed displacements of $-5R$ for compression, and $5R$ for extension. Figure 2.21 compares quantitative results for the force-displacement curves obtained from shooting to similar results obtained using Abaqus. Strong agreement is noted in all comparisons, validating the ability of the presented shooting method to generate solutions for general three-dimensional beam BVP's undergoing large deformation.



(a) Compression Force-Displacement Curve

(b) Extension Force-Displacement Curve

Figure 2.21: The force-displacement curves for the helical spring in (a) compression and (b) extension. The solid lines are results from shooting and the circles are results from Abaqus.

2.5 Conclusions

This chapter develops a shooting method that provides numerical solutions to the intrinsic equations governing beam equilibrium. Test cases show that the presented method avoids iteration for pre-curved cantilever beams subjected to distributed and/or point follower loads. In addition, a general approach for finding all solutions to static beam problems with conservative loading is described and validated with a number of test cases. A procedure is described which guarantees the determination of *all* unstable and stable solutions to beam buckling problems. Solution stability is assessed using a dynamic finite element code based on the same intrinsic beam equations. Due to the method avoiding iteration in follower load problems, it should be attractive for use in model-based control where the solution of a system's response to follower loads is needed in a computationally-efficient manner.

CHAPTER III

HIERARCHICAL MULTISCALE CONTINUUM MODEL

3.1 Introduction

This chapter covers the development and validation of a hierarchical multiscale continuum approach for modeling prototypical α-helical protein segments using the intrinsic beam formulation. Hierarchical continuum modeling of biomolecules reduces the computational expense of molecular simulations by substantially reducing the degrees of freedom of the system and increasing the allowable time step. Representing a discrete atomistic system as an elastic continuum accomplishes the reduction of the degrees of freedom. For the intrinsic formulation, a linear elastic helical beam represents the protein, while the appropriate curvature stiffnesses must be found. To determine the stiffnesses, the shooting method described in Chapter 2 calculates the deformed configuration of the beam model due to external loading. An optimization algorithm varies the intrinsic stiffness terms until the deformed beam configuration matches the deformed configuration of a simplified protein modeled with the LAMMPS MD package. After development and validation of the hierarchical continuum model for simulating α-helical protein segments, the method can be extended for use as a general protein segment modeling tool.

For method validation, the hierarchical multiscale continuum model predicts static and dynamic results for various load cases. The shooting method determines the predicted static deformation of the simplified protein for two load cases which compare well to quasistatic LAMMPS MD solutions. For dynamic simulation validation, a finite element program [32] developed for the intrinsic beam formulation simulates the response of the hierarchical continuum model to impulse loading. Four test cases simulated in the intrinsic FEA code compare well to LAMMPS MD dynamics simulations. These prediction cases demonstrate that the developed hierarchical multiscale continuum model exhibits similar stiffness and dynamic characteristics as the LAMMPS MD model.

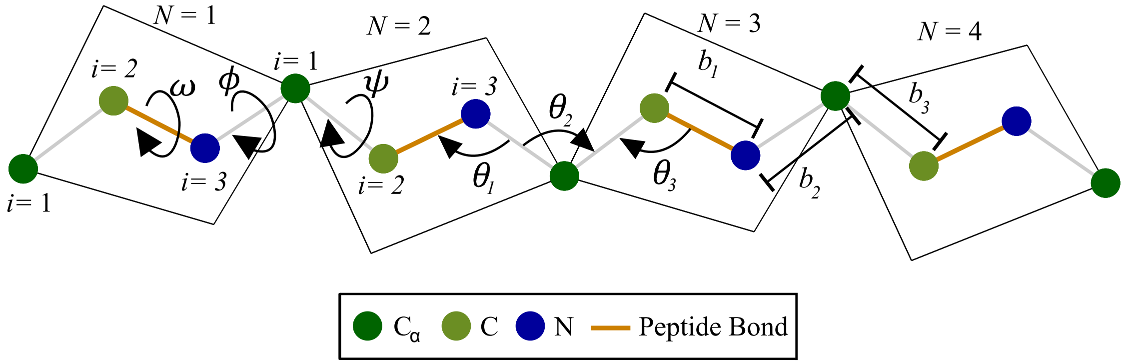


Figure 3.1: The simplified protein model has only three types of bond angles, bond lengths and dihedral angles in its potential function.

3.2 Methods

This section introduces the simplified model that represents the polypeptide backbone, along with relationships which calculate intrinsic formulation curvatures from ϕ and ψ. In addition, the LAMMPS simulations and the optimization procedure for determining the equivalent intrinsic stiffness are outlined.

3.2.1 Simplified Protein Backbone Model

The hierarchical continuum model utilizes a simplified polypeptide model. The simplified model includes only the backbone atoms in order to reduce complexity during model development. Similar simplified backbone models appear in literature for use in coarse-grained protein simulations, and justify the adoption of such a model ([19], [64]). This simplified model is typically studied since the backbone atom locations describe the protein tertiary shape while allowing reduced degrees of freedom. Using corrected potentials causes the simplified model to behave similarly to all-atom models. Furthermore, the overall procedures used in development of the hierarchical continuum model are general, and easily adapted to include all atoms in the protein. Figure 3.1 illustrates the simplified polypeptide, which contains only backbone atoms. As seen in Fig. 3.1, the use of this simplified model reduces the amount of bonded potential parameters to the 3 dihedral angles ϕ, ψ and ω; 3 bond lengths b_1, b_2 and b_3; and 3 bond angles θ_1, θ_2, and θ_3.

Further simplification includes the removal of nonbonded terms from the potential, and the use of harmonic dihedral angle potential terms. The resulting simplified potential function is

$$
\begin{aligned}
U_s(\mathbf{P}_i^N) = \sum_{bonds} K_{b_i}(b_i - b_{i_0})^2 + \sum_{angles} K_{\theta_i}(\theta_i - \theta_{i_0})^2 \\
+ K_\phi(\phi - \phi_0)^2 + K_\psi(\psi - \psi_0)^2 + K_\omega(\omega - \omega_0)^2,
\end{aligned}
\tag{3.1}
$$

where $\mathbf{P}_i^N$ denotes the backbone atom positions in an arbitrary global reference frame to be described in detail later in the section, and the stiffness and equilibrium parameters are given in Table 3.1. Note that the rotation equilibrium values are given in degrees, while the stiffnesses have units requiring the use of radians. These units follow the convention set by the CHARMM potential field [36], which is a commonly used potential field for protein modeling. In fact, the equilibrium and stiffness values for the bond lengths and

Table 3.1: Stiffness constants and equilibrium values used in the potential function given by Eq. (3.1).

Stiffness Constants	**Equilibrium Values**
$K_{b_1} = 370 \left[\frac{\text{kcal}}{\text{mole Å}^2}\right]$	$b_{1_0} = 1.490[\text{Å}]$
$K_{b_2} = 320 \left[\frac{\text{kcal}}{\text{mole Å}^2}\right]$	$b_{2_0} = 1.430[\text{Å}]$
$K_{b_3} = 250 \left[\frac{\text{kcal}}{\text{mole Å}^2}\right]$	$b_{3_0} = 1.490[\text{Å}]$
$K_{\theta_1} = 50.0 \left[\frac{\text{kcal}}{\text{mole rad}^2}\right]$	$\theta_{1_0} = 107.0°$
$K_{\theta_2} = 80.0 \left[\frac{\text{kcal}}{\text{mole rad}^2}\right]$	$\theta_{2_0} = 116.5°$
$K_{\theta_3} = 50.0 \left[\frac{\text{kcal}}{\text{mole rad}^2}\right]$	$\theta_{3_0} = 120.0°$
$K_\omega = 20.5 \left[\frac{\text{kcal}}{\text{mole rad}^2}\right]$	$\omega_0 = 180.0°$
$K_\phi = 2.00 \left[\frac{\text{kcal}}{\text{mole rad}^2}\right]$	$\phi_0 = -58.00°$
$K_\psi = 6.00 \left[\frac{\text{kcal}}{\text{mole rad}^2}\right]$	$\psi_0 = -47.00°$

bond angles in Table 3.1 are the same as those presented by CHARMM; however, the ω, ϕ and ψ stiffnesses have been increased by an order of magnitude to favor an α-helical shape.

3.2.2 Dihedral Angle to Curvature Mapping

The developed hierarchical multiscale continuum model can be used to calculate C_α atom positions and ϕ and ψ dihedral angles. The continuum beam modeling the protein passes through each C_α atom at equal intervals along the centerline in Ω_0 such that material points define the position of each atom. Since material points locate the C_α atoms in the continuum, the position of each C_α atom is known given the deformed configuration's curvature and strain. Relationships that map protein ϕ and ψ dihedral angles to curvature and torsion define the curvatures for the helical beam representing the protein. Inverting the dihedral to curvature mapping defines the dihedral angles at a given point on the beam centerline based on the beam's curvature values. As a result, the work presented herein retains much of the atomic resolution of the modeled protein.

The dihedral to curvature mapping requires three assumptions: the protein segment consists of rigid peptide planes, contains repeating dihedral angles sets and exhibits a constant curvature helical shape. Rigid peptide planes allow the calculation of backbone atom positions for N peptide planes with N $[\phi, \psi]$ dihedral sets as independent variables, since the ω dihedral, bond angles and bond lengths remain constant. To develop a relationship mapping the dihedrals to curvatures at an arbitrary point, the repeating dihedral angle assumption must hold. For example, four C_α atoms mathematically define the helical curve. Since a polypeptide with four C_α atoms contains two $[\phi, \psi]$ dihedral sets, the dihedrals must repeat to allow the direct mapping of ϕ and ψ to curvatures such that $[\phi,\ \psi] \rightarrow \mathbf{K}$ for the helical beam centerline. Otherwise, $[\phi_1,\ \psi_1,\ \phi_2,\ \psi_2]$ define $\mathbf{K}$ when the dihedrals do not repeat. Additionally, the constant curvature assumption constrains the solution to be a helical shape. When inverting the relationship to get dihedrals from curvatures, the method is more general since locally to a point of interest the curvatures are constant, but the dihedrals are not necessarily repeating. As a result of the constant curvatures, dihedral angle estimates can be determined at any point of a deformed beam.

Sugeta and Miyazawa [62] develop the basis for this relationship in their paper outlining the calculation of helical parameters for polymer chains. In this paper, Sugeta and Miyazawa introduce a procedure for calculating the atom positions of a repeating polymer using the protein structure terms defined in Chapter 1. From their work, the ϕ and ψ dihedrals define the position of the simplified protein backbone atoms with

$$\mathbf{P}_i^N(\phi,\psi) = \mathbf{P}_1^1 + \mathbf{P}_2^1 + \mathbf{P}_3^1 + \mathbf{B} + \sum_{m=2}^{N} \left\{ [\mathbf{A}_{12}]\,[\mathbf{A}_{23}]\,[\mathbf{A}_{34}]\,\mathbf{P}_i^{m-1} + \mathbf{B} \right\} \tag{3.2}$$

where

$$\mathbf{A}_{12} = \begin{bmatrix} -\cos\theta_{3_0} & -\sin\theta_{3_0} & 0 \\ \sin\theta_{3_0}\cos\psi & -\cos\theta_{3_0}\cos\psi & -\sin\psi \\ \sin\theta_{3_0}\sin\psi & -\cos\theta_{3_0}\sin\psi & \cos\psi \end{bmatrix}, \tag{3.3}$$

$$\mathbf{A}_{23} = \begin{bmatrix} -\cos\theta_{1_0} & -\sin\theta_{1_0} & 0 \\ \sin\theta_{1_0}\cos\omega_0 & -\cos\theta_{1_0}\cos\omega_0 & -\sin\omega_0 \\ \sin\theta_{1_0}\sin\omega_0 & -\cos\theta_{1_0}\sin\omega_0 & \cos\omega_0 \end{bmatrix}, \tag{3.4}$$

$$\mathbf{A}_{34} = \begin{bmatrix} -\cos\theta_{2_0} & -\sin\theta_{2_0} & 0 \\ \sin\theta_{2_0}\cos\phi & -\cos\theta_{2_0}\cos\phi & -\sin\phi \\ \sin\theta_{2_0}\sin\phi & -\cos\theta_{2_0}\sin\phi & \cos\phi \end{bmatrix}, \tag{3.5}$$

$$\mathbf{P}_1^1 = \left\{ \begin{matrix} P^1_{1_{XYZ}} & P^1_{2_{XYZ}} & P^1_{3_{XYZ}} \end{matrix} \right\}^T, \tag{3.6}$$

$$\mathbf{P}_2^1 = \left\{ \begin{matrix} b_{3_0} & 0 & 0 \end{matrix} \right\}^T, \tag{3.7}$$

$$\mathbf{P}_3^1 = \mathbf{P}_2^1 + [\mathbf{A}_{12}] \left\{ \begin{matrix} b_{1_0} & 0 & 0 \end{matrix} \right\}^T, \tag{3.8}$$

and

$$\mathbf{B} = \mathbf{P}_2^1 + \mathbf{P}_3^1 + [\mathbf{A}_{12}]\,[\mathbf{A}_{23}] \left\{ \begin{matrix} b_{2_0} & 0 & 0 \end{matrix} \right\}^T. \tag{3.9}$$

In the preceding equations, $\mathbf{P}_1^1$, $\mathbf{P}_2^1$, $\mathbf{P}_3^1$ denote the location of the first, second and third atoms in the protein, respectively. The matrices $\mathbf{A}_{12}$, $\mathbf{A}_{23}$, $\mathbf{A}_{34}$ are rotation matrices that apply the dihedral and bond angle rotations between atoms along the polypeptide backbone. The $\mathbf{B}$ vector gives the relative distance between two consecutive C$_\alpha$ atoms and is discussed in detail later in the section. Combining Eqs. (3.2)-(3.9) gives the location of the i^{th} atom in the N^{th} peptide plane with respect to an arbitrary global coordinate system. The atom coordinates $\mathbf{P}_1^1$ arbitrarily define the initial starting point of the protein in the global frame, and the first bond length is oriented in the global X direction.

In addition, Sugeta and Miyazawa demonstrate how to calculate the helical terms d, θ, and ρ from these atom positions; where d represents the translation along the helical axis between consecutive C_α atoms, θ denotes the rotation about the helical axis between consecutive C_α atoms, and ρ denotes the radius of the resulting helix. Figure 3.2 depicts the helical terms for the simplified backbone and a resulting helix centerline. The following equations implicitly define these helical terms:

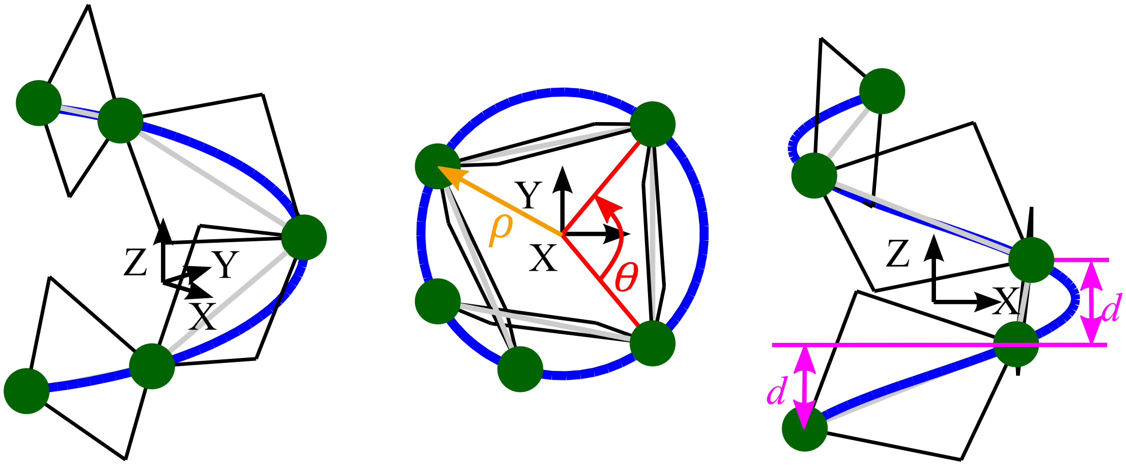


Figure 3.2: Sugeta and Miyazawa outline the procedure to calculate the helix parameters d, θ, and ρ using the position of the C_α atoms. The blue line depicts the resulting helix defined by these terms and the helix axis aligns with the global Z direction in this illustration.

$$\cos(\theta) = (\mathbf{C} \cdot \mathbf{C}')/C^2, \tag{3.10}$$

$$\rho(1 - \cos(\theta)) = \frac{C}{2}, \tag{3.11}$$

$$d^2 + 2\rho(1 - \cos(\theta)) = B^2, \tag{3.12}$$

and

$$d \sin(\theta) = \mathbf{B} \cdot (\mathbf{C} \times \mathbf{C}')/C^2, \tag{3.13}$$

where $\mathbf{B}$ denotes the relative position vector from the 2^{nd} C_α atom to the 3^{rd} C_α atom as decribed earlier, $\mathbf{C}$ denotes the vector that passes from the helical axis through the 2^{nd} C_α

atom, $\mathbf{B}'$ denotes the relative position vector from the 1^{st} C_α atom to the 2^{nd} C_α atom, $\mathbf{C}'$ denotes the vector that passes from the helical axis through the 3^{rd} C_α atom, and $\mathbf{B}''$ denotes the relative position vector from the 3^{rd} C_α atom to the 4^{th} C_α atom. The letters B and C denote the magnitudes of the $\mathbf{B}$ and $\mathbf{C}$ vectors, respectively. Figure 3.3 illustrates these vectors on the four C_α polypeptide used to define the mapping relationship. The helical basis xyz is oriented such that z points down the helix axis, x points toward the 2^{nd} C_α atom and y completes the orthonormal set. The following equations define the helix basis unit vectors with respect to the global basis:

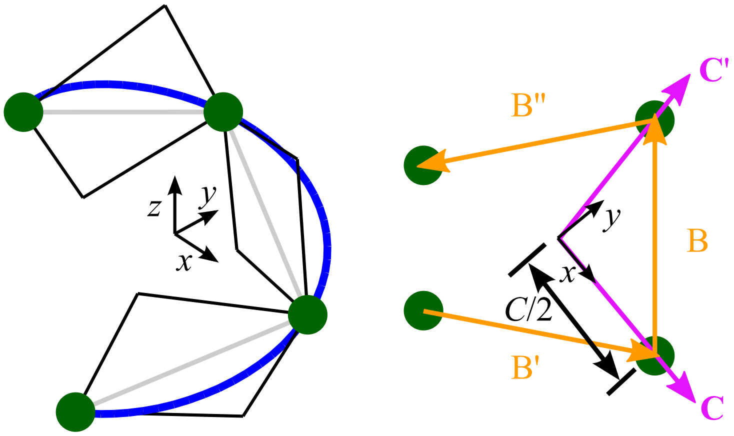


Figure 3.3: The vectors used to calculate the helix parameters include $\mathbf{C}$, $\mathbf{C}'$, $\mathbf{B}$, $\mathbf{B}'$, and $\mathbf{B}''$.

$$\widehat{\mathbf{x}} = \frac{\mathbf{C}}{C}, \tag{3.14}$$

$$\widehat{\mathbf{z}} = \frac{\mathbf{C} \times \mathbf{C}'}{C^2 \sin\theta}, \tag{3.15}$$

and

$$\widehat{\mathbf{y}} = \widehat{\mathbf{z}} \times \widehat{\mathbf{x}}. \tag{3.16}$$

The terms from Eqs. (3.10)-(3.13) define the helical curve relative to the xyz frame using

$$\mathbf{R}_{xyz}(x_1) = \left\{ \begin{array}{c} \rho\cos\left(\frac{x_1}{\sqrt{\rho^2+(d/\theta)^2}}\right) \\ \rho\sin\left(\frac{x_1}{\sqrt{\rho^2+(d/\theta)^2}}\right) \\ \frac{d}{\theta\sqrt{\rho^2+(d/\theta)^2}}x_1 \end{array} \right\}, \tag{3.17}$$

which is valid for $-\theta\sqrt{\rho^2+(d/\theta)^2} \leq x_1 \leq 3\theta\sqrt{\rho^2+(d/\theta)^2}$.

The Frenet-Serret curvature and twist are used to relate the Sugeta and Miyazawa helical terms to the intrinsic formulation curvatures. Similar to the intrinsic formulation, the Frenet-Serret equations define the orientation of a three-dimensional curve using differential geometry,

$$\frac{d\mathbf{R}}{dx_1} = \mathbf{t} \tag{3.18}$$

$$\frac{d\mathbf{t}}{dx_1} = \kappa\mathbf{n} \tag{3.19}$$

$$\frac{d\mathbf{n}}{dx_1} = \tau\mathbf{b} - \kappa\mathbf{t} \tag{3.20}$$

$$\frac{d\mathbf{b}}{dx_1} = -\tau\mathbf{n} \tag{3.21}$$

where $\mathbf{t}$, $\mathbf{n}$ and $\mathbf{b}$ denote the tangent, normal and binormal vectors defining the Frenet-Serret orthonormal basis. The variables κ and τ denote the curvature and twist of the curve centerline. Using Eqs. (3.17)-(3.21), the helix terms presented by Sugeta and Miyazawa relate to the Frenet-Serret helix terms such that

$$\kappa^2 = \frac{\rho^2}{\rho^2+(d/\theta)^2} \tag{3.22}$$

$$\tau^2 = \frac{(d/\theta)^2}{\rho^2+(d/\theta)^2}. \tag{3.23}$$

Leamy [32] relates the intrinsic curvatures to the Frenet-Serret curvature and twist such that

$$\kappa = \sqrt{K_2^2+K_3^2} \tag{3.24}$$

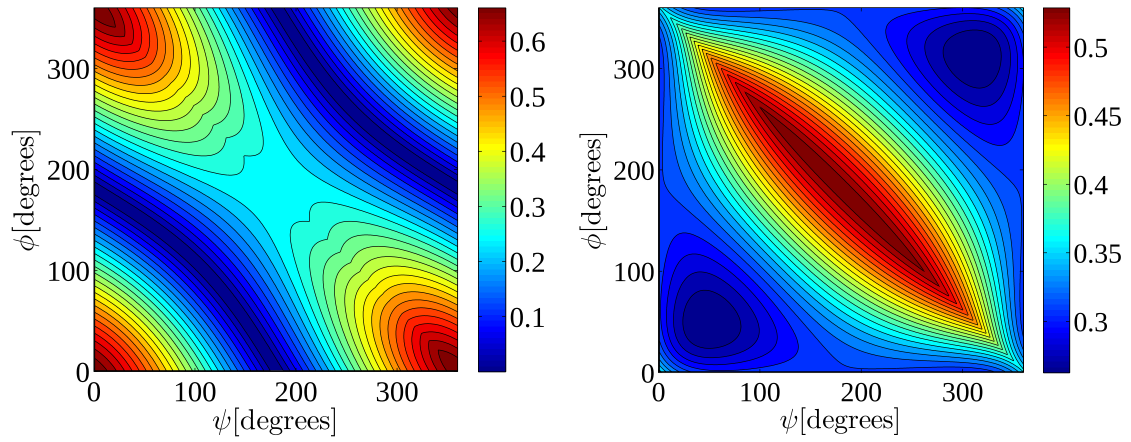
and

$$\tau = K_1 + \frac{1}{K_2^2+K_3^2}\left(K_2'K_3 - K_3'K_2\right). \tag{3.25}$$

Since a helix exhibits constant curvature Eq. (3.25) simplifies to $\tau = K_1$. In addition, aligning the intrinsic basis $\mathbf{B}_i$ with the Frenet-Serret frame such that $\mathbf{B}_2 = \mathbf{n}$ and $\mathbf{B}_3 = \mathbf{b}$ results in $K_2 = 0$. Therefore, the protein dihedrals define the intrinsic curvatures such that

$$K_1(\phi,\psi) = \frac{d}{\theta\sqrt{\rho^2+(d/\theta)^2}}, \tag{3.26}$$

$$K_2(\phi,\psi) = 0, \tag{3.27}$$



(a) K_1

(b) K_3

Figure 3.4: The contour plots give (a) K_1 and (b) K_3 as a function of ϕ and ψ using Eqs. (3.2)- (3.28).

and

$$K_3(\phi, \psi) = \frac{\rho}{\sqrt{\rho^2 + (d/\theta^2)}}. \tag{3.28}$$

Figure 3.4 displays the contour plots obtained using Eqs. (3.26)-(3.28), which map ϕ and ψ to $\mathbf{K}$. These contour plots define the intrinsic curvatures needed for a helical beam to represent a polypeptide consisting of repeating ϕ and ψ dihedral angles. In the following validation cases, they are used to define the Ω_0 beam representation of the simplified protein and can be used to calculate the dihedral angles of the polypeptide. Similarly, these equations define the intrinsic curvatures needed to represent the simplified backbone protein segment with a helical beam for the concurrent multiscale modeling technique described in Chapter 4.

3.2.3 Stiffness Determination with Shooting and Optimization

An optimization algorithm calculates the equivalent protein stiffness $\mathbf{D}_{eq}$ using the shooting method developed for the intrinsic beam formulation. This optimization algorithm intelligently varies the intrinsic stiffness values used in the shooting method until deformed configurations obtained with the shooting method match those from quasistatic LAMMPS MD simulations. Neglecting strain reduces the number of optimization parameters and the equivalent stiffness takes the following form:

$$\mathbf{D}_{eq} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix}, \tag{3.29}$$

where $\mathbf{D}_{eq}$ denotes the intrinsic curvature stiffness given in Eq. (2.13) from Chapter 2. The optimization minimizes the root mean squared deviation (RMSD) between the C$_\alpha$ atom positions calculated with the shooting method and the LAMMPS MD simulations. The RMSD is calculated with

$$\text{RSMD} = \sqrt{\frac{1}{N}\sum_{m=1}^{N+1} ||\mathbf{P}_{1_S}^m - \mathbf{P}_{1_L}^m||^2} \tag{3.30}$$

where $\mathbf{P}_{1_S}^m$ denotes the global C$_\alpha$ positions calculated with the shooting method and $\mathbf{P}_{1_L}^m$ denotes the global C$_\alpha$ positions calculated with LAMMPS. Similarly, a normalized percent error comparing the shooting results to LAMMPS is an alternative metric for monitoring solution accuracy defined by

$$\%\ \text{Error} = 100\frac{||\mathbf{u}_{1_S}^m - \mathbf{u}_{1_L}^m||}{||\mathbf{u}_{1_L}^m||}, \tag{3.31}$$

where $\mathbf{u}_{1_S}^m$ and $\mathbf{u}_{1_L}^m$ denote the global C$_\alpha$ displacements calculated with the shooting method and LAMMPS, respectively. The optimization procedure calculates $\mathbf{D}_{eq}$ for multiple follower load calibration cases, and averaging these stiffness results yields a final $\mathbf{D}_{eq}^f$, which represents the protein stiffness. The calibration cases have beam subjected to follower loads since shooting can solve follower load problems with zero iteration, thus allowing a more efficient optimization process.

3.2.4 Mass Properties

The mass properties for the intrinsic continuum representation of the protein must be calculated from the discrete protein model for use in dynamic simulations. The four mass terms considered in the intrinsic beam formulation include the mass per unit length μ and the mass moments of inertia per unit length i_1, i_2 and i_3 where $i_1 \equiv i_2 + i_3$. An α-helical polypeptide with a helical beam intrinsic representation is used for calculating these mass properties over a single peptide plane. Figure 3.5 illustrates a single peptide plane from the α-helical polypeptide with the intrinsic centerline in blue. The following equation gives the mass per unit length of the beam:

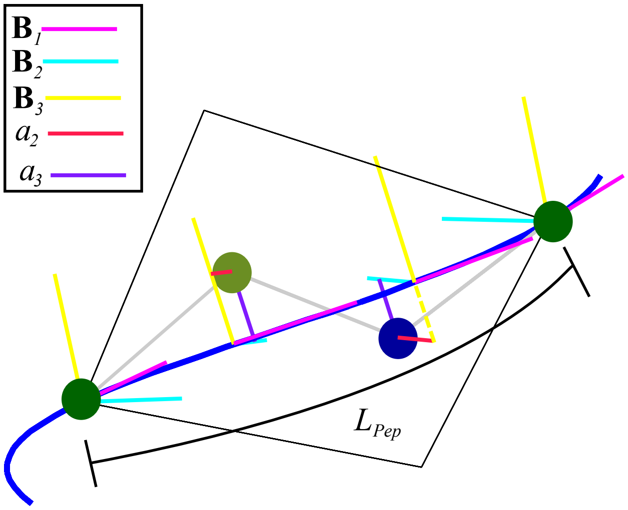


Figure 3.5: The intrinsic beam mass properties are calculated using a_2, a_3 and the mass of the first three atoms in the peptide plane.

$$\mu = \frac{\sum_{i=1}^{3} m_i}{L_{Pep}}, \tag{3.32}$$

where L_{Pep} denotes the length of the helix segment through one peptide plane, and m_i denotes the atomic masses of the three atoms contained in the peptide plane. The length of the helix through a peptide plane is calculated with

$$L_{Pep} = \theta\sqrt{\rho^2 + (d/\theta)^2}. \tag{3.33}$$

The mass for one peptide plane only includes the three atomic masses of the first C$_\alpha$, the C, and the N atoms. The second C$_\alpha$ mass does not contribute to the mass of the current peptide plane since it will be accounted for by the next peptide plane. The moments of inertia per unit length are calculated from

$$i_2 = \frac{\sum_{i=1}^{3} (a_2^i)^2 m_i}{L_{Pep}} \tag{3.34}$$

and

$$i_3 = \frac{\sum_{i=1}^{3} (a_3^i)^2 m_i}{L_{Pep}} \tag{3.35}$$

where a_2^i and a_3^i denote the distances from the centerline in the local $\mathbf{B}_2$ direction and $\mathbf{B}_3$ directions for the i^{th} atom.

Calculating a_2^i and a_3^i requires the numerical solution of a transcendental equation. The transcendental equation is required for determining the x_1 centerline location for each atom. A local basis $\mathbf{B}_i$ exists at the same x_1 centerline location as each atom such that the atom is located in a local $\mathbf{B}_2$-$\mathbf{B}_3$ plane. As a result, the inner product of the local vector position of the atom with the normal of the corresponding local $\mathbf{B}_2$-$\mathbf{B}_3$ plane must be zero. Section 3.2.2 supplies the equations for calculating an undeformed protein configuration such that Eq. (3.2) defines the atomic positions, Eq. (3.17) defines the helix representation, and Eqs. (3.18)-(3.21) define the local basis in the xyz coordinate system. Subtracting the centerline position of the helix $\mathbf{R}_{xyz}(dx_1)$ from the position of the corresponding atom of interest gives the vector from the $\mathbf{B}_i$ basis center to the atom of interest in the xyz basis,

$$\mathbf{p}_{xyz} = \mathbf{P}_{xyz} - \mathbf{R}_{xyz}(dx_1). \tag{3.36}$$

The normal of the $\mathbf{B}_2$-$\mathbf{B}_3$ plane, $\mathbf{B}_1$, aligns with the tangent when the helix is undeformed such that

$$\mathbf{B}_{1_{xyz}}(x_1) = \left\{ \begin{array}{c} -\frac{\rho}{\sqrt{\rho^2+(d/\theta)^2}} \sin\left(\frac{x_1}{\sqrt{\rho^2+(d/\theta)^2}}\right) \\ \frac{\rho}{\sqrt{\rho^2+(d/\theta)^2}} \cos\left(\frac{x_1}{\sqrt{\rho^2+(d/\theta)^2}}\right) \\ \frac{d}{\theta\sqrt{\rho^2+(d/\theta)^2}} \end{array} \right\}. \tag{3.37}$$

Setting the inner product of Eqs. (3.36) and (3.37) to zero results in

$$\begin{gathered} -\frac{\rho}{\nu}\sin\left(\frac{x_1}{\nu}\right)\left[p_x - \rho\cos\left(\frac{x_1}{\nu}\right)\right] + \frac{\rho}{\nu}\cos\left(\frac{x_1}{\nu}\right)\left[p_y - \rho\sin\left(\frac{x_1}{\nu}\right)\right] \\ +\frac{d}{\theta\nu}\left[p_z - \frac{d}{\theta\nu}x_1\right] = 0, \end{gathered} \tag{3.38}$$

where $\nu = \sqrt{\rho^2 + (d/\theta)^2}$. Solving Eq. (3.38) for x_1 yields the centerline location x_1^i for the i^{th} atom of interest. The a_2^i and a_3^i distances are calculated using

$$a_2^i = \left[\mathbf{p}_{xyz}^i - \mathbf{R}_{xyz}\left(x_1^i\right)\right] \cdot \mathbf{B}_{2_{xyz}}\left(x_1^i\right) \tag{3.39}$$

and

$$a_3^i = \left[\mathbf{p}_{xyz}^i - \mathbf{R}_{xyz}\left(x_1^i\right)\right] \cdot \mathbf{B}_{3_{xyz}}\left(x_1^i\right), \tag{3.40}$$

where the $i = 1 - 3$ denotes the i^{th} atom of the peptide plane.

3.2.5 LAMMPS Modeling

Implementing follower loads in LAMMPS requires the creation of a body fixed basis located where the applied load acts on the system. During deformation the body fixed basis is used to update the global direction of any applied load such that the applied load behaves like a follower load. The use of two body fixed bases facilitates the application of follower moments. To implement the body fixed bases, a group of seven atoms is created at the location of the applied load. This atom group creates two sets of orthonormal vectors that align with the global system. The LAMMPS input and data files are used to set the positions of the atoms contained in the body fixed bases, and defines them as a rigid body group with negligible mass such that they do not affect system dynamics. Figure 3.6 depicts both body fixed bases at the end of a simplified protein segment. The $\mathbf{V}_i$ vectors form a

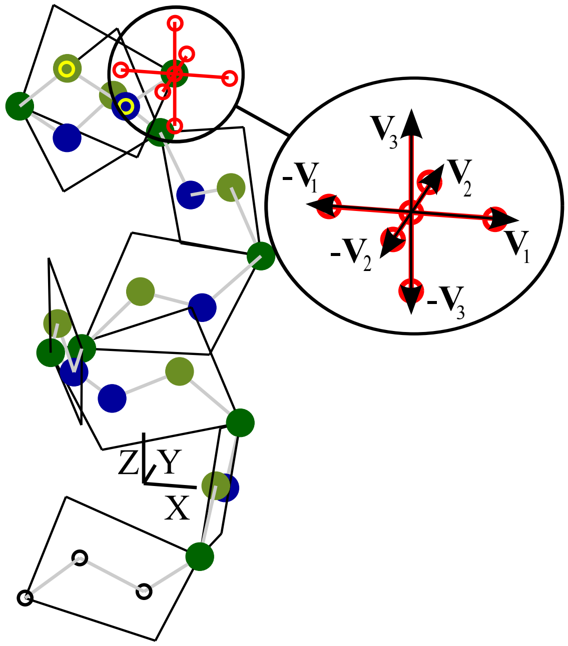


Figure 3.6: The use of two body fixed bases $\mathbf{V}_i$ and -$\mathbf{V}_i$ facilitates implementation of follower loads. The red and yellow circles denote the atoms contained in the body fixed atom group. The black circles denote three fixed atoms, which create a fixed boundary condition for the protein.

rotation matrix which is used to calculate the global orientation of applied follower forces using

$$\begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 & \mathbf{V}_3 \end{bmatrix} \begin{Bmatrix} F_{V_1} \\ F_{V_2} \\ F_{V_3} \end{Bmatrix} \tag{3.41}$$

where $\mathbf{F}_{V_i}$ denotes the body fixed forces in the $\mathbf{V}_i$ basis. The three unit vectors -$\mathbf{V}_i$, oriented in the opposite direction to $\mathbf{V}_i$, allow the application of follower moments with the use of force couples. To apply follower moments, equal and opposite follower forces are applied to the ends of the appropriate $\mathbf{V}_i$ and -$\mathbf{V}_i$ vectors such that they create the desired moment. Note that the distance between the forces in a couple using the previously described bases is two Å which should be considered when applying a follower moment. Additionally, the rigid body group of atoms defining both body fixed bases contains the two atoms nearest to the forced atom as shown in yellow in Fig. 3.6. The group must contain these atoms so that the dihedral and bond angles transfer the moment to the protein since bond lengths cannot resist rotation.

The LAMMPS MD software requires supplemental code in order to model the quadratic dihedral potentials defined in Eq. (3.1). Since LAMMPS is written in C++, the addition of the quadratic dihedrals only requires the creation of a new dihedral class. The dihedral class ‘quadratic’, developed for this simplified protein backbone, implements the quadratic dihedral potentials given in Eq. (3.1). The Appendix B contains the source code for the new dihedral class.

3.3 Results and Discussion

3.3.1 Stiffness Optimization Cases

Optimizing $\mathbf{D}_{eq}$ for three loading cases yielded a final equivalent stiffness for the hierarchical model validation. The three loading cases consisted of a fixed-free prototypical protein with 50 peptide planes loaded on the free end atom by a transverse follower force, an axial follower force, and a combination of the axial and transverse follower forces. The Matlab functions ‘fmincon’ and ‘patternsearch’ performed the multivariable optimizations, and a linearization of the concurrent multiscale stiffness outlined in Chapter 4 provided the initial point to begin optimization. As shown next, the three optimization cases yielded similar

$\mathbf{D}_{eq}$ stiffness components, and averaging the three matrices resulted in the final $\mathbf{D}^f_{eq}$,

$$\mathbf{D}^f_{eq}\left[\frac{Kcal\text{Å}}{mole}\right] = \begin{bmatrix} 27.27 & 7.97 & 19.79 \\ 7.97 & 62.02 & -9.27 \\ 19.79 & -9.27 & 165.01 \end{bmatrix}. \tag{3.42}$$

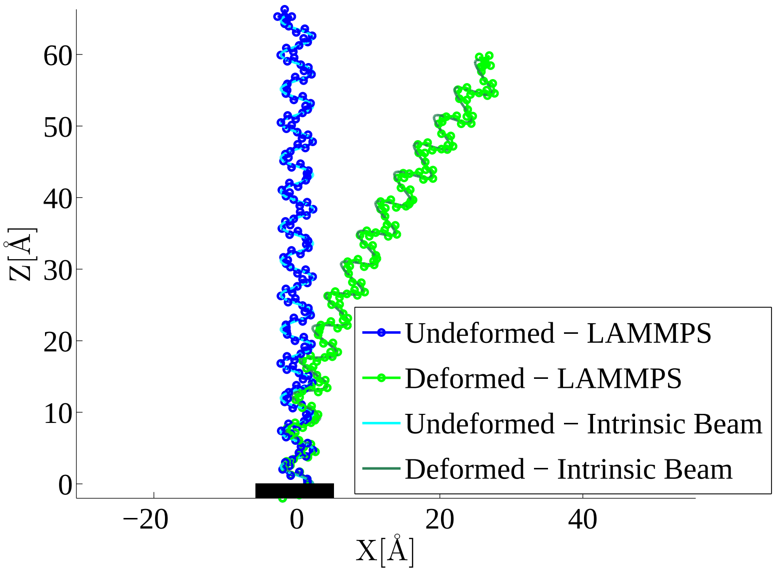
Figures 3.7-3.9 compare the deformed configurations of the hierarchical multiscale continuum model to results obtained from LAMMPS for the three loading cases using the stiffness defined in Eq. (3.42).

Figure 3.7 depicts the results from the transverse loading calibration case where there is a transverse follower force $F_{tr} = 0.0032\frac{Kcal}{mole\text{Å}}$ on the last atom of the protein. The follower force F_{tr} is initially in the positive X direction. The results show that the equivalent stiffness accurately models the protein for this case, although a very minor deviation appears in the XY plane view. The LAMMPS model of the protein exhibits coupling effects such that the protein end displaces in the positive Y direction. The equivalent intrinsic stiffness fails to display this behavior; however, the results are still favorable. The optimization algorithm determined the optimal stiffness for the transverse case to be

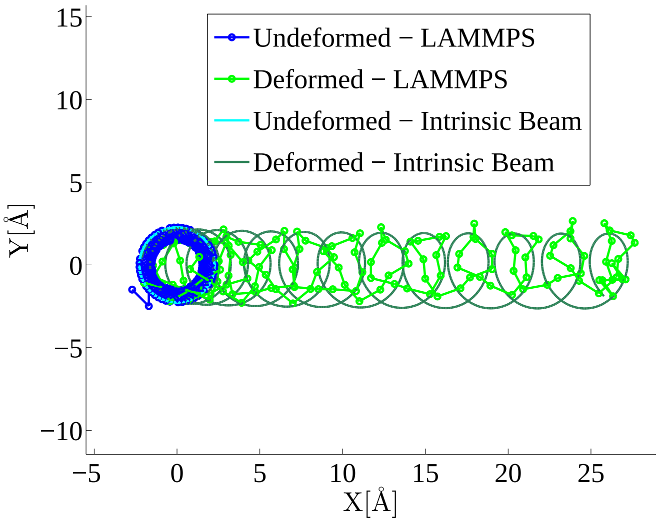
$$\mathbf{D}^{tr}_{eq}\left[\frac{Kcal\text{Å}}{mole}\right] = \begin{bmatrix} 27.29 & 7.89 & 19.67 \\ 7.89 & 62.01 & -9.34 \\ 19.67 & -9.34 & 164.9829 \end{bmatrix}. \tag{3.43}$$

The intrinsic beam model for this transverse load case locates the C_α atoms with an RMSD = 0.715 Å and a percent error of 2.40% when compared to the LAMMPS model.

Figure 3.8 depicts the axial loading calibration case where an axial extension force $F_{ex} = .0063\frac{Kcal}{mole\text{Å}}$ initially in the positive Z direction acts on the last atom in the protein. The results show that the equivalent stiffness accurately models the protein for this case. The same coupling present in the transverse case also appears in this case, resulting in a slight rotation about the Z axis. These results demonstrate strong agreement between the LAMMPS and intrinsic beam model. The optimization algorithm determined the intrinsic



(a) XZ Plane

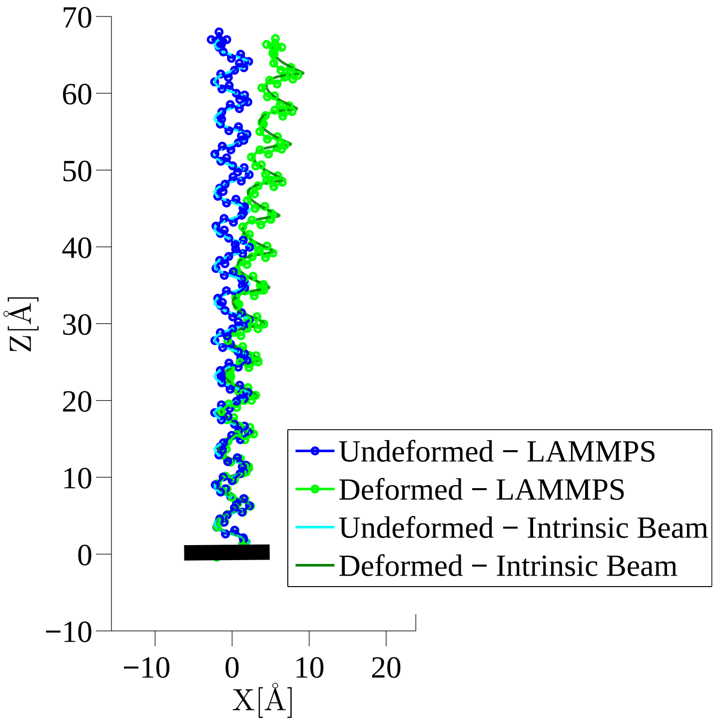


(b) XY Plane

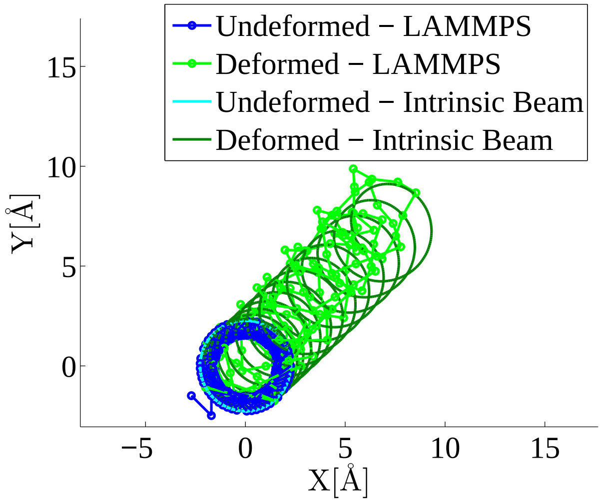
Figure 3.7: The bending loading case results for the optimized stiffness $\mathbf{D}_{eq}^{f}$. The protein is subjected to a transverse follower forces $F_{tr} = 0.0032\frac{Kcal}{mole\text{Å}}$ initially in the positive X direction.

curvature stiffness for this axial case to be

$$\mathbf{D}_{eq}^{Ax}\left[\frac{Kcal\text{Å}}{mole}\right] = \begin{bmatrix} 27.39 & 7.88 & 19.49 \\ 7.88 & 62.06 & -9.36 \\ 19.49 & -9.36 & 165.01 \end{bmatrix}. \tag{3.44}$$



(a) XZ Plane

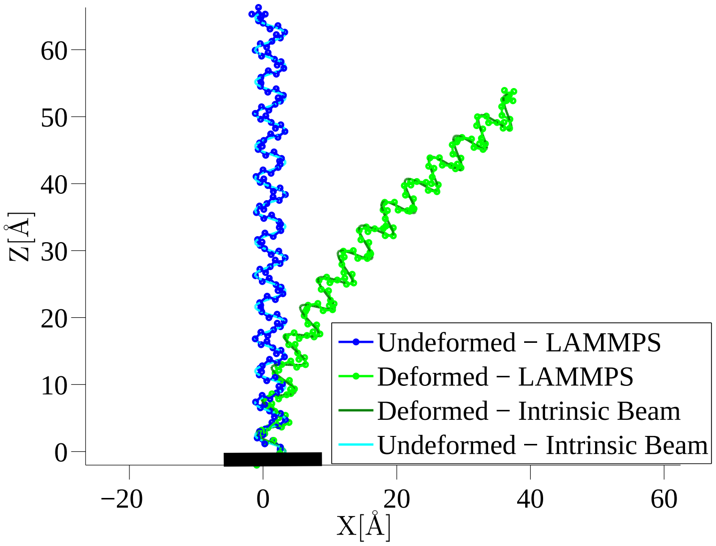


(b) XY Plane

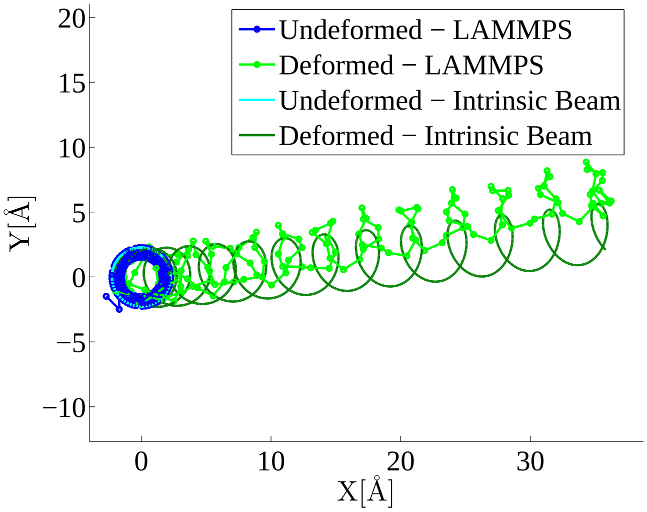
Figure 3.8: The axial loading case results for the optimized stiffness $\mathbf{D}_{eq}^{f}$. The protein is subjected to an axial extension force $F_{ex} = .0063\frac{Kcal}{mole\text{Å}}$ initially in the positive Z direction.

Additionally, the intrinsic beam model locates the C$_\alpha$ atoms with an RMSD = 0.834Å and 7.50% percent error when compared to the LAMMPS model for this axial load case.

Figure 3.9 depicts the combined loading calibration case where there are two follower



(a) XZ Plane



(b) XY Plane

Figure 3.9: The combined loading case results for the optimized stiffness $\mathbf{D}^f_{eq}$. The protein is subjected to two follower forces: an axial extension force $F_{ex} = .0063\frac{Kcal}{mole\text{Å}}$ initially in the positive Z direction and a transverse force $F_{tr} = .0032\frac{Kcal}{mole\text{Å}}$ initially in the positive X direction.

forces on the last atom of the protein. The follower forces are an axial extension force $F_{ex} = .0063\frac{Kcal}{mole\text{Å}}$ initially in the positive Z direction and a transverse force $F_{tr} = .0032\frac{Kcal}{mole\text{Å}}$ initially in the positive X direction. The results display agreement between the LAMMPS and intrinsic formulation models; however, the coupling discrepancy from the transverse

Table 3.2: Diagonal stiffness results for all loading cases.

	Transverse	**Axial**	**Combined**	$\mathbf{D}_{eq}^{D_f}$
$D_{11}\left[\frac{Kcal\text{Å}}{mole}\right]$	19.07	18.69	19.03	18.93
$D_{22}\left[\frac{Kcal\text{Å}}{mole}\right]$	78.25	78.25	78.27	78.26
$D_{33}\left[\frac{Kcal\text{Å}}{mole}\right]$	171.84	166.54	171.85	170.08
RMSD [Å]	0.6504	0.5719	1.3595	NA
% Error	2.12	4.70	3.62	NA

case reappears. The LAMMPS model displaces further in the positive Y direction and the intrinsic beam model does so to a lesser degree. The optimization algorithm determined the equivalent stiffness for the combined case to be

$$\mathbf{D}_{eq}^{Comb}\left[\frac{Kcal\text{Å}}{mole}\right] = \begin{bmatrix} 27.12 & 8.15 & 20.20 \\ 8.15 & 62.00 & -9.12 \\ 20.20 & -9.12 & 165.03 \end{bmatrix}. \tag{3.45}$$

The intrinsic beam model for this combined load case locates the C_α atoms with an RMSD = 1.9174Å and a percent error of 5.05% when compared to the LAMMPS model.

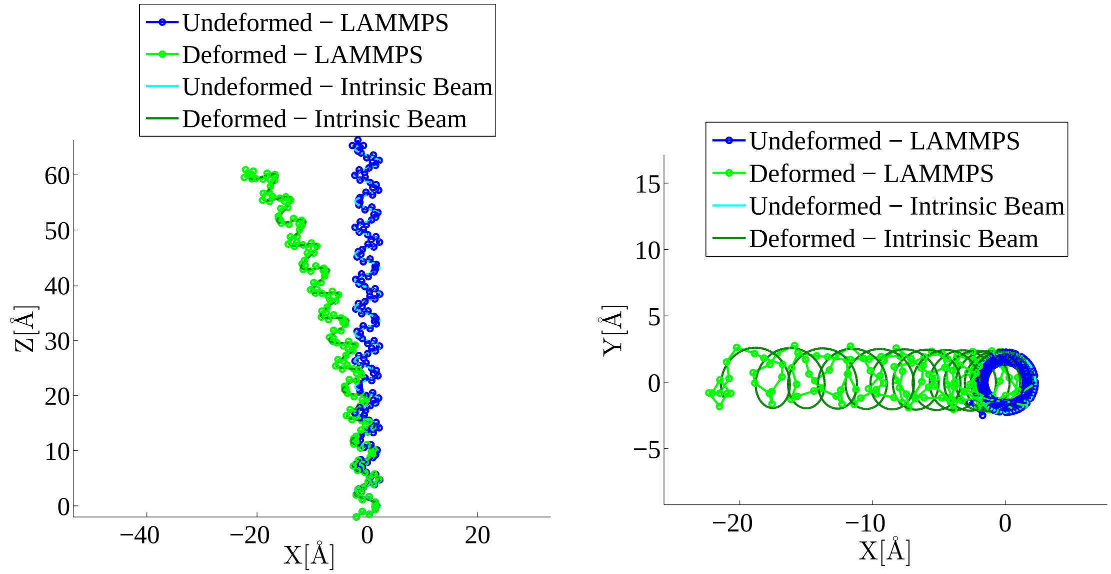
Similar results to these cases were generated for a stiffness consisting only of diagonal terms such as

$$\mathbf{D}_{eq}^{D} = \begin{bmatrix} D_{11} & 0 & 0 \\ 0 & D_{22} & 0 \\ 0 & 0 & D_{33} \end{bmatrix}. \tag{3.46}$$

The deformed and undeformed configuration figures for these cases are omitted for brevity. However, Table 3.2 reports the stiffness values and the RMSD results for all calibration loading cases, where $\mathbf{D}_{eq}^{D_f}$ denotes the average of the stiffnesses calculated for the individual load cases. Note that the RMSD values for the diagonal matrix case have improved slightly over the symmetric matrix cases, unexpectedly. This behavior may be explained by the fact that the symmetric stiffness had a larger solution space than the diagonal stiffness matrix, and that the cost function for $\mathbf{D}_{eq}$ appears to be "rugged" with many local minimums. By varying the initial starting point, the optimization would occasionally yield different $\mathbf{D}_{eq}$ matrices; therefore, a more optimal global solution may exist that was not identified in this preliminary model development.

3.3.2 Protein Static Deformed Configuration Prediction

Two static configuration prediction cases are used to validate the optimized stiffness values. The first case compares the intrinsic beam model to a LAMMPS model where a follower moment initially in the negative Y direction acts on the protein. Figure 3.10 displays the results, where strong qualitative and quantitative agreement is evident with the RMSD between the two models being 0.4784 and 0.4206 and percent errors of 2.09% and 1.56% for the $\mathbf{D}_{eq}^{f}$ and $\mathbf{D}_{eq}^{D_f}$ intrinsic stiffnesses, respectively. The second case involves the axial buckling of the protein. Figure 3.11 displays the buckling results for a 50 peptide plane simplified protein. A conservative force $F_{ax} = -0.045\frac{Kcal}{mole\text{Å}}$ acts on the protein in the negative Z direction and causes the protein helix to buckle. The deformed configuration displays good agreement with results from LAMMPS; however, a rotational difference about the Z axis between the two models demonstrates that a small coupling error exists. Notably, the Z displacement of the end of the intrinsic model matches the LAMMPS model and shows

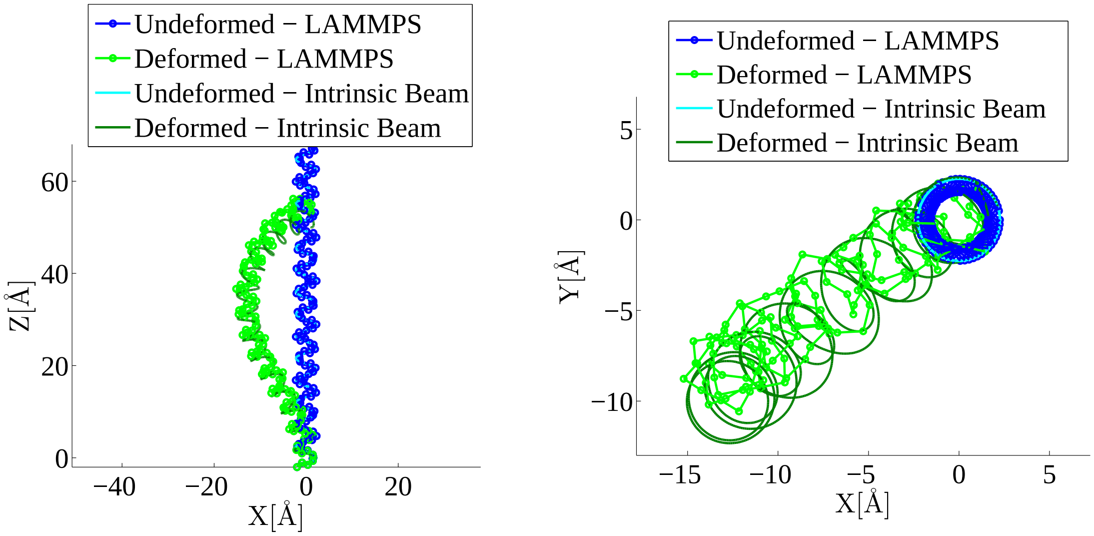


(a) XZ Plane

(b) XY Plane

Figure 3.10: The follower moment load case results for the optimized stiffness $\mathbf{D}_{eq}^{f}$.

only 10.1% and 7.67% error for $\mathbf{D}_{eq}^{f}$ and $\mathbf{D}_{eq}^{D_f}$, respectively. Furthermore, the overall buckled shapes between the two models show agreement with RMSDs of the C_α atoms equal to 2.34 Å and 1.94 Å and percent errors of 12.62% and 9.83% for $\mathbf{D}_{eq}^{f}$ and $\mathbf{D}_{eq}^{D_f}$, respectively. For these loading cases, the results agree favorably with LAMMPS MD simulations, verifying the optimized stiffnesses.



(a) XZ Plane

(b) XY Plane

Figure 3.11: The buckled load case results for the optimized stiffness $\mathbf{D}_{eq}^{f}$. The protein experiences a conservative axial force $F_{ax} = 0.045 \frac{Kcal}{mole\text{Å}}$ in the negative Z direction.

3.3.3 Dynamic Response of the Hierarchical Continuum Model

The optimized stiffness values and calculated mass values allow dynamic simulations of the simplified protein model using the intrinsic hierarchical multiscale continuum model. The dynamic finite element code written for the intrinsic formulation [32] accurately simulated the dynamic test cases using 50 intrinsic beam elements. The hierarchical model with 50 elements contains 606 degrees of freedom in comparison to the 246 degrees of freedom in an equivalent LAMMPS model. However, computational savings for the intrinsic beam model are realized by the larger allowable times steps which for the intrinsic FEA model are 2.44e-16 s compared to 1e-17 s in LAMMPS. The time steps could be increased further if strains were removed from the FEA formulation since large strain stiffnesses are utilized to keep strains small and negligible. Tests using strain stiffnesses on the order of the curvature stiffnesses required times steps of 2e-15 s, which are 200 times larger than the allowable LAMMPS time step. Similar times steps should be possible for an intrinsic FEA code where the internal force in Eq. (2.3) replaces the strain field variable since the high strain stiffnesses are removed.

Table 3.3 contains the mesh convergence results for the $\mathbf{D}_{eq}^{D_f}$ dynamic test cases. The fixed-free axial test case converges at 25 elements since an axially extended helix still exhibits constant curvature. The remaining dynamic test cases converge with 50 elements, which is 6 elements per helix turn for the modeled 25 peptide plane proteins. For these test cases, 50 elements are required for convergence since the curvature varies sinusoidally through the arc length similar to results depicted in Fig. 3.12. The quadratic shape functions used in the intrinsic beam elements cannot accurately represent the sinusoidal variance unless there are multiple elements per helix turn. As a result, 50 elements are used in the dynamic test cases for the stiffness validation; however, for studying protein dynamics, fewer elements could yield desirable results. For example, if a segment of the protein is not critical to the overall simulation results, hierarchical multiscale models with 2 to 3 finite elements per helix turn could reasonably represent the low frequency and rigid body response of the protein segment. This would yield a model with a similar number of degrees of freedom as an all-atom MD model with much larger allowable time steps. Furthermore, new shape

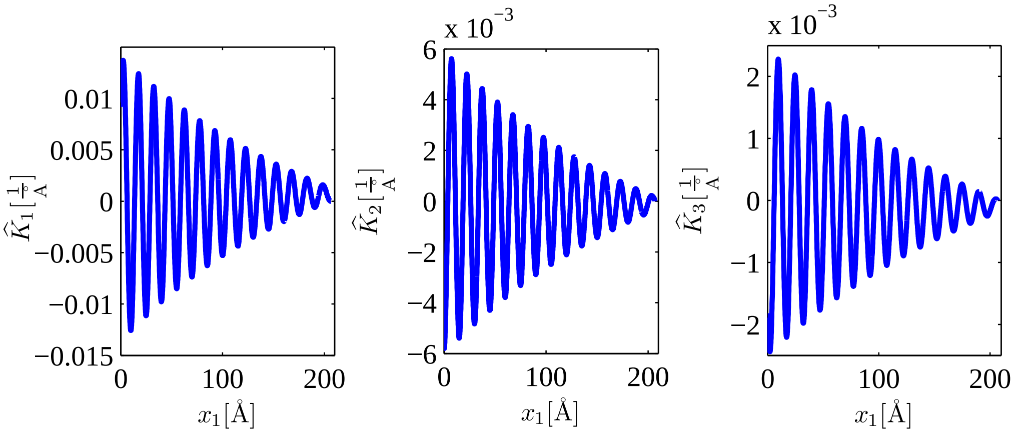


Figure 3.12: The curvatures for the static combined load calibration case are plotted as a function of x_1. The quadratic shape functions of the beam elements do not represent the sinusoidal variance well.

Table 3.3: Mesh convergence for diagonal intrinsic stiffness results. Mesh convergence for models using the symmetric intrinsic stiffness is similar.

Fixed-Fixed Axial	1^{st} Frequency %Error	**Fixed-Free Axial**	1^{st} Frequency %Error
25 Elements, 306 DoFs	11.08	25 Elements, 306 DoFs	-12.70
50 Elements, 606 DoFs	-4.18	50 Elements, 606 DoFs	-12.69
100 Elements, 1206 DoFs	-4.18	100 Elements, 1206 DoFs	-12.69
Fixed-Fixed Trans.	1^{st} Frequency %Error	**Fixed-Free Trans.**	1^{st} Frequency %Error
25 Elements, 306 DoFs	12.30	25 Elements, 306 DoFs	21.15
50 Elements, 606 DoFs	9.57	50 Elements, 606 DoFs	5.00
100 Elements, 1206 DoFs	9.57	100 Elements, 1206 DoFs	5.00

functions could be implemented that accurately represent the sinusoidal variance such that a hierarchical multiscale model with much fewer elements could accurately model the protein.

The validation of the hierarchical multiscale model's dynamics involves comparison of the intrinsic beam model results and LAMMPS MD simulation results. Specifically, natural frequencies calculated from four test cases validate the ability of the developed hierarchical continuum model to simulate the simplified protein backbone dynamics. Impulse loading applied to the models excite the system's natural frequencies, and Fast Fourier Transforms (FFT) of the atomic displacement time history results yield the natural frequencies of each test case for comparison with LAMMPS MD results. The four test cases involve the axial impulse response of a fixed-fixed and fixed-free prototypical protein and the transverse impulse response of a fixed-fixed and fixed-free prototypical protein.

3.3.3.1 Impulse Response of Fixed-Fixed Backbone Model

The hierarchical multiscale method reproduces the primary frequency of the protein with error on the order of 10% or less for the two following cases. The first validation case is a fixed-fixed simplified protein with 25 peptide planes excited with a transverse impulse load in the center. Figure 3.13 depicts the loading and boundary conditions for this case.

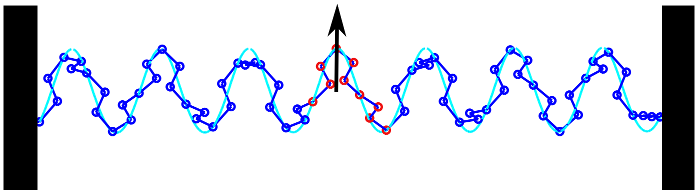
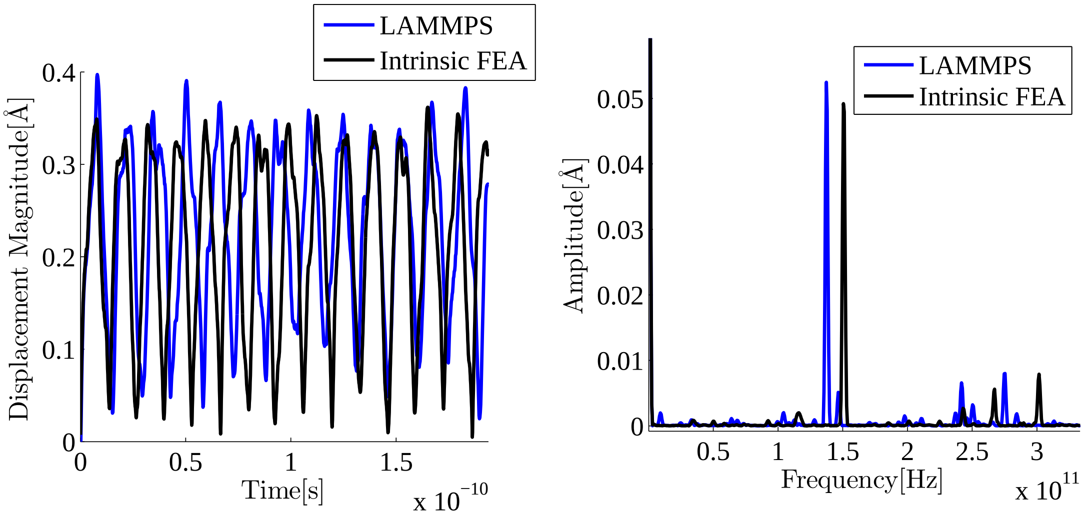


Figure 3.13: Fixed-fixed protein loaded transversely with an impulse load. The dark blue line depicts the LAMMPS model, the light blue line depicts the intrinsic formulation model, and the red circles show the forced atoms

Figure 3.14 displays the time history and frequency content of the response from both the LAMMPS model and the hierarchical multiscale model using $\mathbf{D}_{eq}^{D_f}$. These results give the displacement magnitude of the center C$_\alpha$ atom and show favorable agreement. The primary frequency demonstrated in the intrinsic continuum has 11.85% and 9.57% error for the $\mathbf{D}_{eq}^{f}$ and $\mathbf{D}_{eq}^{D_f}$ optimized stiffnesses, respectively, when compared to the primary frequency exhibited by the LAMMPS model.



(a) Time history

(b) Frequency Response

Figure 3.14: The dynamic results from the hierarchical multiscale continuum model using $\mathbf{D}_{eq}^{D_f}$ correspond with results from LAMMPS MD for the fixed-fixed validation case with transverse impulse loading.

The second validation case is a fixed-fixed simplified protein with 25 peptide planes excited with an axial impulse loading in the center. Figure 3.15 depicts the loading and boundary conditions for this case. Figure 3.16 displays the time history and frequency content of the response from both the LAMMPS model and the hierarchical multiscale

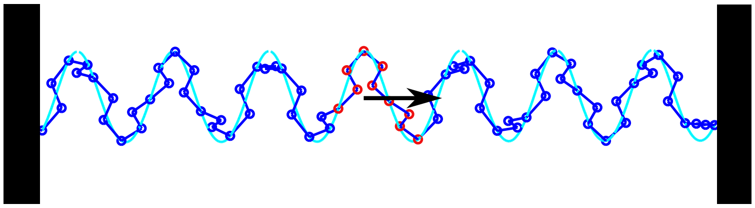
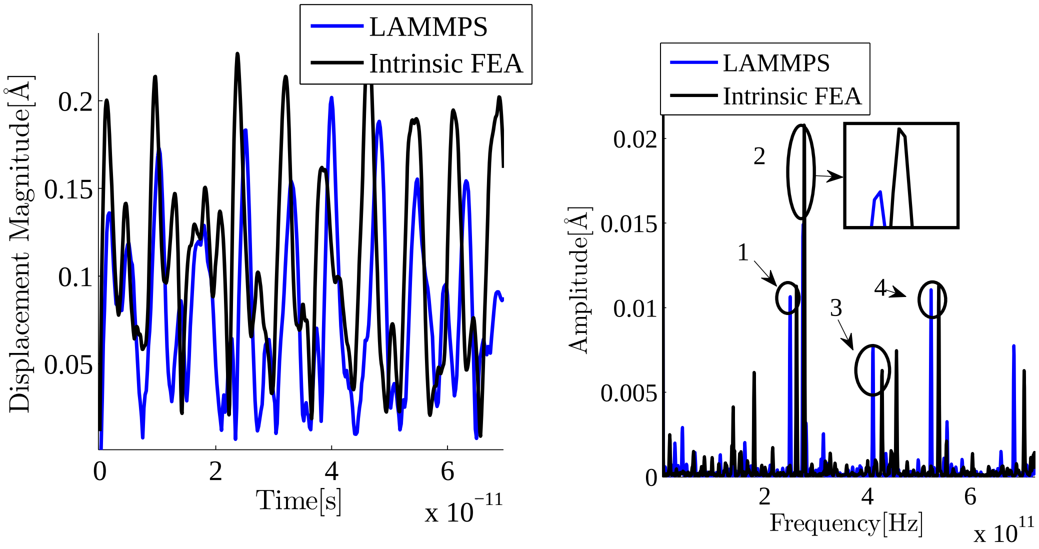


Figure 3.15: Fixed-fixed protein loaded axially with an impulse load. The dark blue line depicts the LAMMPS model, the light blue line depicts the intrinsic formulation model, and the red circles show the forced atoms.



(a) Time history

(b) Frequency Response

Figure 3.16: The dynamic results from the hierarchical multiscale continuum model using $\mathbf{D}_{eq}^{f}$ correspond with results from LAMMPS MD for the fixed-fixed validation case with axial impulse loading.

model using $\mathbf{D}_{eq}^{f}$. These results present the displacement magnitude of the center C_α atom and show favorable agreement. The four primary frequencies demonstrated in the intrinsic continuum have percent errors less than 5% and 9% for $\mathbf{D}_{eq}^{f}$ and $\mathbf{D}_{eq}^{D_f}$, respectively, when compared to the primary frequencies exhibited by the LAMMPS model. Table 3.4 contains the four investigated frequencies for both the intrinsic beam models and LAMMPS along with the associated percent errors.

3.3.3.2 Impulse Response of Fixed-Free Backbone Model

The hierarchical multiscale method reproduces the first primary frequency of the protein with error on the order 10% or less for the two following fixed-free cases. The first validation

Table 3.4: Axial Fixed-Fixed Case Results

	Peak 1	**Peak 2**	**Peak 3**	**Peak 4**
LAMMPS Freq[Hz]	2.497e11	2.747e11	4.108e11	5.238e11
Intrinsic $\mathbf{D}_{eq}^{f}$ Freq[Hz]	2.617e11	2.766e11	4.284e11	5.389e11
% Error	4.832	0.710	4.282	2.896
Intrinsic $\mathbf{D}_{eq}^{D_f}$ Freq[Hz]	2.392e11	2.510e11	3.897e11	4.9021e11
% Error	-4.175	-8.610	-5.148	-6.40382

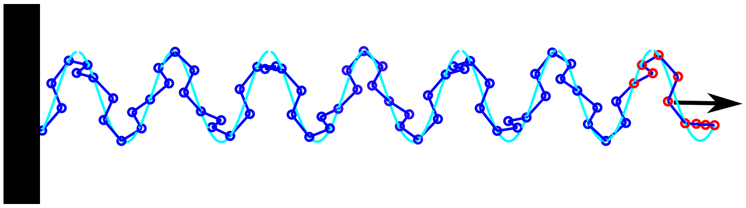
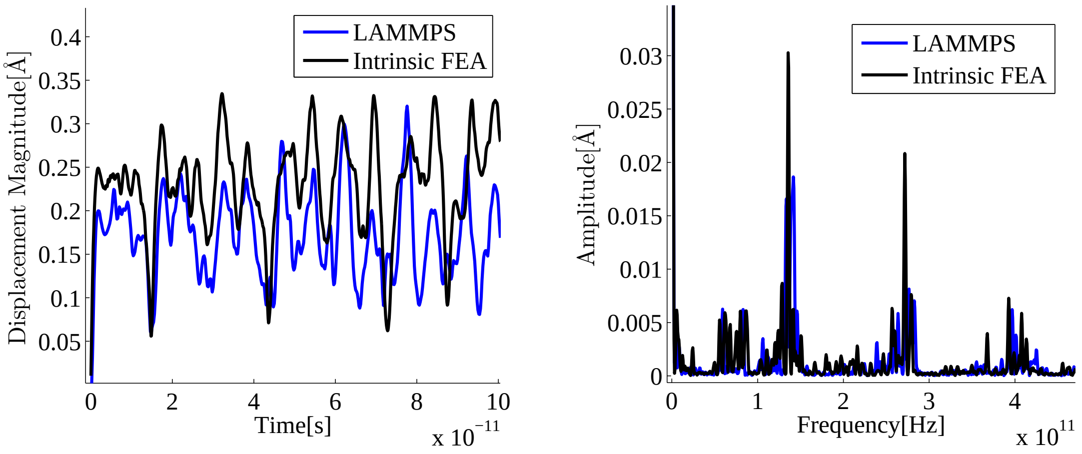


Figure 3.17: Fixed-free protein loaded axially with an impulse load. The dark blue line depicts the LAMMPS model, the light blue line depicts the intrinsic formulation model, and the red circles show the forced atoms.

case is a fixed-free simplified protein with 25 peptide planes excited with an axial impulse loading at the free end. Figure 3.17 depicts the loading and boundary conditions for this case. Figure 3.18 displays the time history and frequency content of the response from both



(a) Time history

(b) Frequency Response

Figure 3.18: The dynamic results from the hierarchical multiscale continuum model using $\mathbf{D}_{eq}^{f}$ correspond with results from LAMMPS MD for the fixed-free validation case with axial impulse loading.

the LAMMPS model and the hierarchical multiscale model using $\mathbf{D}_{eq}^{f}$. These results give

Table 3.5: Transverse Fixed-Free Case Results

	Peak 1	**Peak 2**	**Peak 3**	**Peak 4**
LAMMPS Freq[Hz]	2.320e10	4.700e10	5.921e10	8.302e10
Intrinsic $\mathbf{D}_{eq}^{f}$ Freq[Hz]	2.436e10	4.901e10	6.120e10	8.605e10
% Error	5.000	4.270	3.360	3.65
Intrinsic $\mathbf{D}_{eq}^{D_f}$ Freq[Hz]	2.346e10	4.810e10	5.995e10	8.430e10
% Error	5.000	-2.302	1.248	1.547

the displacement magnitude of the center C_α atom and show favorable agreement. The primary frequency demonstrated in the intrinsic continuum has only -4.31% and -12.69% error for the $\mathbf{D}_{eq}^{f}$ and $\mathbf{D}_{eq}^{D_f}$ stiffnesses, respectively, when compared to the primary frequency exhibited by the LAMMPS model. Additionally, the FFT results show strong agreement with the grouping of minor frequencies between both models.

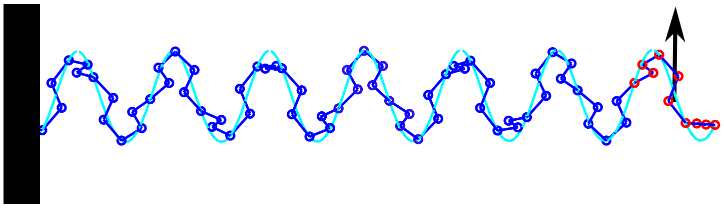
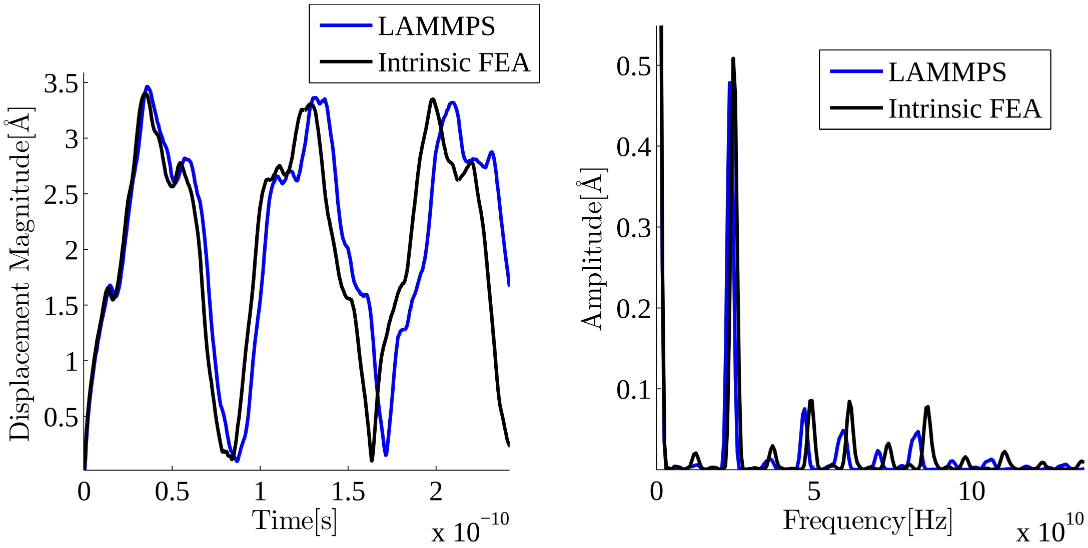


Figure 3.19: Fixed-free protein loaded transversely with an impulse load. The dark blue line depicts the LAMMPS model, the light blue line depicts the intrinsic formulation model, and the red circles show the forced atoms.

The second validation case is a fixed-free simplified protein with 25 peptide planes excited with a transverse impulse loading at the free end. Figure 3.19 depicts the loading and boundary conditions for this case. Figure 3.20 displays the time history and frequency content of the response from both the LAMMPS model and the hierarchical multiscale model $\mathbf{D}_{eq}^{f}$. These results give the displacement magnitude of the center C_α atom and show favorable agreement. The four primary frequencies demonstrated in the intrinsic continuum have percent errors near 5% for both $\mathbf{D}_{eq}^{f}$ and $\mathbf{D}_{eq}^{D_f}$ stiffnesses. Table 3.4 contains the first four frequencies exhibited by both the intrinsic beam models and the LAMMPS MD simulations along with the associated percent errors.



(a) Time history (b) Frequency Response

Figure 3.20: The dynamic results from the hierarchical multiscale continuum model using $\mathbf{D}^f_{eq}$ correspond with results from LAMMPS MD for the fixed-free validation case with transverse impulse loading.

3.4 Conclusion

The developed hierarchical multiscale continuum model uses the intrinsic beam formulation to model protein dynamics and statics. The developed model captures atomic resolution by locating C_α atoms in the protein and dihedral angles can be estimated due to the dihedral angle to curvature mapping. The optimization routine accurately calculated the correct intrinsic stiffness such that the helical beam behaves similar to all-atom solutions determined using LAMMPS MD. The multiscale hierarchical continuum model predicted the static and dynamic characteristics of the protein when compared to the LAMMPS MD results. Chapter 5 discusses future areas of improvement for the model and introduces an incremental plan for its further development.

CHAPTER IV

CONCURRENT MULTISCALE CONTINUUM MODEL

4.1 Introduction

Concurrent multiscale modeling techniques combine models of two differing accuracies such that higher resolution models allow high accuracy where desired, and lower resolution models allow reduced degrees of freedom elsewhere in the model. Typically, quantum mechanics (QM) models and MD models are combined for modeling systems where protein interactions and chemical reactions occur. The QM models represent the protein sections where chemical reactions occur, and the MD model sections represent the remaining protein segments where locating atoms and determining protein configurations remain of interest. Similarly, concurrent multiscale models have been developed which combine coarse-grained MD models with all-atom MD models such that the all-atom models correspond to areas where high resolution models are needed ([53], [40]). These techniques yield accurate results and decrease computation time significantly when compared to corresponding models that consist of a complete high resolution model. However, these models still contain a prohibitive amount of degrees of freedom and require small time steps. A possible solution is a concurrent multiscale continuum model for protein dynamics, which allows atomic resolution for sampling internal energy while reducing the number of degrees of freedom such that the model remains of reasonable size.

This chapter introduces a preliminary concurrent, multiscale, continuum model for simulating protein dynamics. As with the hierarchical multiscale model in Chapter 3, the simplified backbone model is used for initial model development and a finite element program simulates system dynamics [31]. The model maintains atomistic resolution of the protein at points of interest, and utilizes atomistic potential terms to calculate internal forces and moments for the continuum. The model extends the work performed by Leamy

[31], where a similar concurrent multiscale technique accurately simulates the dynamic response of carbon nanotubes.

4.2 Methods

This section introduces the kinematics relating the continuum field variables to the atom positions such that the atomistic potential can be evaluated. Additionally, this section presents the representative volume element describing the protein and a brief overview of the finite element discretization of the protein.

4.2.1 Atomistic-based Constitutive Modeling

The concurrent multiscale continuum model evaluates atomistic potentials by calculating the relative position between any two atoms, which may be located away from the deformed beam centerline. An equation that relates the intrinsic curvatures and strains to global position vectors defines the relative atom positions. These relative position vectors can be used to determine dihedral angles, bond angles and bond lengths. The relation is formulated through an expansion of material point position vectors $\mathbf{R}^*(x_1 + dx_1, x_2, x_3)$ and the $\mathbf{B}_i$ basis vectors. The presented equations are limited to expansions of $O(dx_1^3)$; however, the method requires expansions up to order $O(dx_1^9)$ for convergence.

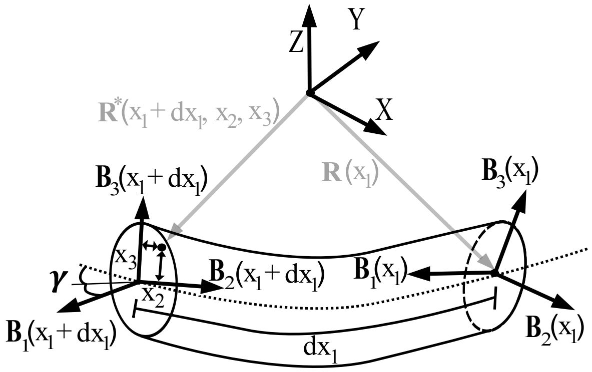


Figure 4.1: The material point parameters used in the intrinsic beam element development are used to calculate the relative position of atoms.

Figure 4.1 displays the vectors required to calculate the atomistic potential terms. The position vector $\mathbf{R}^*(x_1 + dx_1,\ x_2(x1),\ x_3(x1))$ gives the deformed configuration position of any atom relative to the centerline position $\mathbf{R}(x_1)$. The $\mathbf{R}^*(x_1 + dx_1)$ position vector is expressed using $\mathbf{R}(x_1 + dx_1)$ and the material points $x_2(x_1 + dx_1) = b_2,\ x_3(x_1 + dx_1) = b_3$ such that

$$\mathbf{R}^*(x_1 + dx_1;\ b_2,\ b_3) = \mathbf{R}(x_1 + dx_1) + b_2\mathbf{B}_2(x_1 + dx_1) + b_3\mathbf{B}_3(x_1 + dx_1). \tag{4.1}$$

To determine $\mathbf{R}(x_1 + dx_1)$ and $\mathbf{B}_i(x_1 + dx_1)$, an expansion is applied to both $\mathbf{R}(x_1)$ and $\mathbf{B}_i(x_1)$. Applying an expansion to $\mathbf{B}_i(x_1)$ gives

$$\begin{aligned} \mathbf{B}_i(x_1 + dx_1) &= \mathbf{B}_i(x_1) + \mathbf{B}_i'(x_1)dx_1 + \mathbf{B}_i''(x_1)dx_1^2 + O(dx_1^3) \\ &= \mathbf{B}_i(x_1) + (\mathbf{K}(x_1) \times \mathbf{B}_i(x_1)dx_1) + \frac{1}{2}\big(\mathbf{K}(x_1) \times \mathbf{K}(x_1) \times \mathbf{B}_i(x_1) \\ &\quad + \mathbf{K}'(x_1) \times \mathbf{B}_i(x_1)\big)\, dx_1^2 + O(dx_1^3) \end{aligned} \tag{4.2}$$

Similarly, the expansion of $\mathbf{R}(x_1)$ results in

$$\mathbf{R}(x_1 + dx_1) = \mathbf{R}(x_1) + \mathbf{R}'(x_1)dx_1 + \mathbf{R}''(x_1)dx_1^2 + O(dx_1^3) \tag{4.3}$$

where $\mathbf{R}'(x_1) = (1 + \gamma_{11})\,\mathbf{B}_1(x_1) + 2\gamma_{12}\mathbf{B}_2(x_1) + 2\gamma_{13}\mathbf{B}_3(x_1)$. Combining Eqs. (4.1)-(4.3) yields the final expression for $\mathbf{R}^*(x_1 + dx_1)$,

$$\begin{aligned} \mathbf{R}^*(x_1 + dx_1;\ b_2,\ b_3) = \mathbf{R}(x_1) &+ \left[(1 + \gamma_{11} + b_3K_2 - b_2 + K_3)\, dx_1\right] \mathbf{B}_1(x_1) \\ &+ \left[b_2 + (2\gamma_{12} - b_3K_1)\, dx_1\right] \mathbf{B}_2(dx_1) \\ &+ \left[b_3 + (2\gamma_{13} + b_2K_1)\, dx_1\right] \mathbf{B}_3(dx_1) + O(dx_1^2). \end{aligned} \tag{4.4}$$

Equation (4.4) results in the relative position of atom i to atom j by using $\mathbf{r}_{ij} = \mathbf{R}^*\left(x_1 + dx_1^i; b_2^i, b_3^i\right) - \mathbf{R}^*\left(x_1 + dx_1^j; b_2^j, b_3^j\right)$. As a result, $\mathbf{R}(x_1)$ cancels out and the relative position vector is known without any need for the global centerline position. Using this relationship, the following equations define the protein lengths and angles required to evaluate the simplified protein potential from Eq. (3.1) in Chapter 3:

$$\begin{gathered} b_{ij} = ||\mathbf{r}_{ij}||\,,\ \cos(\theta_{ijk}) = \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{jk}}{||\mathbf{r}_{ij}||\,||\mathbf{r}_{jk}||},\ \cos(\chi_{ijkl}) = \frac{\mathbf{r}_{ij} \times \mathbf{r}_{jk} \cdot \mathbf{r}_{jk} \times \mathbf{r}_{kl}}{||\mathbf{r}_{ij} \times \mathbf{r}_{jk}||\,||\mathbf{r}_{jk} \times \mathbf{r}_{kl}||} \\ \text{and } \sin(\chi_{ijkl}) = \frac{\left[(\mathbf{r}_{kl} \times \mathbf{r}_{ij}) \cdot \mathbf{r}_{jk}\right] ||\mathbf{r}_{jk}||}{||\mathbf{r}_{ij} \times \mathbf{r}_{jk}||\,||\mathbf{r}_{jk} \times \mathbf{r}_{kl}||\,.} \end{gathered} \tag{4.5}$$

In Eq. (4.5), b_{ij} denotes the bond length between two atoms, θ_{ijk} denotes the bond angle between three atoms, and χ_{ijkl} denotes the dihedral angle between four atoms.

The atomistic-based constitutive law is complete with the definition of the internal forces and moments as a function of the energy per unit length, u_e, of the continuum. The internal forces and moments relate to the derivatives of the internal energy per unit length via

$$\mathbf{F} = \frac{\partial u}{\partial \boldsymbol{\gamma}}, \tag{4.6}$$

and

$$\mathbf{M} = \frac{\partial u}{\partial \mathbf{K}}. \tag{4.7}$$

In this constitutive relationship, a representative volume element (rve) locally yields the internal energy per unit length of the continuum in terms of the protein lengths and angles given by Eq. (4.5).

4.2.2 Representative Volume Elements

Representative volume elements connect the intrinsic formulation field variables $(\mathbf{K}, \boldsymbol{\gamma}, \boldsymbol{\Omega}, \mathbf{V})$ to the atomic potential energy at pre-defined centerline locations. The rve allows the sampling and averaging of energy at the atomic level without requiring calculation of all the atom positions, thus significantly reducing the order of the model. The rve must be a repeating unit that is representative of the system such that at any x_1 location in the continuum an rve accurately describes the system internal energy. The left peptide plane depicted in Fig. 4.2 represents the most general rve for a protein since it is the smallest representative unit that defines all bonded terms in the potential from Eq. (1.3). Similarly,

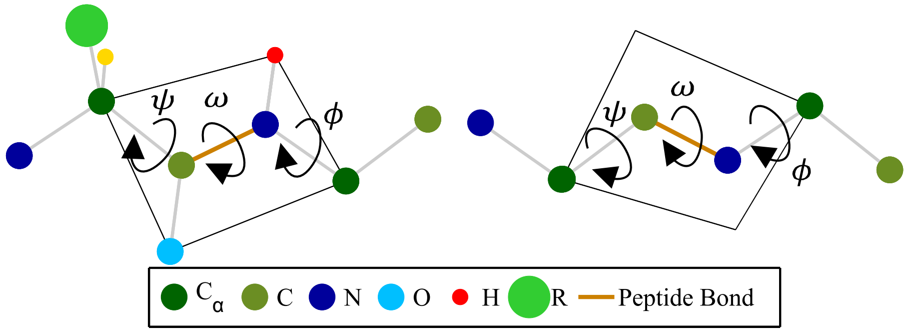


Figure 4.2: The left peptide plane represents the rve used for a full protein, while the right peptide plane represents the simplified protein rve.

Fig. 4.2 displays the peptide plane rve used for the simplified backbone protein on the right.

The nitrogen and carbon atoms on the exterior of the peptide plane are not considered part of the rve; however, their positions affect the ϕ and ψ dihedrals so their locations must be calculated.

The internal energy of the continuum per unit length u_e relates to the rve atomistic energy potential U_s^{rve} via

$$u_e = \frac{U_s^{rve}}{l^{rve}}, \tag{4.8}$$

where l^{rve} represents the characteristic length of the rve and is equal to L_{Pep} for the developed prototypical protein model. Due to the chosen rve, three bond lengths, dihedral angles, and bond angles determine the simplified protein rve atomistic energy,

$$\begin{aligned} U_s^{rve} &= K_{b_1}(b_1 - b_{1_0})^2 + K_{b_2}(b_2 - b_{2_0})^2 + K_{b_3}(b_3 - b_{3_0})^2 \\ &\quad + K_{\theta_1}(\theta_1 - \theta_{1_0})^2 + \frac{1}{2}K_{\theta_2}(\theta_2^1 - \theta_{2_0})^2 + \frac{1}{2}K_{\theta_2}(\theta_2^2 - \theta_{2_0})^2 + K_{\theta_3}(\theta_3 - \theta_{3_0})^2 \\ &\quad + K_\phi(\phi - \phi_0)^2 + K_\psi(\psi - \psi_0)^2 + K_\omega(\omega - \omega_0)^2. \end{aligned} \tag{4.9}$$

The two bond angles θ_2^1 and θ_2^2 are required in the energy function given in Eq. (4.9) since the peptide plane contains only half of each angle. Figure 4.3 shows the location of each term from Eq. (4.9) in the peptide plane, and illustrates that half of the bond angle energy must be used. Using the chain rule, the internal forces and moments are calculated using

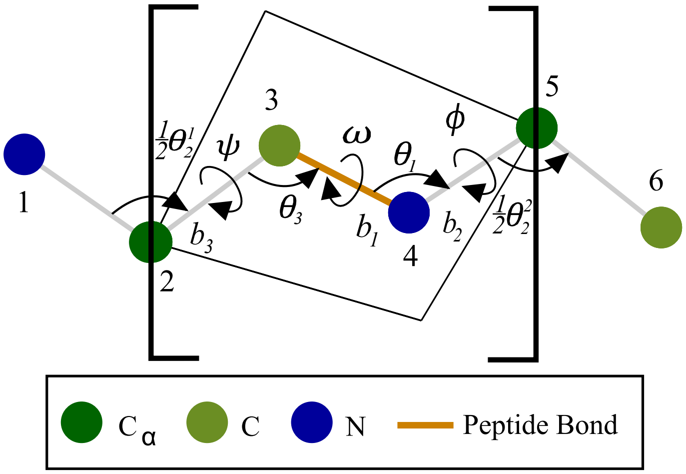


Figure 4.3: The simplified peptide plane rve contains 3 bond lengths, bond angles, and dihedral angles. The bond lengths of the two carbon and nitrogen atoms external to the peptide plane do not affect the rve energy.

$$\mathbf{F} = \frac{\partial U_s^{rve}}{\partial b_{ij}} \frac{\partial b_{ij}}{\partial \boldsymbol{\gamma}} + \frac{\partial U_s^{rve}}{\partial \theta_{ijk}} \frac{\partial \theta_{ijk}}{\partial \boldsymbol{\gamma}} + \frac{\partial U_s^{rve}}{\partial \chi_{ijkl}} \frac{\partial \chi_{ijkl}}{\partial \boldsymbol{\gamma}} \tag{4.10}$$

and

$$\mathbf{M} = \frac{\partial U_s^{rve}}{\partial b_{ij}} \frac{\partial b_{ij}}{\partial \mathbf{K}} + \frac{\partial U_s^{rve}}{\partial \theta_{ijk}} \frac{\partial \theta_{ijk}}{\partial \mathbf{K}} + \frac{\partial U_s^{rve}}{\partial \chi_{ijkl}} \frac{\partial \chi_{ijkl}}{\partial \mathbf{K}}, \tag{4.11}$$

where the indices i, j, k and l represent consecutive atoms appearing in the rve. With Eqs. (4.4) and (4.5), finite difference approximations yield the partial derivatives of the protein potential parameters with respect to the deformed state variables $\boldsymbol{\gamma}$ and $\mathbf{K}$. Due to the high order expansion needed, the finite difference technique reduces the complexity of the derivative calculations during numerical implementation. Equations (4.10) and (4.11) supply the link between the continuum stiffness and atomistic potentials without the need for bulk material properties such as Young's modulus, thus allowing concurrent multiscale modeling of the prototypical protein polypeptide.

Calculating the material point parameters dx_1, b_2 and b_3 requires the same steps used to calculate a_2^i and a_3^i. A numerical technique solves the transcendental equation given by Eq. (3.38) for the dx_1 material point parameter in place of x_1 for each atom. Then, the b_2 and b_3 material point parameters are calculated similar to a_2^i and a_3^i using

$$b_2^i = \left[\mathbf{p}_{xyz}^i - \mathbf{R}_{xyz}\left(dx_1^i\right)\right] \cdot \mathbf{B}_{2_{xyz}}\left(dx_1^i\right) \tag{4.12}$$

and

$$b_3^i = \left[\mathbf{p}_{xyz}^i - \mathbf{R}_{xyz}\left(dx_1^i\right)\right] \cdot \mathbf{B}_{3_{xyz}}\left(dx_1^i\right), \tag{4.13}$$

where the $i = 1 - 6$ denotes the i^{th} atom of the peptide rve shown in Fig. 4.3.

The atomic positions calculated with the continuum field variables γ and $\mathbf{K}$ require an 8^{th} order expansion of Eqs. (4.2) and (4.3) such that Eq. (4.4) converges. The high order expansion is needed since the centerline through the rve is approximately 7Å in length in the Ω_0 configuration. With $\mathbf{R}(x_1)$ located at the center of the rve, the smallest possible centerline material point dx_1 is 3.5Å; therefore, the large distance away from x_1 requires the high order expansion. Since the dx_1 values calculated above use the xyz helix coordinate system, $\mathbf{R}(x_1)$ is located at the first C_α atom. To shift the atoms such that $\mathbf{R}(x_1)$ is located in the center of the rve, $\frac{1}{2}L_{Pep}$ is subtracted from each dx_1^i. This effectively shifts the location of $\mathbf{R}(x_1)$ and minimizes the largest dx_1^i material point parameter such that Eq. (4.4) converges with the lowest order expansion possible.

4.2.3 Inner Displacements

Inner displacements allow for more realistic modeling of the rve by reducing undesired kinematic constraints and are imposed between time steps as to not affect the rve dynamics. Inner displacements result from an rve energy relaxation which reduces kinematic constraints on the rve imposed by the chosen element shape function and Eq. (4.4). Due to quadratic shape functions governing the field variables variation through the element, the curvature and strain through an rve are constrained. Such a constraint can limit how atoms move in an artificial manner. To reduce the effect of the kinematic constraints, an energy relaxation is used to calculate the energy minimum for the current Ω_f configuration and updates material points accordingly. During inner displacement calculations, the field variables and time remain constant resulting in quasistatic displacements.

The energy relaxation updates the material points for rve atoms 1 and 6 such that only the ϕ and ψ dihedral angles are changed. The inner displacements cause the dihedrals to update using

$$\phi = \phi + \eta^d \tag{4.14}$$

and

$$\psi = \psi - \eta^d, \tag{4.15}$$

where η^d denotes the inner displacement value. This inner displacement form allows the dihedral angles to rotate out of phase since the element constrains the dihedrals to rotate in phase. A minimization of the ϕ and ψ dihedral energy terms yields the inner displacement η^d, where the ϕ and ψ energy is

$$E_{\phi,\psi} = K_\phi(\phi + \eta^d - \phi_0)^2 + K_\psi(\psi - \eta^d - \psi_0)^2. \tag{4.16}$$

The η^d is calculated by solving $\frac{\partial E_{\phi,\psi}}{\partial \eta^d} = 0$ resulting in

$$\eta^d = \frac{-K_\phi(\phi - \phi_0) - K_\psi(\psi - \psi_0)}{K_\phi + K_\psi}. \tag{4.17}$$

To implement the η^d inner displacement, atoms 1 and 6 are rotated by η^d about the corresponding ϕ and ψ bond angles, and their material point parameters are recalculated using the process described previously. This material point update process remains valid for small displacements since the deformed configuration is approximately equal to the undeformed configuration, so Eqs. (3.37)-(3.40) from Chapter 3 remain valid. A procedure for updating material points during large displacements for the use of implementing inner displacements has not been developed.

4.2.4 Finite Element Discretization and Implementation

Leamy [31] presents a detailed derivation of the finite element formulation used for modeling the concurrent multiscale continuum model. The important features of the finite element formulation are reviewed here including shape function definition and implementation of the rve. Three-noded isoparametric shape functions define the field variables such that

$$x_1 = N_I(\varsigma)x_1^I, \mathbf{V} = N_I(\varsigma)\mathbf{V}^I, \mathbf{\Omega} = N_I(\varsigma)\mathbf{\Omega}^I, \mathbf{K} = N_I(\varsigma)\mathbf{K}^I, \boldsymbol{\gamma} = N_I(\varsigma)\boldsymbol{\gamma}^I, \tag{4.18}$$

where ς denotes a local coordinate ranging from -1 to 1, and the $I = 1 - 3$ superscript represents the nodal quantities for the element. As stated previously, the developed intrinsic beam elements utilize quadratic shape functions given by

$$N_1 = \frac{1}{2}\varsigma\left(\varsigma - 1\right), N_2 = 1 - \varsigma^2, N_1 = \frac{1}{2}\varsigma\left(\varsigma + 1\right) \tag{4.19}$$

where the shape functions evaluate to one at their corresponding node and zero at the other two nodal locations. These quadratic shape functions limit the complexity of Eq. (4.4) in higher order expansions since the derivatives of the shape functions evaluate to zero for third order and higher spatial derivatives. With the shape functions defined, the internal forces and moments are calculated using a Guass quadrature routine. Since the quadratic shape functions govern the field variables, four integration points accurately calculate the integrals defining the internal forces and moments defined in [31]. At these integration points, it is assumed that the rves accurately represent the local atomic energy of the system. Therefore, the rve energy is evaluated at each integration point using Eqs. (4.4) and (4.5) resulting in the internal forces and moments at the integration points. Using these internal forces and moments along with the mass properties described in Chapter 3, the finite element code simulates the concurrent multiscale continuum model dynamics.

4.3 Results

4.3.1 Step Repsonse

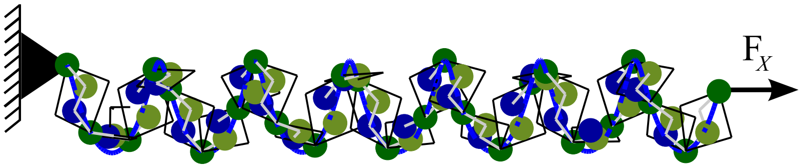


Figure 4.4: The concurrent multiscale continuum model test case studied the step response of the simplified protein to an axial force.

This test case documents the response of the concurrent multiscale continuum protein model to a step force input. Figure 4.4 illustrates the test case where a step force input acts on a pinned-free simplified protein. Specifically, three intrinsic protein elements modeled a 24 peptide plane simplified protein with an applied step force input $\mathbf{F}_X = 0.01\frac{Kcal}{mole\text{Å}}$. The metrics used for comparison to LAMMPS MD simulations included the displacement magnitude of the end of the protein and the simplified protein potential terms. Figures 4.5 and 4.6 display these results and compare them to results obtained from LAMMPS MD. Figure 4.5 illustrates that the concurrent multiscale continuum model behaves in an

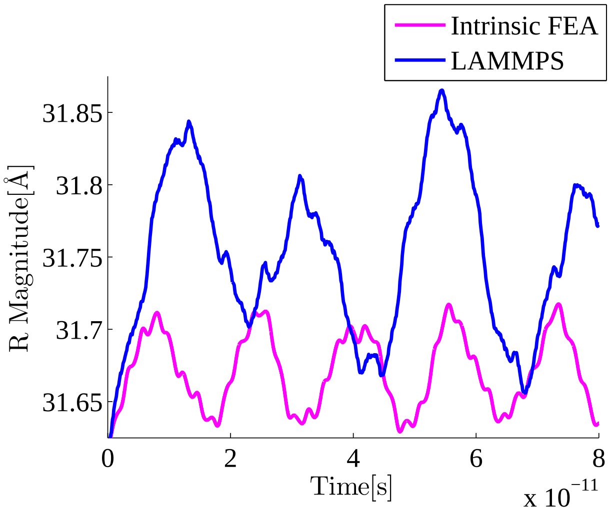
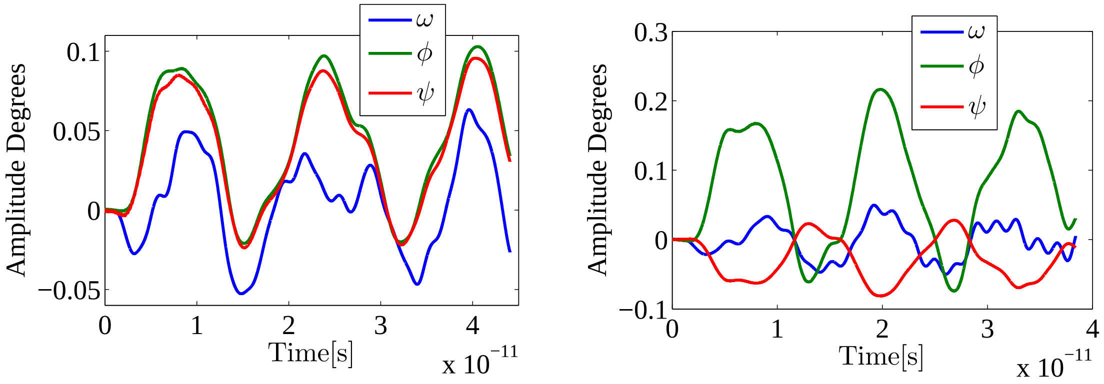


Figure 4.5: The concurrent multiscale continuum model exhibits stiff behavior when compared to the LAMMPS MD simulations.

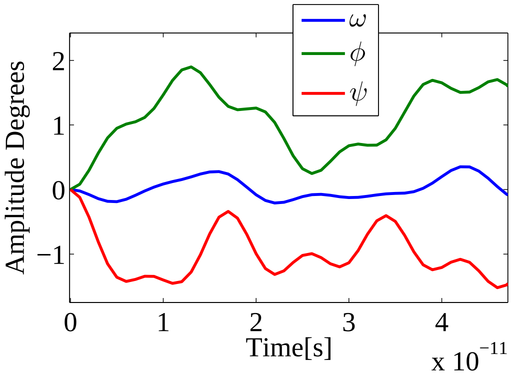
overly stiff manner since the displacement magnitude is significantly less and the response frequency is higher.

To explain the cause of this over stiffness, Figure 4.6 compares the dihedral angle results from the continuum model without inner displacements, the continuum model with inner displacements and the LAMMPS MD simulations. Notably, the dihedral angle magnitudes predicted by the concurrent multiscale model are an order of magnitude less than the LAMMPS results presented by Fig. 4.6c. Additionally, Figs. 4.6a and 4.6b demonstrate that the inner displacements behave as desired and relax the energy of the rve such that dihedral angles vary out of phase with greater magnitude. However, the inner displacements do not change the frequency of the continuum response nor do they affect the overall displacement magnitude of the response as presented in Fig. 4.4. This leads to the conclusion



(a) Continuum Results without η^d

(b) Continuum Results with η^d



(c) LAMMPS Results

Figure 4.6: The concurrent multiscale continuum model exhibits stiff behavior when compared to the LAMMPS MD simulations.

that the method imposes additional kinematic constraint which η^d does not alleviate. The only remaining assumption constraining the kinematics includes the use of quadratic shape functions.

Imposing quadratic shape functions for each element creates an artificially stiff protein element. Since quadratic shape functions define the field variables through an element, the curvature and strain values can only vary in a quadratic nature through a single rve. This quadratic variation does not represent the rve well since the protein potential terms through the rve exhibit disparate stiffness values that vary by orders of magnitude. As a result, the shape functions constrain atom movement in an rve and the system exhibits an artificial stiffness. Figure 4.7 illustrates an rve and a representative spring model of the rve. Since the stiffnesses in the rve vary greatly, a quadratic shape stiffens the model, while one with more variation similar to the green curve in Fig. 4.7 could correctly model the rve. As shown

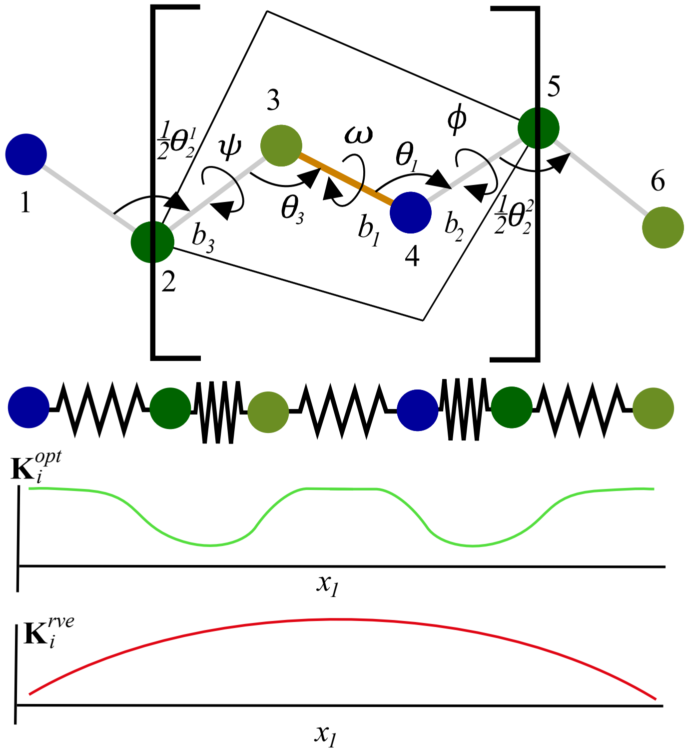


Figure 4.7: The atomistic rve can be represented by a composite spring model with varying stiffnesses. The quadratic shape function shown in red cannot correctly capture the relative displacement of atoms internal to the rve; however, an optimal shape function similar to the one in green may allow the correct relative displacements.

Table 4.1: Derivatives of the rve Potential terms with respect to Intrinsic Curvatures

Dihedral Derivative	Value $[\text{Deg Å}]$	Angle Derivative	Value$[\text{Deg Å}]$
$\frac{\partial\phi}{\partial\mathbf{K}_1}$	1.84	$\frac{\partial\theta_1}{\partial\mathbf{K}_1}$	0.57
$\frac{\partial\phi}{\partial\mathbf{K}_2}$	-1.59	$\frac{\partial\theta_1}{\partial\mathbf{K}_2}$	-1.24
$\frac{\partial\phi}{\partial\mathbf{K}_3}$	1.75	$\frac{\partial\theta_1}{\partial\mathbf{K}_3}$	-0.93
$\frac{\partial\psi}{\partial\mathbf{K}_1}$	2.10	$\frac{\partial\theta_2}{\partial\mathbf{K}_1}$	0.59
$\frac{\partial\psi}{\partial\mathbf{K}_2}$	1.24	$\frac{\partial\theta_2}{\partial\mathbf{K}_2}$	1.33
$\frac{\partial\psi}{\partial\mathbf{K}_3}$	1.77	$\frac{\partial\theta_2}{\partial\mathbf{K}_3}$	-0.65
$\frac{\partial\omega}{\partial\mathbf{K}_1}$	0.62	$\frac{\partial\theta_3}{\partial\mathbf{K}_1}$	1.51
$\frac{\partial\omega}{\partial\mathbf{K}_2}$	0.30	$\frac{\partial\theta_3}{\partial\mathbf{K}_2}$	-0.02
$\frac{\partial\omega}{\partial\mathbf{K}_3}$	-2.67	$\frac{\partial\theta_3}{\partial\mathbf{K}_3}$	-1.19

in Table 4.1, the derivatives of the protein potential terms with respect to $\mathbf{K}$ demonstrate that a change in curvature has a similar effect on dihedral angles as it does on bond angles, thus stiffening the system. Since the bond angle derivatives are approximately equal to the dihedral angle derivatives, a small change in curvature has similar effects on the bond angles as it does on the dihedral angles. However, the bond angle stiffness coefficients are much larger than the dihedral angle stiffness coefficients as shown in Table 3.1. As a result, a small curvature change causes an inaccurate increase in the atomic potential of the rve causing the over stiffening of the element.

Traditionally, increasing mesh density allows more variation through a continuum when using FEA; however, with the rve implementation this is not the case. The rve only realizes the quadratic shape function through the peptide plane since the shape functions are used in the expansions used to derive Eq. (4.4). As a result, increasing mesh density does not remove the kinematic constraints imposed by the shape function.

4.4 Conclusion

This chapter presents a method for simulating the dynamics of a simplified protein backbone using a concurrent multiscale continuum model. The method utilizes atomic potential terms to calculate internal forces in the continuum model by using material points to locate individual atoms when needed. Representative volume elements allow sampling and averaging of energy through the element at integration points since the rve is representative of the system at any point. The beam representation of the protein is discretized into three-noded finite elements with quadratic shape functions. As a result, the quadratic shape functions imposed on the system cause an over stiffening of the element by not allowing the required curvature and strain variation across the rve. Since a peptide plane consists of protein parameters with disparate stiffness coefficients, the curvatures and strains must be able to vary considerably through an rve to accurately allow correct atom movement. Chapter 5 discusses a possible solution to overcome this artificial stiffening of the system.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

Accurately and efficiently modeling protein dynamics presents a formidable challenge; however, the solution has the potential to aid in the design of novel therapeutics for debilitating diseases and novel molecular machines. Since proteins govern most processes in biological life, understanding and predicting how they function in their native environment will lead to the development of new medicines for diseases with limited treatments. Additionally, efficient protein modeling will enable the design of novel molecular machines for uses such as chemical processing and computing. Although the physical and chemical processes governing the protein dynamics problem are well understood, the large scale of the problem and the extremely small time steps required create a prohibitively computationally expensive model when using conventional modeling techniques and computing resources. Therefore, novel models are required which can simulate proteins accurately while reducing the degrees of freedom and increasing the allowable time steps of the problem. This thesis presents two preliminary protein dynamics models utilizing intrinsic beam theory that work towards that goal.

5.1 Hierarchical Multiscale Continuum Model Conclusions and Future Work

The first proposed model utilizes a hierarchical multiscale continuum technique to simulate the dynamics of prototypical α-helical protein segments. A novel shooting method developed for calculating the solution to static three-dimensional intrinsic beam BVP's allows the efficient calculation of deformed curved beams to follower loads with zero iteration. As a result, an optimization algorithm efficiently determines the optimal intrinsic stiffness values for a helical beam such that the deformed configuration of the beam corresponds with the deformed configuration of an MD model for a variety of calibration load cases. This optimal stiffness accurately governs the hierarchical multiscale model's response to external loading, as demonstrated by static and dynamic prediction results which compare well to LAMMPS MD simulations. Furthermore, the hierarchical multiscale model produces the results with a larger allowable time step than the LAMMPS models. The number of degrees of freedom could be reduced significantly with the development of a specialized element which can accurately represent the sinusoidal variance through the helix. Although these results are promising, work must be done to improve the model and adapt it to model full proteins.

The hierarchical multiscale model could be improved and extended to model full protein segments. The model can be improved by determining a more optimal protein stiffness. An apparent stiffness error is evident in both the static and dynamic test cases for the developed model since the displacement and frequency errors range from 3% to 10%. With a more robust optimization algorithm, an equivalent protein stiffness could be located that yields more favorable results. After the optimization technique is improved, LAMMPS MD simulations of full protein α-helical segments could be used to determine the equivalent intrinsic stiffness values for these full protein segments. The final step would require extending the model to simulate other general protein secondary structures.

The ultimate goal is to combine this hierarchical code with an MD code in a concurrent multiscale manner. This would involve replacing segments of an all-atom MD model with the hierarchical multiscale model. Due to the larger allowable time steps for the hierarchical model, the implementation of the model in this way could allow partial atomic resolution of the entire protein backbone while decreasing computational expense. Additionally, developing a method to rebuild the entire protein at the atomistic level from the hierarchical multiscale model would allow the transition from the hierarchical multiscale model to an all-atom MD model during simulation. This would be useful in dynamics simulations where protein segments that originally are stable secondary structures modeled with the hierarchical multiscale model become more active such that all-atom resolution is desired. The computational expense is reduced by allowing the hierarchical models to use larger a time step such that fewer degrees of freedom are updated each MD time step.

5.2 Preliminary Concurrent Multiscale Continuum Model Conclusions and Future Work

The second proposed solution to the protein dynamics problem involved the development of a concurrent multiscale model. The preliminary concurrent multiscale model links atomistic behavior of the protein to the bulk material properties of the continuum beam through the use of an rve. The rve represents the internal energy of the continuum at any point and allows the calculation of atom positions using the continuum field variables $\mathbf{K}$ and γ. As a result, atomistic information is used concurrently with the continuum beam model to simulate the model's dynamic behavior. The rve implemented in the finite element model caused an artificially stiff peptide plane since the finite element discretization of the protein utilizes quadratic shape functions. These shape functions restrict the curvature and strain variation through the rve such that the correct stiffness response cannot be modeled.

The preliminary concurrent multiscale model must be improved such that sufficient variation of curvature and strain can be modeled through the rve. One possible solution is to modify the shape function of a beam element such that the required variation through the rve is satisfied. A multiscale finite element has recently been designed which facilitates the use of arbitrary shape functions in an element such that higher variation is obtained where desired [12]. A similar technique could be explored for use in the preliminary concurrent multiscale technique which allows more variation through an rve.

After the artificial stiffness issue has been resolved, the concurrent multiscale continuum model can be extended for use in modeling general proteins. This will require the addition of side chain atoms and residues to the rve. Additionally, multiple rves should be defined and their locations in the continuum model should be stored so that an arbitrary protein containing any of the 20 amino acid residues given in Table 1.1 can be modeled. With the addition of residues and side chains, a proximity search will need to be implemented such that hydrogen bonding and other nonbonded protein potential terms can be accurately modeled.

The concurrent multiscale model could be improved further by decreasing the computational expense of the developed protein finite element. Computational expense could be lessened by reducing the length of the rve such that a smaller dx_1 allows a lower order expansion. The current required expansion to $O(dx_1^9)$ results in a large equation for calculating atomic potentials, and reducing the size of this equation significantly reduces computational cost. Additionally, the stiff protein potential bond terms could be treated as rigid bodies which would allow the use of larger time steps. With the aforementioned improvements, the concurrent multiscale continuum model could accurately and efficiently model proteins when compared to traditional MD modeling packages.

5.3 Mass Representation Improvement

The continuum model mass terms could be improved to better represent the discrete protein. Two approaches should be explored: an optimization of mass terms and the development of an implicit FEA solver for the intrinsic beam code. The products of inertia of the beam model are non-negligible, but are ignored in order to use an explicit FEA code. This could be accounted for by running an optimization that varied the continuum mass terms until the frequency content of the hierarchical multiscale model more closely matches LAMMPS results. Another solution is to reprogram the FEA code with an implicit solver so that all mass terms can be considered.

APPENDIX A

SHOOTING EQUATION DERIVATIVES

A.1 Explicit Definition of Derivatives for Equations with Strain

$$\frac{\partial \widehat{K}_1}{\partial x_1} = \frac{E\left[i_2\left(K_{0_3}+\widehat{K}_3\right)\widehat{K}_2 - i_3\left(K_{0_2}+\widehat{K}_2\right)\widehat{K}_3\right]}{G\left(i_2+i_3\right)} \tag{A.1}$$

$$\frac{\partial \widehat{K}_2}{\partial x_1} = \frac{\left[Ei_3\left(K_{0_1}+\widehat{K}_1\right)\widehat{K}_3 - G\left(i_2+i_3\right)\left(K_{0_3}+\widehat{K}_3\right)\widehat{K}_1 - EA\widehat{\gamma}_{13}\widehat{\gamma}_{11} + 2\,\frac{GA(1+\widehat{\gamma}_{11})\widehat{\gamma}_{13}}{k}\right]}{Ei_2} \tag{A.2}$$

$$\frac{\partial \widehat{K}_3}{\partial x_1} = \frac{\left[\left(i_2+i_3\right)G\left(K_{0_2}+\widehat{K}_2\right)\widehat{K}_1 - Ei_2\left(K_{0_1}+\widehat{K}_1\right)\widehat{K}_2 - 2\,\frac{GA(1+\widehat{\gamma}_{11})\widehat{\gamma}_{12}}{k} + EA\widehat{\gamma}_{12}\widehat{\gamma}_{11}\right]}{Ei_3} \tag{A.3}$$

$$\frac{\partial \widehat{\gamma}_{11}}{\partial x_1} = \frac{2\,G\left[\left(K_{0_3}+\widehat{K}_3\right)\widehat{\gamma}_{12} - \left(K_{0_2}+\widehat{K}_2\right)\widehat{\gamma}_{13}\right]}{Ek} \tag{A.4}$$

$$\frac{\partial \widehat{\gamma}_{12}}{\partial x_1} = \frac{2\,G\left(K_{0_1}+\widehat{K}_1\right)\widehat{\gamma}_{13} - Ek\left(K_{0_3}+\widehat{K}_3\right)\widehat{\gamma}_{11}}{2G} \tag{A.5}$$

$$\frac{\partial \widehat{\gamma}_{13}}{\partial x_1} = \frac{Ek\left(K_{0_2}+\widehat{K}_2\right)\widehat{\gamma}_{11} - 2\,G\left(K_{0_1}+\widehat{K}_1\right)\widehat{\gamma}_{12}}{2G} \tag{A.6}$$

$$\frac{\partial B_{1_x}}{\partial x_1} = \left(K_{0_2}+\widehat{K}_2\right)B_{2_y}B_{1_z} - \left(K_{0_2}+\widehat{K}_2\right)B_{2_z}B_{1_y} + \left(K_{0_3}+\widehat{K}_3\right)B_{3_y}B_{1_z} - \left(K_{0_3}+\widehat{K}_3\right)B_{3_z}B_{1_y} \tag{A.7}$$

$$\frac{\partial B_{1_y}}{\partial x_1} = \left(K_{0_2}+\widehat{K}_2\right)B_{2_z}B_{1_x} - \left(K_{0_2}+\widehat{K}_2\right)B_{2_x}B_{1_z} + \left(K_{0_3}+\widehat{K}_3\right)B_{3_z}B_{1_x} - \left(K_{0_3}+\widehat{K}_3\right)B_{3_x}B_{1_z} \tag{A.8}$$

$$\frac{\partial B_{1_z}}{\partial x_1} = \left(K_{0_2}+\widehat{K}_2\right)B_{2_x}B_{1_y} - \left(K_{0_2}+\widehat{K}_2\right)B_{2_y}B_{1_x} + \left(K_{0_3}+\widehat{K}_3\right)B_{3_x}B_{1_y} - \left(K_{0_3}+\widehat{K}_3\right)B_{3_y}B_{1_x} \tag{A.9}$$

$$\frac{\partial B_{2_x}}{\partial x_1} = \left(K_{0_1}+\widehat{K}_1\right)B_{1_y}B_{2_z} - \left(K_{0_1}+\widehat{K}_1\right)B_{1_z}B_{2_y} + \left(K_{0_3}+\widehat{K}_3\right)B_{3_y}B_{2_z} - \left(K_{0_3}+\widehat{K}_3\right)B_{3_z}B_{2_y} \tag{A.10}$$

$$\frac{\partial B_{2_y}}{\partial x_1} = \left(K_{0_1}+\widehat{K}_1\right)B_{1_z}B_{2_x} - \left(K_{0_1}+\widehat{K}_1\right)B_{1_x}B_{2_z} + \left(K_{0_3}+\widehat{K}_3\right)B_{3_z}B_{2_x} - \left(K_{0_3}+\widehat{K}_3\right)B_{3_x}B_{2_z} \tag{A.11}$$

$$\frac{\partial B_{2_z}}{\partial x_1} = \left(K_{0_1}+\widehat{K}_1\right)B_{1_x}B_{2_y} - \left(K_{0_1}+\widehat{K}_1\right)B_{1_y}B_{2_x} + \left(K_{0_3}+K_3\right)B_{3_x}B_{2_y} - \left(K_{0_3}+\widehat{K}_3\right)B_{3_y}B_{2_x} \tag{A.12}$$

$$\frac{\partial B_{3_x}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_y} B_{3_z} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_z} B_{3_y} + \left(K_{0_2} + \widehat{K}_2\right) B_{2_y} B_{3_z} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_z} B_{3_y} \quad (A.13)$$

$$\frac{\partial B_{3_y}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_z} B_{3_x} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_x} B_{3_z} + \left(K_{0_2} + \widehat{K}_2\right) B_{2_z} B_{3_x} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_x} B_{3_z} \quad (A.14)$$

$$\frac{\partial B_{3_z}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_x} B_{3_y} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_y} B_{3_x} + \left(K_{0_2} + \widehat{K}_2\right) B_{2_x} B_{3_y} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_y} B_{3_x} \quad (A.15)$$

$$\frac{\partial R_x}{\partial x_1} = \left(1 + \widehat{\gamma}_{11}\right) B_{1_x} + 2\,\widehat{\gamma}_{12} B_{2_x} + 2\,\widehat{\gamma}_{13} B_{3_x} \quad (A.16)$$

$$\frac{\partial R_y}{\partial x_1} = \left(1 + \widehat{\gamma}_{11}\right) B_{1_y} + 2\,\widehat{\gamma}_{12} B_{2_y} + 2\,\widehat{\gamma}_{13} B_{3_y} \quad (A.17)$$

$$\frac{\partial R_z}{\partial x_1} = \left(1 + \widehat{\gamma}_{11}\right) B_{1_z} + 2\,\widehat{\gamma}_{12} B_{2_z} + 2\,\widehat{\gamma}_{13} B_{3_z} \quad (A.18)$$

A.2 Explicit Definition of Derivatives for Equations without Strain

$$\frac{\partial \widehat{K}_1}{\partial x_1} = \frac{E\left[i_2 \left(K_{0_3} + \widehat{K}_3\right) \widehat{K}_2 - i_3 \left(K_{0_2} + \widehat{K}_2\right) \widehat{K}_3\right]}{G\left(i_2 + i_3\right)} \quad (A.19)$$

$$\frac{\partial \widehat{K}_1}{\partial x_1} = \frac{E\left[i_2 \left(K_{0_3} + \widehat{K}_3\right) \widehat{K}_2 - i_3 \left(K_{0_2} + \widehat{K}_2\right) \widehat{K}_3\right]}{G\left(i_2 + i_3\right)} \quad (A.20)$$

$$\frac{\partial \widehat{K}_2}{\partial x_1} = \frac{E i_3 \left(K_{0_1} + \widehat{K}_1\right) \widehat{K}_3 - G\left(i_2 + i_3\right) \left(K_{0_3} + \widehat{K}_3\right) \widehat{K}_1 - F_3 F_1 + \left(1 + F_1\right) F_3}{E i_2} \quad (A.21)$$

$$\frac{\partial \widehat{K}_3}{\partial x_1} = \frac{G\left(i_2 + i_3\right) \left(K_{0_2} + \widehat{K}_2\right) \widehat{K}_1 - E i_2 \left(K_{0_1} + \widehat{K}_1\right) \widehat{K}_2 - \left(1 + F_1\right) F_2 + F_2 F_1}{E i_3} \quad (A.22)$$

$$\frac{\partial F_1}{\partial x_1} = \left(K_{0_3} + \widehat{K}_3\right) F_2 - \left(K_{0_2} + \widehat{K}_2\right) F_3 \quad (A.23)$$

$$\frac{\partial F_1}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) F_3 - \left(K_{0_3} + \widehat{K}_3\right) F_1 \quad (A.24)$$

$$\frac{\partial F_1}{\partial x_1} = \left(K_{0_2} + \widehat{K}_2\right) F_1 - \left(K_{0_1} + \widehat{K}_1\right) F_2 \quad (A.25)$$

$$\frac{\partial B_{1_x}}{\partial x_1} = \left(K_{0_2} + \widehat{K}_2\right) B_{2_y} B_{1_z} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_z} B_{1_y} + \left(K_{0_3} + \widehat{K}_3\right) B_{3_y} B_{1_z} - \left(K_{0_3} + \widehat{K}_3\right) B_{3_z} B_{1_y} \quad (A.26)$$

$$\frac{\partial B_{1_y}}{\partial x_1} = \left(K_{0_2} + \widehat{K}_2\right) B_{2_z} B_{1_x} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_x} B_{1_z} + \left(K_{0_3} + \widehat{K}_3\right) B_{3_z} B_{1_x} - \left(K_{0_3} + \widehat{K}_3\right) B_{3_x} B_{1_z} \quad (A.27)$$

$$\frac{\partial B_{1_z}}{\partial x_1} = \left(K_{0_2} + \widehat{K}_2\right) B_{2_x} B_{1_y} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_y} B_{1_x} + \left(K_{0_3} + \widehat{K}_3\right) B_{3_x} B_{1_y} - \left(K_{0_3} + \widehat{K}_3\right) B_{3_y} B_{1_x} \quad (A.28)$$

$$\frac{\partial B_{2_x}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_y} B_{2_z} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_z} B_{2_y} + \left(K_{0_3} + \widehat{K}_3\right) B_{3_y} B_{2_z} - \left(K_{0_3} + \widehat{K}_3\right) B_{3_z} B_{2_y} \quad (A.29)$$

$$\frac{\partial B_{2_y}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_z} B_{2_x} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_x} B_{2_z} + \left(K_{0_3} + \widehat{K}_3\right) B_{3_z} B_{2_x} - \left(K_{0_3} + \widehat{K}_3\right) B_{3_x} B_{2_z} \quad (A.30)$$

$$\frac{\partial B_{2_z}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_x} B_{2_y} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_y} B_{2_x} + \left(K_{0_3} + K_3\right) B_{3_x} B_{2_y} - \left(K_{0_3} + \widehat{K}_3\right) B_{3_y} B_{2_x} \quad (A.31)$$

$$\frac{\partial B_{3_x}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_y} B_{3_z} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_z} B_{3_y} + \left(K_{0_2} + \widehat{K}_2\right) B_{2_y} B_{3_z} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_z} B_{3_y} \quad (A.32)$$

$$\frac{\partial B_{3_y}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_z} B_{3_x} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_x} B_{3_z} + \left(K_{0_2} + \widehat{K}_2\right) B_{2_z} B_{3_x} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_x} B_{3_z} \quad \text{(A.33)}$$

$$\frac{\partial B_{3_z}}{\partial x_1} = \left(K_{0_1} + \widehat{K}_1\right) B_{1_x} B_{3_y} - \left(K_{0_1} + \widehat{K}_1\right) B_{1_y} B_{3_x} + \left(K_{0_2} + \widehat{K}_2\right) B_{2_x} B_{3_y} - \left(K_{0_2} + \widehat{K}_2\right) B_{2_y} B_{3_x} \quad \text{(A.34)}$$

$$\frac{\partial R_x}{\partial x_1} = B_{1_x} \quad \text{(A.35)}$$

$$\frac{\partial R_y}{\partial x_1} = B_{1_y} \quad \text{(A.36)}$$

$$\frac{\partial R_z}{\partial x_1} = B_{1_z} \quad \text{(A.37)}$$

APPENDIX B

LAMMPS ‘QUADRATIC’ DIHEDRAL CLASS

B.1 Dihedral_Quadratic.h

```
#ifdef DIHEDRAL_CLASS

DihedralStyle(quadratic,DihedralQuadratic)

#else

#ifndef LMP_DIHEDRAL_QUADRATIC_H
#define LMP_DIHEDRAL_HARMONIC_H

#include "stdio.h"
#include "dihedral.h"

namespace LAMMPS_NS {

class DihedralQuadratic : public Dihedral {
 public:
  DihedralQuadratic(class LAMMPS *);
  ~DihedralQuadratic();
  void compute(int, int);
  void coeff(int, char **);
  void write_restart(FILE *);
  void read_restart(FILE *);

 private:
  double *k,*Equil;

  void allocate();
};
}
#endif
#endif
```

B.2 Dihedral_Quadratic.cpp

```
#include "lmptype.h"
```

```
#include "mpi.h"
#include "math.h"
#include "stdlib.h"
#include "dihedral_quadratic.h"
#include "atom.h"
#include "comm.h"
#include "neighbor.h"
#include "domain.h"
#include "force.h"
#include "update.h"
#include "memory.h"
#include "error.h"

using namespace LAMMPS_NS;

#define TOLERANCE 0.05
#define SMALL     0.001

/* ---------------------------------------------------------------------- */
DihedralQuadratic::DihedralQuadratic(LAMMPS *lmp) : Dihedral(lmp) {}
/* ---------------------------------------------------------------------- */

DihedralQuadratic::~DihedralQuadratic()
{
  if (allocated) {
    memory->sfree(setflag);
    memory->sfree(k);
    memory->sfree(Equil);
    }
}

/* ---------------------------------------------------------------------- */

void DihedralQuadratic::compute(int eflag, int vflag)
{
  int i1,i2,i3,i4,i,m,n,type;
  double vb1x,vb1y,vb1z,vb2x,vb2y,vb2z,vb3x,vb3y,vb3z,vb2xm,vb2ym,vb2zm;
  double edihedral,f1[3],f2[3],f3[3],f4[3];
  double ax,ay,az,bx,by,bz,rasq,rbsq,rgsq,rg,rginv,ra2inv,rb2inv,rabinv;
  double df,df1,ddf1,fg,hg,fga,hgb,gaa,gbb;
  double dtfx,dtfy,dtfz,dtgx,dtgy,dtgz,dthx,dthy,dthz;
```

```
double c,s,p,sx2,sy2,sz2;

//Kyle
double dQuaEn, dihedral, QuaEn;

edihedral = 0.0;
if (eflag || vflag) ev_setup(eflag,vflag);
else evflag = 0;

double **x = atom->x;
double **f = atom->f;
int **dihedrallist = neighbor->dihedrallist;
int ndihedrallist = neighbor->ndihedrallist;
int nlocal = atom->nlocal;
int newton_bond = force->newton_bond;

for (n = 0; n < ndihedrallist; n++) {
  i1 = dihedrallist[n][0];
  i2 = dihedrallist[n][1];
  i3 = dihedrallist[n][2];
  i4 = dihedrallist[n][3];
  type = dihedrallist[n][4];

  // 1st bond


  vb1x = x[i1][0] - x[i2][0];
  vb1y = x[i1][1] - x[i2][1];
  vb1z = x[i1][2] - x[i2][2];
  domain->minimum_image(vb1x,vb1y,vb1z);

  // 2nd bond


  vb2x = x[i3][0] - x[i2][0];
  vb2y = x[i3][1] - x[i2][1];
  vb2z = x[i3][2] - x[i2][2];
  domain->minimum_image(vb2x,vb2y,vb2z);

  vb2xm = -vb2x;
  vb2ym = -vb2y;
  vb2zm = -vb2z;
  domain->minimum_image(vb2xm,vb2ym,vb2zm);
```

```
    // 3rd bond

    vb3x = x[i4][0] - x[i3][0];
    vb3y = x[i4][1] - x[i3][1];
    vb3z = x[i4][2] - x[i3][2];
    domain->minimum_image(vb3x,vb3y,vb3z);

    // c,s calculation

    ax = vb1y*vb2zm - vb1z*vb2ym;
    ay = vb1z*vb2xm - vb1x*vb2zm;
    az = vb1x*vb2ym - vb1y*vb2xm;
    bx = vb3y*vb2zm - vb3z*vb2ym;
    by = vb3z*vb2xm - vb3x*vb2zm;
    bz = vb3x*vb2ym - vb3y*vb2xm;

    rasq = ax*ax + ay*ay + az*az;
    rbsq = bx*bx + by*by + bz*bz;
    rgsq = vb2xm*vb2xm + vb2ym*vb2ym + vb2zm*vb2zm;
    rg = sqrt(rgsq);

    rginv = ra2inv = rb2inv = 0.0;
    if (rg > 0) rginv = 1.0/rg;
    if (rasq > 0) ra2inv = 1.0/rasq;
    if (rbsq > 0) rb2inv = 1.0/rbsq;
    rabinv = sqrt(ra2inv*rb2inv);

    c = (ax*bx + ay*by + az*bz)*rabinv;
    s = rg*rabinv*(ax*vb3x + ay*vb3y + az*vb3z);

    // error check

    if (c > 1.0 + TOLERANCE || c < (-1.0 - TOLERANCE)) {
      int me;
      MPI_Comm_rank(world,&me);
      if (screen) {
char str[128];
sprintf(str,"Dihedral problem: %d %d %d %d %d %d",
me,update->ntimestep,
atom->tag[i1],atom->tag[i2],atom->tag[i3],atom->tag[i4]);
```

```
error->warning(str);
fprintf(screen,"  1st atom: %d %g %g %g\n",
me,x[i1][0],x[i1][1],x[i1][2]);
fprintf(screen,"  2nd atom: %d %g %g %g\n",
me,x[i2][0],x[i2][1],x[i2][2]);
fprintf(screen,"  3rd atom: %d %g %g %g\n",
me,x[i3][0],x[i3][1],x[i3][2]);
fprintf(screen,"  4th atom: %d %g %g %g\n",
me,x[i4][0],x[i4][1],x[i4][2]);
      }
    }

    if (c > 1.0) c = 1.0;
    if (c < -1.0) c = -1.0;

   dihedral=atan2(s,c);

   if(dihedral < 0 && type == 1)
   {
       dihedral = dihedral + 2*PI;
   }

   dQuaEn = 2.0*k[type] * (dihedral - Equil[type]);

   QuaEn = k[type] * (dihedral - Equil[type])* (dihedral - Equil[type]);

    if (eflag) edihedral = QuaEn;

    fg = vb1x*vb2xm + vb1y*vb2ym + vb1z*vb2zm; // dot product of 1 st bond onto -2nd bond
    hg = vb3x*vb2xm + vb3y*vb2ym + vb3z*vb2zm; // dot product of 3 rd bond onto -2nd bond
    fga = fg*ra2inv*rginv;  // (dot product of 1st bond onto -2nd bond)/magnitude 2nd bond and mag of b1 x b2 squared
    hgb = hg*rb2inv*rginv;  // (dot product of 3rd bond onto -2nd bond)/magnitude 2nd bond and mag of b3 x b2 squared
    gaa = -ra2inv*rg;
    gbb = rb2inv*rg;

    dtfx = gaa*ax;
    dtfy = gaa*ay;
    dtfz = gaa*az;
    dtgx = fga*ax - hgb*bx;
    dtgy = fga*ay - hgb*by;
    dtgz = fga*az - hgb*bz;
```

```
dthx = gbb*bx;
dthy = gbb*by;
dthz = gbb*bz;

df = -dQuaEn;

sx2 = df*dtgx;
sy2 = df*dtgy;
sz2 = df*dtgz;

f1[0] = df*dtfx;
f1[1] = df*dtfy;
f1[2] = df*dtfz;

f2[0] = sx2 - f1[0];
f2[1] = sy2 - f1[1];
f2[2] = sz2 - f1[2];

f4[0] = df*dthx;
f4[1] = df*dthy;
f4[2] = df*dthz;

f3[0] = -sx2 - f4[0];
f3[1] = -sy2 - f4[1];
f3[2] = -sz2 - f4[2];

// apply force to each of 4 atoms

if (newton_bond || i1 < nlocal) {
  f[i1][0] += f1[0];
  f[i1][1] += f1[1];
  f[i1][2] += f1[2];
}

if (newton_bond || i2 < nlocal) {
  f[i2][0] += f2[0];
  f[i2][1] += f2[1];
  f[i2][2] += f2[2];
}

if (newton_bond || i3 < nlocal) {
```

```
      f[i3][0] += f3[0];
      f[i3][1] += f3[1];
      f[i3][2] += f3[2];
    }

    if (newton_bond || i4 < nlocal) {
      f[i4][0] += f4[0];
      f[i4][1] += f4[1];
      f[i4][2] += f4[2];
    }

    if (evflag)
      ev_tally(i1,i2,i3,i4,nlocal,newton_bond,edihedral,f1,f3,f4,
       vb1x,vb1y,vb1z,vb2x,vb2y,vb2z,vb3x,vb3y,vb3z);
  }
}

/* ---------------------------------------------------------------------- */

void DihedralQuadratic::allocate()
{
  allocated = 1;
  int n = atom->ndihedraltypes;

  k = (double *) memory->smalloc((n+1)*sizeof(double),"dihedral:k");
  Equil = (double *) memory->smalloc((n+1)*sizeof(double),"dihedral:Equilibrium");

  setflag = (int *) memory->smalloc((n+1)*sizeof(int),"dihedral:setflag");
  for (int i = 1; i <= n; i++) setflag[i] = 0;
}

/* ----------------------------------------------------------------------
   set coeffs for one type
------------------------------------------------------------------------- */

void DihedralQuadratic::coeff(int narg, char **arg)
{
  if (narg != 3) error->all("Incorrect args for dihedral coefficients");
  if (!allocated) allocate();

  int ilo,ihi;
```

```
  force->bounds(arg[0],atom->ndihedraltypes,ilo,ihi);

  double k_one = force->numeric(arg[1]);
  double Equil_one = force->numeric(arg[2]);

  if (k<=0)
    error->all("Incorrect dihedral potential constant");

  int count = 0;
  for (int i = ilo; i <= ihi; i++) {
    k[i] = k_one;
    Equil[i] = Equil_one*PI/180;
    setflag[i] = 1;
    count++;
  }

  if (count == 0) error->all("Incorrect args for dihedral coefficients");
}

/* ----------------------------------------------------------------------
   proc 0 writes out coeffs to restart file
------------------------------------------------------------------------- */

void DihedralQuadratic::write_restart(FILE *fp)
{
  fwrite(&k[1],sizeof(double),atom->ndihedraltypes,fp);
  fwrite(&Equil[1],sizeof(int),atom->ndihedraltypes,fp);
}

/* ----------------------------------------------------------------------
   proc 0 reads coeffs from restart file, bcasts them
------------------------------------------------------------------------- */

void DihedralQuadratic::read_restart(FILE *fp)
{
  allocate();

  if (comm->me == 0) {
    fread(&k[1],sizeof(double),atom->ndihedraltypes,fp);
    fread(&Equil[1],sizeof(double),atom->ndihedraltypes,fp);
  }
```

```
  MPI_Bcast(&k[1],atom->ndihedraltypes,MPI_DOUBLE,0,world);
  MPI_Bcast(&Equil[1],atom->ndihedraltypes,MPI_INT,0,world);

  for (int i = 1; i <= atom->ndihedraltypes; i++) {
    setflag[i] = 1;
      }
}
```

REFERENCES

[1] Al-Sadder, S. and Al-Rawi, R., "Finite difference scheme for large-deflection analysis of non-prismatic cantilever beams subjected to different types of continuous and discontinuous loadings," *Archive of Applied Mechanics*, vol. 75, pp. 459–473, 2006. 10.1007/s00419-005-0422-5.

[2] Anfinsen, C., "Principles that Govern Folding of Protein Chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[3] "National resource for biomedical supercomputing." NRBS, url-http://www.nrbsc.org/antonrfp/, Feb. 2012.

[4] Argyris, J. and Symeonidis, S., "Nonlinear finite element analysis of elastic systems under nonconservative loading-Natural formulation. Part I. Quasistatic problems," *Computer Methods in Applied Mechanics and*, vol. 26, pp. 75–123, 1981.

[5] Aurora, R., Srinivasan, R., and Rose, G. D., "Rules for α-helix termination by glycine," *Science*, vol. 264, no. 5162, pp. pp. 1126–1130, 1994.

[6] Barral, J. M., Broadley, S. A., Schaffar, G., and Hartl, F., "Roles of molecular chaperones in protein misfolding diseases," *Seminars in Cell & Developmental Biology*, vol. 15, no. 1, pp. 17 – 29, 2004. Protein Misfolding and Human Disease and Developmental Biology of the Retina.

[7] Barten, H., "On the deflection of a cantilever beam," *Quarterly of Applied Math*, vol. 2, pp. 168–171, 1944.

[8] Bekard, I. B., Asimakis, P., Bertolini, J., and Dunstan, D. E., "The effects of shear flow on protein structure and function," *Biopolymers*, vol. 95, no. 11, pp. 733–745, 2011.

[9] Bisshopp, K. E. and Drucker, D. C., "Large deflections of cantilever beams," *Quarterly of Applied Math*, vol. 3, pp. 272–275, 1945.

[10] Brooks, B., Bruccoleri, R., Olafson, D., States, D., Swaminathan, S., and Karplus, M., "Charmm: A program for macromolecular energy, minimization, and dynamics calculations," *Journal of Computational Chemistry*, vol. 4, pp. 187–217, 1983.

[11] Buehler, M. J., "Atomistic and continuum modeling of mechanical properties of collagen: Elasticity, fracture, and self-assembly," *Journal of Materials Research*, vol. 21, pp. 1947–1961, 2006.

[12] Casadei, F. and Ruzzene, M., "Application of the multi-scale finite element method to wave propagation problems in damaged structures," in *Proc. SPIE 7984, 79842Q*, March 2011.

[13] Case, D. A. and et. al., "The amber biomolecular simulation programs," *Journal of Computational Chemistry*, vol. 26, no. 16, pp. 1668–1688, 2005.

[14] Chan, Hue SunDill, K. A., "The protein folding problem.," *Physics Today*, vol. 46, no. 2, p. 24, 1993.

[15] Chiti, F. and Dobson, C. M., "Protein misfolding, functional amyloid, and human disease," *Annual Review of Biochemistry*, vol. 75, pp. 333–366, 2006.

[16] Cohen, Fred E.Kelly, J. W., "Therapeutic approaches to protein-misfolding diseases.," *Nature*, vol. 426, no. 6968, pp. 905 – 909, 2003.

[17] Conrad, M., "On design principles for a molecular computer," *Commun. ACM*, vol. 28, pp. 464–480, May 1985.

[18] Cook, P., *Nonlinear Dynamical Systems.* Prentice Hall International, 1986.

[19] Dewitte, R. S. and Shakhnovich, E. I., "Pseudodihedrals: Simplified protein backbone representation with knowledge-based energy," *Protein Science*, vol. 3, no. 9, pp. 1570–1581, 1994.

[20] Edwards, C. and Penney, D., *Elementary Differential Equations with Boundary Value Problems.* Pearson Prentice Hall, 5^{th} ed., 2004.

[21] Floudas, C., Fung, H., McAllister, S., Mnnigmann, M., and Rajgaria, R., "Advances in protein structure prediction and de novo protein design: A review," *Chemical Engineering Science*, vol. 61, no. 3, pp. 966 – 988, 2006.

[22] Frauenfelder, H., "The secondary structure," in *The Physics of Proteins* (Chan, S. S. and Chan, W. S., eds.), Biological and Medical Physics, Biomedical Engineering, pp. 65–73, Springer New York, 2010.

[23] Gatti-Bono, C. and Perkins, N., "Physical and numerical modeling of the dynamic behavior of a fly line," *Journal of Sound and Vibration*, vol. 255, no. 3, pp. 555 – 577, 2002.

[24] Goyal, S., Lillian, T., Blumberg, S., Meiners, J.-C., Meyhfer, E., and Perkins, N., "Intrinsic curvature of dna influences lacr-mediated looping," *Biophysical Journal*, vol. 93, no. 12, pp. 4342 – 4359, 2007.

[25] Henstra, A. M., Sipma, J., Rinzema, A., and Stams, A. J., "Microbiology of synthesis gas fermentation for biofuel production," *Current Opinion in Biotechnology*, vol. 18, no. 3, pp. 200 – 206, 2007.

[26] Herrmann, G., "Stability of equilibrium of elastic systems subjected to nonconservative forces," *Applied Mechanics Reviews*, vol. 20, no. 2, pp. 103–108, 1967.

[27] Hodges, D., "Geometrically exact, intrinsic theory for dynamics of curved and twisted anisotropic beams," *AIAA Journal*, vol. 41, no. 6, pp. 1131–1137, 2003.

[28] Honig, B. and Yang, A.-S., "Free energy balance in protein folding," in *Protein Stability* (C.B. Anfinsen, Frederic M. Richards, J. T. E. and Eisenberg, D. S., eds.), vol. 46 of *Advances in Protein Chemistry*, pp. 27 – 58, Academic Press, 1995.

[29] Hughes, T., *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover, 2000.

[30] K.Eric and Drexler, "Building molecular machine systems," *Trends in Biotechnology*, vol. 17, no. 1, pp. 5 – 7, 1999.

[31] Leamy, M., "Bulk dynamic response of modeling of carbon nanotubes using an intrinsic finite element formulation incorporationg interatomic potentials," *International Journal of Solids and Structures*, vol. 44, pp. 874–894, 2006.

[32] Leamy, M., "Intrinsic finite element modeling of nonlinear dynamic response in helical springs," *Journal of Computational and Nonlinear Dynamics*, in press.

[33] Lee, K., "Large deflections of cantilever beams of non-linear elastic material under a combined loading," *International Journal of Non-Linear Mechanics*, vol. 37, pp. 439–443, Apr. 2002.

[34] Lindorff-Larsen, K., Piana, S., Dror, R. O., and Shaw, D. E., "How fast-folding proteins fold," *Science*, vol. 334, no. 6055, pp. 517–520, 2011.

[35] Lou, J. and Zhu, C., "Flow induces loop-to--hairpin transition on the -switch of platelet glycoprotein iba," *Proceedings of the National Academy of Sciences*, vol. 105, no. 37, pp. 13847–13852, 2008.

[36] MacKerel Jr., A., Brooks III, C., Nilsson, L., Roux, B., Won, Y., and Karplus, M., *CHARMM: The Energy Function and Its Parameterization with an Overview of the Program*, vol. 1 of *The Encyclopedia of Computational Chemistry*, pp. 271–277. John Wiley & Sons: Chichester, 1998.

[37] MacKerell, A. D. and et. al., "All-atom empirical potential for molecular modeling and dynamics studies of proteins," *The Journal of Physical Chemistry B*, vol. 102, no. 18, pp. 3586–3616, 1998.

[38] Miller, R., Ortiz, M., Phillips, R., Shenoy, V., and Tadmor, E., "Quasicontinuum models of fracture and plasticity," *Engineering Fracture Mechanics*, vol. 61, no. 34, pp. 427 – 444, 1998.

[39] Mutyalarao, M., Bharathi, D., and Nageswara Rao, B., "On the uniqueness of large deflections of a uniform cantilever beam under a tip-concentrated rotational load," *International Journal of Non-Linear Mechanics*, vol. 45, pp. 433–441, May 2010.

[40] Neri, M., Anselmi, C., Cascella, M., Maritan, A., and Carloni, P., "Coarse-grained model of proteins incorporating atomistic detail of the active site," *Phys. Rev. Lett.*, vol. 95, p. 218102, Nov 2005.

[41] Neumaier, A., "Molecular modeling of proteins and mathematical prediction of protein structure," *SIAM Review*, vol. 39, pp. 407–460, 1997.

[42] Nikolai, E., "On the stabilty of the rectilinear form of a compressed and twisted bar," *Izvest. Leningr. Politekhn*, vol. 31, 1928.

[43] Nikolai, E., "On the problem of stability of a twisted bar," *Vestnik prikl. mat. i mekh.*, vol. 1, 1929.

[44] Odegard, G. M., Gates, T. S., Nicholson, L. M., and Wise, K. E., "Equivalent-continuum modeling of nano-structured materials," *Composites Science and Technology*, vol. 62, no. 14, pp. 1869 – 1880, 2002.

[45] Pai, P. and Lee, S., "Three-dimensional postbuckling analysis of curved beams," *International Journal of Computational*, vol. 3, no. 3, pp. 305–338, 2002.

[46] Pai, P. and Palazotto, A. N., "Large Deformation Analysis of Flexible Beams," *International Journal of Solids and Structures*, vol. 33, no. 9, pp. 1335–1353, 1996.

[47] Pastore, A. and Lesk, A. M., *Protein Structure.* John Wiley & Sons, Ltd, 2005.

[48] "Residue count histogram." RCSB Protein Data Bank, urlhttp://www.pdb.org/, Feb. 2012.

[49] Peng, J. and Xu, J., "Raptorx: Exploiting structure information for protein alignment by statistical inference," *Proteins: Structure, Function, and Bioinformatics*, vol. 79, no. S10, pp. 161–171, 2011.

[50] Pieper, D. H. and Reineke, W., "Engineering bacteria for bioremediation," *Current Opinion in Biotechnology*, vol. 11, no. 3, pp. 262 – 270, 2000.

[51] Plimpton, S., "Fast parallel algorithms for short-range molecular dynamics," *Journal of Computational Physics*, vol. 117, pp. 1–19, 1995.

[52] Ponder, J. W. and Case, D. A., "Force fields for protein simulations," in *Protein Simulations* (Daggett, V., ed.), vol. 66 of *Advances in Protein Chemistry*, pp. 27 – 85, Academic Press, 2003.

[53] Praprotnik, M., Site, L. D., and Kremer, K., "Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly.," *Journal of Chemical Physics*, vol. 123, no. 22, p. 224106, 2005.

[54] Roberts, S. and Shipman, J., *Two-point Boundary Value Problems: Shooting Methods.* Elsevier, 1972.

[55] Rohl, C. A., Strauss, C. E., Misura, K. M., and Baker, D., "Protein structure prediction using rosetta," in *Numerical Computer Methods, Part D* (Brand, L. and Johnson, M. L., eds.), vol. 383 of *Methods in Enzymology*, pp. 66 – 93, Academic Press, 2004.

[56] Roos, W., Gibbons, M., Arkhipov, A., Uetrecht, C., and et. al., "Squeezing protein shells: How continuum elastic models, molecular dynamics simulations, and experiments coalesce at the nanoscale," *Biophysical Journal*, vol. 99, no. 4, pp. 1175 – 1181, 2010.

[57] Roy, A., Kucukural, A., and Zhang, Y., "I-tasser: a unified platform for automated protein structure and function prediction.," *Nature Protocols*, vol. 5, no. 4, pp. 725–738, 2010.

[58] Sapsaman, T., *An Energy Landscaping Approach to the Protein Folding Problem.* PhD thesis, Georgia Institute of Technology, December 2009.

[59] Schueler-Furman, O., Wang, C., Bradley, P., Misura, K., and Baker, D., "Progress in modeling of protein structures and interactions," *Science*, vol. 310, no. 5748, pp. 638–642, 2005.

[60] Sherwood, P., Brooks, B. R., and Sansom, M. S., "Multiscale methods for macromolecular simulations," *Current Opinion in Structural Biology*, vol. 18, no. 5, pp. 630 – 640, 2008.

[61] Shvartsman, B., "Direct method for analysis of flexible cantilever beam subjected to two follower forces," *International Journal of Non-Linear Mechanics*, vol. 44, pp. 249–252, Mar. 2009.

[62] Sugeta, H. and Miyazawa, T., "General method for calculating helical parameters of polymer chains from bond lengths, bond angles, and internal-rotation angles," *Biopolymers*, vol. 5, no. 7, pp. 673–679, 1967.

[63] Timoshenko, S. P. and Gere, J. M., *Theory of Elastic Stability.* McGraw-Hill, 1961.

[64] Tozzini, V., "Coarse-grained models for proteins," *Current Opinion in Structural Biology*, vol. 15, no. 2, pp. 144 – 150, 2005.

[65] Tozzini, V., "Multiscale modeling of proteins," *Accounts of Chemical Research*, vol. 43, no. 2, pp. 220–230, 2010. PMID: 19785400.

[66] Veeramani, A., *A Transformative Tool for Minimally Invasive Procedures: Design, Modeling and Real-Time Control of a Polycrystalline Shape Memory Alloy Actuated Robotic Catheter.* PhD thesis, North Carolina State University, April 2009.

[67] Voelz, V. A., Bowman, G. R., Beauchamp, K., and Pande, V. S., "Molecular simulation of ab initio protein folding for a millisecond folder ntl9(1-39)," *Journal of the American Chemical Society*, vol. 132, no. 5, pp. 1526–1528, 2010. PMID: 20070076.

[68] Wang, C. and Kitipornchai, S., "Shooting-optimization technique for large deflection analysis of structural members," *Engineering Structures*, vol. 14, no. 4, pp. 231–240, 1992.

[69] Wedemeyer, W. J. and Scheraga, H. A., *Protein Folding: Overview of Pathways.* John Wiley & Sons, Ltd, 2001.

[70] Yu, W. and Hodges, D., "Generalized timoshenko theory of the variational asymptotic beam sectional analysis," *Journal of the American Helicopter Society*, vol. 50, no. 1, pp. 46–55, 2005.

[71] Ziegler, H., "Linear elastic stability," *ZAMP*, vol. 4, pp. 89–184, 11953.