A NEW ADAPTIVE MULTISCALE APPROACH FOR THE DYNAMIC MODELING AND SIMULATION OF BIOMOLECULAR SYSTEMS

by

ASHLEY CHASE GUY

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2018

To Erin and others,

you know who you are.

Say I'm weary, say I'm sad

Say that health and wealth have missed me

Say I'm growing old, but add <she> kissed me.

ACKNOWLEDGEMENTS

I thank my advisor Dr. Alan Bowling for the numerous opportunities he has afforded me by taking me under his wing and giving me a place in his research group. His advice and expertise have been invaluable and I will always remember my time at UTA fondly. I also thank my committee members, Dr. Ashfaq Adnan, Dr. Hyejin Moon, Dr. Wen Shen, and Dr. Kamesh Subbarao, for their time and expertise in reviewing my work.

I would also like to thank my research colleagues, both past and current, who have reflected and enhanced my studies. In particular, I thank my predecessor Dr. Mahdi Haghshenas-Jaryani, my counterpart Abhishek Chatterjee, and those that come after me, Vatsal Joshi and Manoochehr Rabiei. I wish you all the best of luck.

I am especially grateful to my family - both genetic and adopted - who have supported and encouraged me throughout the years. To my parents, Rand and Sandra, I thank for their continuing care and generosity. To my brother Boyd and sister Ella, I thank for always looking out for their baby brother. I thank the great minds, Dr. Arif Yurdagul and Kyle Prather, for always setting an example, even when they weren't around. I also thank mi hermanos, Casey Kugler, Jason Weilbecher, Kellen McKinney, and Roman Mercado, for ensuring my mental health. And Aimee!

And finally, my deepest thanks go to my partner Erin for her continued support and patience throughout this entire endeavor. Life, from here, is only up.

April 11, 2018

ABSTRACT

A NEW ADAPTIVE MULTISCALE APPROACH FOR THE DYNAMIC MODELING AND SIMULATION OF BIOMOLECULAR SYSTEMS

ASHLEY CHASE GUY, Ph.D.

The University of Texas at Arlington, 2018

Supervising Professor: Alan Bowling

This dissertation presents three developments in the simulation of sub-micron dynamical systems. First, a new thermostat formulation is shown to strictly enforce the constant temperature constraint necessary in many thermodynamic ensembles. This formulation also allows the system to quickly evolve to a low-potential state. Next, this new thermostat formulation is leveraged to extend a continuum-regime multiscale formulation to the atomistic regime allowing for equivalent time histories to be generated in significantly less computation time. Finally, an adaptive multiscale formulation allows for potential forces with time varying magnitudes to contribute to the overall dynamics similarly resulting in equivalent time histories in significantly less computation time.

Sub-micron dynamical simulations traditionally require extensive computational resources to generate desired time evolutions. Microscale phenomena are often driven by even smaller scale dynamics, requiring multiscale system definitions to combine these effects. At the smallest scale, large active forces lead to large resultant accelerations, requiring small integration time steps to fully capture the motion and dictating

the integration time for the entire model. Multiscale modeling techniques aim to reduce this burdensome computational cost.

A multiscale method has been previously shown to greatly reduce the real time required to generate equivalent data. This method identifies a portion of the active and dissipative forces that cancel and contribute little to the overall motion. By scaling these forces, high-frequency vibrations are diminished without significantly affecting the time histories. Integration step sizes can then be increased, reducing the total number of required integration steps and reducing the real time required. This method has been previously applied to systems at length scales ranging from several hundred nanometers to large coarse grained organic molecules.

This multiscale method is first extended to the atomistic regime using a new thermostat formulation. Equivalent results were generated in significantly less time while maintaining the constant temperature constraint. Unlike previous multiscale formulations, this approach relies on a time-variant term to satisfy the multiscale analysis. This reliance on time-variant terms raised the question of how to reformulate the multiscale method to account for the changing proportions in the equations of motion.

The multiscale method was then extended to include an adaptive scaling consideration for forces whose magnitude changes significantly. This adaptive consideration accounts for instances in which certain small forces are not canceled. This consideration allows for small forces acting over long time periods to still produce significant results on the overall time history. Adaptively scaled results showed equivalent time histories to the unscaled case with computation time reduction consistent with previous works.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF ILLUSTRATIONS x

LIST OF TABLES xiii

Chapter Page

1. INTRODUCTION 1

1.1 Overview 1

1.2 Interests in Micro and Nanoscale Systems 2

1.3 Micro and Nanoscale Simulations 3

1.4 Multiscale Systems at the Micro and Nanoscale 7

2. CONTINUUM REGIME MULTISCALE METHOD 9

2.1 Introduction 9

2.2 Multiscale Derivation 10

2.3 Model Description and Equations of Motion 15

2.4 Forces and Energy 16

2.5 Hardware and Simulation Parameters 19

2.6 Results Analysis 20

2.7 Computation Time Reduction 21

2.8 Conclusions 22

3. MODIFICATION TO THE NÒSE-HOOVER THERMOSTAT 26

3.1 Introduction 26

3.2 Nóse-Hoover Formulation 30

3.3 Model Description and Equations of Motion 32
3.4 Forces and Energy 33
3.5 Hardware and Simulation Parameters 36
3.6 Results Analysis 38
3.7 Conclusions 44
4. ATOMISTIC REGIME MULTISCALE METHOD 49
4.1 Introduction 49
4.2 Disproportionality Assumptions 51
4.3 Multiscale Derivation 53
4.4 Model Description and Equations of Motion 56
4.5 Forces and Energy 57
4.6 Determining Appropriate Q and k_d Values 58
4.7 Hardware and Simulation Parameters 60
4.8 Results Analysis 61
4.9 Computation Time Reduction 68
4.10 Conclusions 69
5. ADAPTIVE MULTISCALE METHOD 71
5.1 Introduction 71
5.2 Adaptive Scaling 72
5.3 Model Description and Equation of Motion 74
5.4 Forces and Energy 76
5.5 Hardware and Simulation Parameters 80
5.6 Results Analysis 83
5.7 Computation Time Reduction 91
5.8 Conclusions 91
5.9 Future Work 92

Appendix

A. RECURSIVE FORWARD DYNAMICS ALGORITHM 94

B. EULER PARAMETERS 110

C. CONSTRAINT EMBEDDING 124

D. SIMULATION PARAMETERS 131

REFERENCES 138

BIOGRAPHICAL STATEMENT 159

LIST OF ILLUSTRATIONS

Figure Page

2.1 Transmission electron micrograph of Ebola virion. © Cynthia Goldsmith, CDC 10

2.2 Ebola Glycoprotein physiological structure 11

2.3 System model showing the bulk protein with active-site side chain groups 16

2.4 Plot of drug particle interacting with EBOV GP1 receptor 20

2.5 Bulk protein and side chain base link joint angles. (a) bulk protein, (b) L57, (c) L63, (d) R64, (e) F88, (f) K95 23

2.6 Position of drug particle in inertial frame. (a) $\hat{N}_1$ direction, (b) $\hat{N}_2$, (c) $\hat{N}_3$ 24

2.7 Energy of unscaled system 24

2.8 Energy of scaled system 25

2.9 Comparison of potential energy from unscaled and scaled systems. Note that the scaled system potential has been multiplied by a_2^{-1} 25

3.1 Mechanical model of the sodium, potassium, and nitrate ions represented by spherical particles 33

3.2 Initial positions. Cations (small) and anions (medium) are shown here interacting with a silicon dioxide nanoparticle (large). Ionic radii are enlarged for viewing 37

3.3 Final positions for system using the original Nóse-Hoover thermostat . 38

3.4 Final positions for system using the modified Nóse-Hoover thermostat 39

3.5 Temperature response of standard thermostat showing fixed and stepped target temperature profiles. Stochastic forces disabled 40
3.6 Temperature response of modified thermostat showing fixed and stepped target temperature profiles. Stochastic forces disabled 41
3.7 Stochastic forces enabled 42
3.8 Stochastic forces enabled 43
3.9 Effect of varying gains on response. (a) O(Q) = 1; (b) O(Q) = 10; (c) O(Q) = 100; (d) O(Q) = 1000. Insert shows the order of the derivative gain 46
3.10 Effect of varying gains on simulation time 47
3.11 System energy and thermostat work. (a) standard thermostat; (b) modified thermostat; (c) comparison of total system energy from (a) and (b) 48
4.1 Initial positions. Smaller bodies are sodium and potassium cations. Larger bodies are nitrate anions. The large central body is the silicon dioxide cluster 61
4.2 Final positions 62
4.3 Temperature response for the three simulations 63
4.4 System energy using the original thermostat 64
4.5 System energy using the unscaled modified thermostat 65
4.6 System energy using the scaled modified thermostat 66
4.7 Phase plots for all three simulated systems 67
4.8 Phase plot for the unscaled modified thermostat 68
5.1 Model of the conjugated nanoparticle. Nanoparticle center of mass at point O. Bodies A, B, C, and D are the four sub-bodies of the coarse grained GP1b protein 75

5.2 Initial positions 81
5.3 Final positions of the adaptively scaled simulation 82
5.4 Nanoparticle position in $\widehat{\mathbf{N}}_1$ direction for the adaptively scaled, constantly scaled, and unscaled cases 84
5.5 Adaptively scaled system energy. W denotes the work done by friction (subscript d), potential (p), Brownian (b), and conformational (k) forces. T denotes kinetic energy 85
5.6 Unscaled system energy. W denotes the work done by friction (subscript d), potential (p), Brownian (b), and conformational (k) forces. T denotes kinetic energy 86
5.7 Force magnitudes from the scaled system. Forces shown are damping (subscript d), conformational (k), potential (p), and Brownian (b). Dashed vertical line denotes time when adaptive scaling is activated . 87
5.8 Adaptive scaling factor a_2^* over time 88
5.9 Comparison of kinetic energy from the adaptively scaled and unscaled simulations. Solid line is the adaptively scaled case and the dashed line is the unscaled case. Dashed vertical line denotes time when adaptive scaling is activated 89
5.10 Comparison of work done by damping (subscript d) and Brownian (b) forces from the adaptively scaled and unscaled simulations 90

LIST OF TABLES

Table Page

4.1 Equations of motion disproportionalities. 52

CHAPTER 1

INTRODUCTION

1.1 Overview

This dissertation presents several advancements in the modeling and simulation of micro and nanoscale systems. An established multiscale method has been shown to significantly reduce the computation time required to generate equivalent time histories in continuum regime microscale dynamical simulations by allowing the smallest bodies to be observable from larger time scales. Chapter 2 presents a derivation of this multiscale method and shows its application to a simulation of the Ebola virus glycoprotein receptor interacting with a nanoparticle. While attempting to extend this multiscale method to the atomistic regime, it was necessary to implement a temperature control mechanism called a thermostat. It was found that modifications to a popular thermostat would more strictly enforce the temperature constraint and quickly move the system to a low potential state. Chapter 3 presents the modification and testing of that thermostat. Chapter 4 presents how that modified thermostat can be leveraged to extend the continuum regime multiscale method to the atomistic regime, similarly reducing the computation time required to generate the atomistic time histories. Chapter 5 then presents an extension to the continuum multiscale formulation to adaptively scale potential forces that have significant changes in magnitude over long time frames. This adaptive approach is necessary to capture the work done by these small forces acting consistently over time. The appendices present some of the technical tools used, such as a recursive forward dynamics formulation, Euler parameters, and online constraint embedding.

1.2 Interests in Micro and Nanoscale Systems

Advancements in micro and nanoscale systems have led to impressive developments in medicine [1, 2, 3, 4], materials science [5, 6], sensing [7, 8], and energy generation [9, 10, 11]. Engineering systems at this scale allows for some properties to be minimized - such as the number of defects per volume - while others can be maximized - such as surface area per volume. When these considerations are applied at such small a scale, the larger scale properties can exhibit significant changes [12]. Micro and nanoscale engineered systems in medicine and therapeutics are now smaller than individual cells and are capable of manipulating biochemistry at the molecular level, leading to significant improvements in patients' health and quality of life. Along with our ability to manufacture these small scale systems has come an increased need for visualization, understanding, and control.

An active and exciting area of research in medicine is the use of nanoparticles for targeted drug delivery. Nanoparticles are synthesized submicron bodies that can be tailored to control properties such as diffusivity, preferential binding, and decomposition. For example, the size and surface charge of the particle can affect diffusivity through arterial walls, the blood-brain barrier, and cellular membranes. Structures like antibodies and biopolymers conjugated, or connected, to the nanoparticle surface allow researchers and physicians to externally direct them to specific regions and cells within the patient, reducing collateral damage to healthy tissue. Hollow nanoparticles can enclose drug molecules that passively diffuse into surrounding tissue [13] while other particles are themselves the mechanism of treatment through toxicity [14] or mechanically destroying pathogens [15]. Current works are even personalizing nanoparticles for specific patients [16].

Manufacturing and testing such bioengineered systems can be costly and time consuming, creating a bottleneck in the discovery process. For even a small proposed

change in nanoparticle or conjugate structure, an entirely new experiment would need to be performed to test the efficacy of the proposed structure. These experiments then can be overly reliant on trial and error when there is no other supporting method of investigation in place. Simulations of nanoparticles can investigate a whole range of structures without the need for costly materials or laboratory space. However, simulations require some prior knowledge in order to know what system should be modeled. Combining the empirical studies with simulations allows for each method to inform the other, increasing the rate at which new discoveries are made.

1.3 Micro and Nanoscale Simulations

Simulations of sub-micron dynamical systems have a well established history dating back to the early days of computation [17, 18, 19, 20]. These simulations rely on developments in physics, chemistry, and thermodynamics, the foundations of which were formed long before Alan Turing cracked the Enigma machine. While the idea of using a machine to compute time histories for individual atoms, much less systems with thousands or millions of distinct bodies, must have seemed far-fetched at the time, pioneers such as Boltzmann, Gibbs, Kelvin, Maxwell, and others laid down the mathematical formulations connecting macroscale properties to atomistic scale quantities. These formulations form the basis of the fields of Statistical Mechanics and Thermodynamics today. Combining their discoveries with the laws of mechanics defined by Newton and Euler allows for micro and nanoscale systems to be simulated with high levels of accuracy and precision.

However, simulations of sub-micron dynamical systems are notorious for requiring significant computational resources to generate even modest time evolutions. In both the continuum and atomistic regimes, the large computational burden can be attributed to the complexities of solving the forward dynamics problem. For large

kinematic chains, the interconnectedness of the bodies leads to joint constraints and reaction forces that must be addressed. For large multibody systems, body-to-body potential interactions must be resolved to solve for potential forces. For even simple systems, mass terms are often several orders of magnitude smaller than the active forces, leading to large resultant accelerations and high frequency vibrations that require small integration time steps to fully capture. Generally, as the size of the modeled bodies decreases, so does the integration step size. These small step sizes can be sub-nano for continuum models and sub-femto for atomistic models.

Depending on the phenomena being studied, these small integration step sizes may not inherently be a problem. For investigations of short-duration pico- and nanosecond effects, a femtosecond integration step may be appropriate. However, fast dynamical systems such as biopolymers [21, 22] are connected to engineered systems like nanoparticles [23]. The potential interactions of these fast systems can significantly affect the overall system time history but require small step sizes to fully resolve. If these engineered systems were simulated, milliseconds worth of data may be needed to observe the overall behavior of the large bodies, requiring a daunting number of integration steps. Methods for reducing the real time required to simulate such systems either reduce the computational cost of solving the forward dynamics or increase the integration step size to reduce the total number of integration steps required.

Methods for reducing the time required to compute the forward dynamics include computational and algorithmic methods. Computational methods include parallel processing and octrees. Parallel processing using graphics processing units (GPUs) and field-programmable gate arrays (FPGAs) is highly effective for dynamical simulations, computational fluid dynamics, and structural analysis. The CUDA [24] and OpenCL [25] libraries allow independent computations to be distributed among

numerous GPU and FPGA cores and calculated concurrently. In addition to resolving independent calculations, such as the numerous body-to-body potential interactions, parallel processing is also highly useful for performing the linear algebra common in kinematic analysis. Divide-and-conquer formulations separate both branching and non-branching kinematic chains into subgroups to resolve them concurrently [26, 27]. These parallel processing approaches have been shown to reduce computation time by orders of magnitude [28, 29]. Another useful computational tool is the octree, a method of data structuring used to partition the simulation volume into subspaces [30]. Long-range potential interactions can then be approximated, reducing the number of interactions that are calculated.

Some algorithmic methods include numerical formulations for large kinematically linked systems that avoid inversion of mass matrices [31, 26, 27, 32, 33, 34, 35, 36, 37, 38]. Included in these is the the Newton Euler Inverse Mass Operator (NEIMO) method for calculating the forward dynamics in joint space rather than cartesian [31], allowing for integration time steps an order larger than expected for systems with explicit hydrogens [39]. Also included is the useful reformulation of the equations of motion to solve them in an iterative, recursive fashion [32, 26, 27]. This recursive approach eliminates reaction forces using joint constraints. Mass matrices and forces are modified through back substitution to reflect the effective inertia and forces experienced at each joint. The symbolic equations of motion are thereby avoided completely. While inverting a dense matrix has a computational cost of $O(N^3)$, these recursive formulations can have a computational cost as low as $O(logN)$ when parallelized, a significant improvement over symbolic approaches for large systems.

Another algorithmic approach to reduce computation time is the use of a potential field calculator for minimizing the number of potential interactions that must be calculated explicitly. A molecular dynamics "truth" model would require that

each atom or charged point interact with every other point in the model, excluding those on a shared rigid body. Popular force field calculators, such as the Particle-Mesh (PM) and Particle-Particle-Particle-Mesh (P^3M) method, replace the interacting bodies with a mesh [40, 41, 42, 43]. The force acting on a particular body may then be found without the need to solve each point-to-point interaction explicitly. While the PM method uses the mesh to approximate all interactions, the P^3M method uses the mesh for long-range interactions and point-to-point for close interactions. Many long-range force approximations neglect the resultant moments, but recent work has provided the necessary derivations [44]. The efficacy of these force field formulations continues to be improved [45, 46]. Another alternative is to simply neglect long-range potential calculations, though this approach is usually accompanied by some compensation to account for the omitted energy [47, 48, 49].

Other algorithmic methods reformulate the equations of motion to eliminate high frequency vibrations. This can be accomplished by eliminating high frequency modes to isolate the higher order dynamics [50] or by isolating low order perturbations [51, 52, 53]. Another method considers the significantly large damping term compared to the mass and omits the mass term altogether [54, 55, 56]. Since these massless models halve the number of integrated variables, computation time is reduced, but underdamped behavior can no longer be predicted. This approximation leads to the reduced order models described by the well-known overdamped Langevin equation.

Another popular approach is to define the model at larger scales and reduce resolution. This usually involves coarse grained models that approximate clusters of bodies such that internal degrees of freedom can be eliminated [57]. By removing these degrees of freedom, high frequency vibrations are eliminated and integration step sizes can be increased. Finer coarse graining methods remove the degrees of freedom associated with bond angle and bond length while retaining rotations about single

bonds [32]. For biopolymers, residue based coarse graining groups the atoms into backbone and residue clusters [58] while shape-based approaches use neural nets for identifying clusters [59]. Some implementations allows for system resolution to change dynamically by identifying and constraining kinematically inactive joints forming larger rigid bodies and a coarser graining [60, 61].

While coarse graining an entire model may be tempting, since many slow phenomena are in fact driven by fast mechanics [21, 22], observation of slower phenomena may require that higher resolution be retained, at least in some critical regions. For the simulation of biopolymers, some dynamically active regions of the polymer might require an atomistic resolution. For the simulation of conjugated nanoparticles [23], coarse grained biopolymers are still several orders of magnitude smaller than the host bodies. Combining all these bodies into a single system model leads to a multiscale problem in which integration step sizes are defined by the smallest bodies included. The term multiscale refers to a system in which multiple phenomena at various scales are combined to show the interconnectedness of these phenomena. It can be seen then that it is difficult to avoid multiscale problems in biological simulations at the micro and nanoscale and it is clear from the research it is an on-going consideration [62, 63, 64]. Multiscale system definitions provide distinct opportunities for addressing the problem of burdensome computation time.

1.4 Multiscale Systems at the Micro and Nanoscale

Methods for accommodating multiscale problems can be classified as serial and concurrent approaches. Serial approaches separate the whole system into several subsystems defined at varying resolution [65]. These systems are arranged in a hierarchical manner where high-resolution system states are used to define inputs for medium-resolution systems, which in turn define inputs for lower-resolution systems.

These formulations are useful for studies in which properties are easily delineated by scale [66]. Concurrent methods combine each of the simulation layers and share information between layers [67]. These formulations have found extensive use in material studies [68] and investigations of crack propagation [69, 70].

Another approach for addressing the multiscale problem is to model all bodies together and reformulate the equations of motion such that the motion of the small bodies may be observable from a larger time scale consistent with the larger bodies. Previous works have developed a multiscale formulation applicable to the continuum regime that has been shown to significantly reduce the required computation time while generating equivalent time histories to the unscaled models [71, 72, 73, 74, 75, 76, 77, 78, 79, 80]. This multiscale method identifies a significant disproportionality in the equations of motion. The method of multiple scales [51] then uses that disproportionality to reveal a scaling of the generalized forces such that they do not produce large accelerations, but do yield an accurate estimate of the system's motion. This formulation has been successfully applied to simulations of nanobeads in optical tweezers [76], estrogen docking [77], motor protein-cytoskeleton interactions [79], and long kinematic chains of amino acids [80]. In the case of [76], a computation time reduction of 99.9% was achieved while more closely matching experimental results than the unscaled case. These works applied this multiscale method to systems of differing scales and complexity within the continuum regime. While later chapters present extensions to this multiscale method, Chapter 2 provides a derivation of this method and shows its application to a simulation of the Ebola virus glycoprotein receptor interacting with a nanoparticle.

CHAPTER 2

CONTINUUM REGIME MULTISCALE METHOD

2.1 Introduction

This chapter presents the established continuum regime multiscale method and shows its application to a simulation of a finely coarse grained biopolymer. Previous works have applied this multiscale method to systems at various length scales ranging from nanobeads with radii of hundreds of nanometers [76], to coarsely grained biopolymers [79], to cholesterol molecules with radii on the order of Angstroms [77]. In [79], coarse graining was applied to a model of Myosin V walking along an actin filament. This coarse graining did not consider individual residues or rotations about single bonds. In this chapter, a model of a finely coarse grained protein is used to demonstrate the multiscale method. This fine coarse graining retains the degrees of freedom about single bonds within dynamically active residue sidechains.

The model chosen is the receptor site of the Ebola virus (EBOV) glycoprotein (GP) interacting with a theoretical drug particle. Figure 2.1 shows a colored transmission electron micrograph of an EBOV virion. The EBOV GP is a virally encoded trimer bound within the virion envelope. Figure 2.2 shows a 3D model of the EBOV GP structure under physiological conditions [81]. The interaction between this structure and host cell receptors has been shown to play a key role in host cell infection [82, 83, 84]. Each protein within the trimer is composed of 676 amino acids and is divided into the GP1 region containing the active site and the GP2 region serving as a transmembrane anchor. The GP1 region is structurally distinct from human receptors, meaning this structure could be selectively targeted by drugs. The GP1 receptor

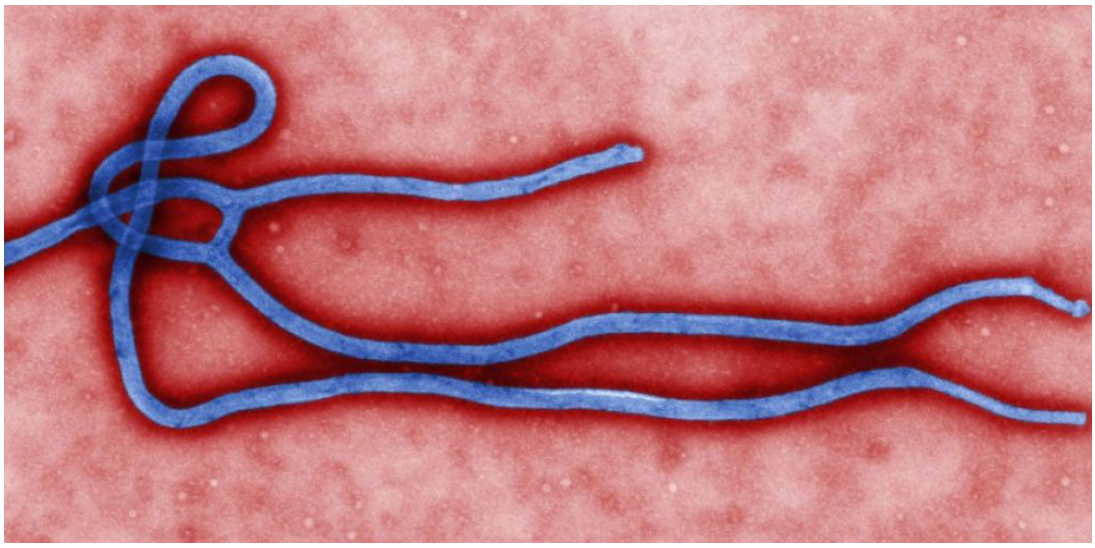

Figure 2.1. Transmission electron micrograph of Ebola virion. © Cynthia Goldsmith, CDC.

site is a protein consisting of 148 amino acids spanning residues 54-201 in the GP primary sequence. Five specific amino acids form the receptor site: leucine-57 (L57), leucine-63 (L63), arginine-64 (R64), phenylalanine-88 (F88), and lysine-95 (K95). A theoretical drug particle is modeled which interacts with the GP1 receptor site.

The multiscale analysis is applied to the this system and shown to significantly reduce the computation time required to generate the time evolution data. While the multiscale analysis had been applied to coarse grained biopolymers before [79], the work detailed in this chapter is the first application of this method to a fine coarse graining allowing rotations around individual chemical bonds.

2.2 Multiscale Derivation

Consider the equation of motion for a rigid multibody model:

$$M(\mathbf{q})\ \ddot{\mathbf{q}}\ +\ C(\mathbf{q},\dot{\mathbf{q}})\ =\ \mathbf{\Gamma}(\mathbf{q},\dot{\mathbf{q}}) \tag{2.1}$$

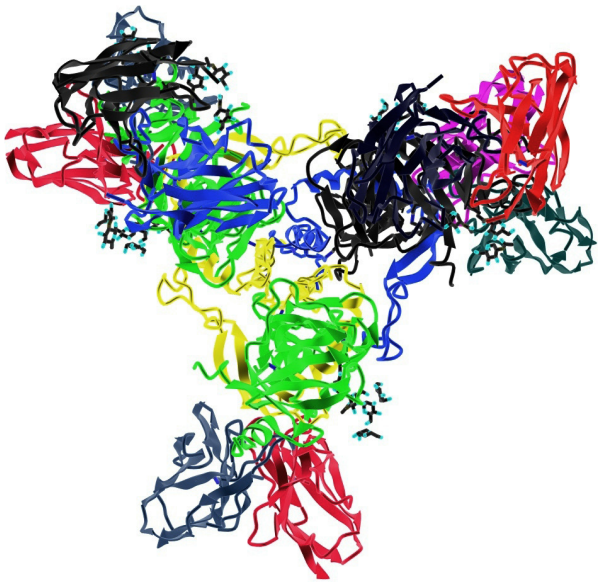

Figure 2.2. Ebola Glycoprotein physiological structure.

where $M(\mathbf{q}) \in \mathbb{R}^{N \times N}$ denotes the mass matrix where N is the number of degrees of freedom, $\mathbf{q} \in \mathbb{R}^{N \times 1}$ is a vector containing the generalized coordinates with time derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, and $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{N \times 1}$ is the nonlinear accelerations. The term $\mathbf{\Gamma}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{N \times 1}$ on the right side of Eqn. (2.1) denotes the generalized active forces and is defined by:

$$\mathbf{\Gamma}(\mathbf{q}, \dot{\mathbf{q}}) \;=\; \mathbf{\Gamma}_{damping}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{\Gamma}_{others}(\mathbf{q}) \tag{2.2}$$

$$\mathbf{\Gamma}_{damping} \;=\; -\beta \; D(\mathbf{q}) \; \dot{\mathbf{q}} \tag{2.3}$$

where the subscript *damping* denotes the friction forces and *others* denotes all other active forces. The term $-\beta D(\mathbf{q})\dot{\mathbf{q}}$ models the friction forces where β is a scalar characteristic damping term and $D(\mathbf{q}) \in \mathbb{R}^{NxN}$ maps the friction to the generalized speeds.

Deriving the multiscale formulation begins by rearranging Eqn. (2.1):

$$m \; \overline{M}(\mathbf{q}) \; \ddot{\mathbf{q}} \;+\; m \; \overline{C}(\mathbf{q}, \dot{\mathbf{q}}) \;+\; \beta \; D(\mathbf{q}) \; \dot{\mathbf{q}} \;=\; \mathbf{\Gamma}_{others}(\mathbf{q}) \tag{2.4}$$

where m is a characteristic mass term that has been extracted from M and C leaving the modified $\overline{M}$ and $\overline{C}$ terms, respectively. Normalizing Eqn. (2.4) by the damping term β yields:

$$\frac{m}{\beta}\,\overline{M}(\mathbf{q})\,\ddot{\mathbf{q}} \;+\; \frac{m}{\beta}\,\overline{C}(\mathbf{q},\dot{\mathbf{q}}) \;+\; D(\mathbf{q})\,\dot{\mathbf{q}} \;=\; \frac{\mathbf{\Gamma}_{others}(\mathbf{q})}{\beta} \tag{2.5}$$

At the sub-micron scale, mass terms are often several orders of magnitude smaller than damping terms. A small parameter may then be defined to represent this disproportionality:

$$\epsilon \;=\; \frac{m}{\beta} \tag{2.6}$$

Using this new parameter ϵ, Eqn. (2.5) may now be written as:

$$\epsilon\,\overline{M}(\mathbf{q})\,\ddot{\mathbf{q}} \;+\; \epsilon\,\overline{C}(\mathbf{q},\dot{\mathbf{q}}) \;+\; D(\mathbf{q})\,\dot{\mathbf{q}} \;=\; \frac{\mathbf{\Gamma}_{others}(\mathbf{q})}{\beta} \tag{2.7}$$

When a small parameter such as ϵ appears on the leading term of a differential equation, the result is a singular perturbation problem where it can be difficult to find exact solutions. An approximate solution to this perturbation problem can be found by replacing the terms with asymptotic expansions. The Method of Multiple Scales [51] uses the small parameter ϵ to similarly expand time into an asymptotic series: $T_i = \epsilon^i\, t$. The time derivatives can then be expanded into series:

$$\dot{\mathbf{q}} \;=\; \sum_{i=0}^{\infty} \epsilon^i\, \frac{\partial \mathbf{q}}{\partial T_i} \tag{2.8}$$

$$\ddot{\mathbf{q}} \;=\; \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \epsilon^i\,\epsilon^j \frac{\partial^2 \mathbf{q}}{\partial T_i \partial T_j} \tag{2.9}$$

Substituting these expansions into Eqn. (2.7) and arranging by increasing order of ϵ yields:

$$\mathbf{0} \;=\; \epsilon^0\left(D(\mathbf{q})\,\frac{\partial \mathbf{q}}{\partial T_0} \;-\; \frac{\mathbf{\Gamma}_{others}(\mathbf{q})}{\beta}\right) \;+\; \epsilon^1\Big(\overline{M}(\mathbf{q})\frac{\partial^2 \mathbf{q}}{\partial T_0^2} \;+\; \overline{C}_{00} \;+\; \cdots\Big) \;+\; \cdots \tag{2.10}$$

where

$$\overline{C}_{00} \;=\; \frac{\partial \overline{M}(\mathbf{q})}{\partial T_0}\frac{\partial \mathbf{q}}{\partial T_0} \;-\; \Big(\frac{\partial \mathbf{q}^T}{\partial T_0}\overline{M}(\mathbf{q})\frac{\partial \mathbf{q}}{\partial T_0}\Big) \tag{2.11}$$

$$\overline{C}_{ij} \;=\; \epsilon^i\epsilon^j\Big(\frac{\partial \overline{M}(\mathbf{q})}{\partial T_i}\frac{\partial \mathbf{q}}{\partial T_j} \;-\; \frac{\partial}{\partial \mathbf{q}}\Big(\frac{\partial \mathbf{q}^T}{\partial T_i}\overline{M}(\mathbf{q})\frac{\partial \mathbf{q}}{\partial T_j}\Big)\Big) \tag{2.12}$$

Note that the first term on the left side of Eqn. (2.10) contains the damping and remaining active forces. The second term contains the generalized accelerations and the first set of higher order terms. Considering the definition of the small parameter ϵ in Eqn. (2.6), for Eqn. (2.10) to equal zero, the first term on the left side must largely cancel. If these forces cancel, they do no work and produce no motion. These forces then can be scaled to eliminate the large portions of these forces that cancel without significantly affecting the time evolution.

Note that the claim above assumes the generalized accelerations are not significantly large. Often in unscaled "truth" models, the first term in Eqn. (2.10) does not equal zero resulting in large accelerations. These accelerations can be oscillatory, requiring small integration step sizes to resolve but ultimately do not contribute to the overall time evolution. This oscillatory behavior becomes even more pronounced for finer coarse grained and atomistic models, but is still present in lower resolution models and is often the result of stiff conformational springs. *The multiscale formulation shown here imposes the assumption of small accelerations by scaling the generalized active forces.*

The scaling of the generalized active forces is accomplished by splitting these forces into small and large parts:

$$(a_1 + a_2)\,\Big(D(\mathbf{q})\,\frac{\partial \mathbf{q}}{\partial T_0} \;-\; \frac{\mathbf{\Gamma}_{others}(\mathbf{q})}{\beta}\Big) \;=\; \Big(D(\mathbf{q})\,\frac{\partial \mathbf{q}}{\partial T_0} \;-\; \frac{\mathbf{\Gamma}_{others}(\mathbf{q})}{\beta}\Big) \tag{2.13}$$

where the scalars $a_1 + a_2 = 1$ and $a_1 \gg a_2$. Substituting this definition into Eqn. (2.4) yields the scaled equation of motion:

$$M(\mathbf{q})\ \ddot{\mathbf{q}}\ +\ C(\mathbf{q},\dot{\mathbf{q}})\ +\ a_2\ \beta\ D(\mathbf{q})\ \dot{\mathbf{q}}\ =\ a_2\ \mathbf{\Gamma}_{others}(\mathbf{q}) \tag{2.14}$$

where it is assumed that the forces associated with the a_1 term cancel such that Eqn. (2.10) is true. This scaled equation of motion suggests that for systems satisfying the initial assumptions, specifically that the characteristic damping term is significantly larger than the characteristic mass, the generalized active forces may be scaled by the small parameter a_2. The value a_2 is user-specified and found by comparing scaled and unscaled response for short-time simulations. A useful first guess is $a_2 \approx \epsilon$. The practical effect of this force scaling is an elimination of the high frequency accelerations that require small integration time steps to resolve. This allows for integration to be performed at larger time steps, usually an order of magnitude increase. By increasing the integration time step, the number of integration steps that must be performed is reduced, thus reducing the real time required to generate the time evolution data.

The multiscale formulation derived above was extended to accommodate a multibody coarse grained biopolymer system with conformational forces arising from torsional springs [79]. That analysis showed that when the spring constant is significantly large, the conformational forces may be included within the scaled portion of Eqn. (2.10). If these spring constants show a large disproportinality with respect to the damping term, two small parameters, ϵ_1 and ϵ_2, can be defined. This analysis yields the scaled equation of motion:

$$M(\mathbf{q})\ \ddot{\mathbf{q}}\ +\ C(\mathbf{q},\dot{\mathbf{q}})\ +\ a_2\ \beta\ D(\mathbf{q})\ \dot{\mathbf{q}}\ +\ a_2 b_2 \mathbf{\Gamma}_k(\mathbf{q})\ =\ a_2 b_2 \mathbf{\Gamma}_{others}(\mathbf{q}) \tag{2.15}$$

where $\mathbf{\Gamma}_k(\mathbf{q}) \in \mathbb{R}^{N\times 1}$ denotes the forces produced by conformational springs and b_2 is a second scaling factor. This formulation was used in damped coarse-grained models

in which the conformational forces showed a large disproportionality with respect to the damping forces. By using two scaling factors, all forces are brought into an order similar to that of the mass.

2.3 Model Description and Equations of Motion

As mentioned before, the system selected for testing the multsicale method on coarse grained biopolymers is a drug particle interacting with a EBOV GP1 receptor. The equation of motion is similar to that given by Eqn. (2.14):

$$M(\mathbf{q})\ \ddot{\mathbf{q}}\ +\ C(\mathbf{q},\dot{\mathbf{q}})\ +\ a_2\ \beta\ D(\mathbf{q})\ \dot{\mathbf{q}}\ =\ a_2\ \mathbf{\Gamma}_p(\mathbf{q}) \tag{2.16}$$

where $\mathbf{\Gamma}_p(\mathbf{q}) \in \mathbb{R}^{N\times 1}$ denotes forces arising from potential interactions.

This model consists of a large body branching into five kinematic trees and an ungrounded spherical drug particle. The large body represents the bulk of the receptor protein and is considered to be in a state of internal equilibrium and behave as a rigid body. This bulk protein is connected by a rotational joint to a fixed base - assumed to the remainder of the GP1 structure. The five kinematic trees model the five amino acid residues identified as the specific receptor site: leucine-57 (L57), leucine-63 (L63), arginine-64 (R64), phenylalanine-88 (F88), and lysine-95 (K95). A fine coarse graining is applied to the residues which removes the degrees of freedom associated with bond length and bond angle but retains the rotations about single bonds. The drug particle is modeled as a sphere with three degrees of translation. Figure 2.3 shows the mechanical model for this system.

The forward dynamics are solved using a recursive numerical approach similar to [26] in which quantities are defined using 6-dimensional spatial notation. This approach eliminates reaction forces using joint constraints. Mass matrices and forces are modified through back substitution to reflect the effective inertia and forces ex-

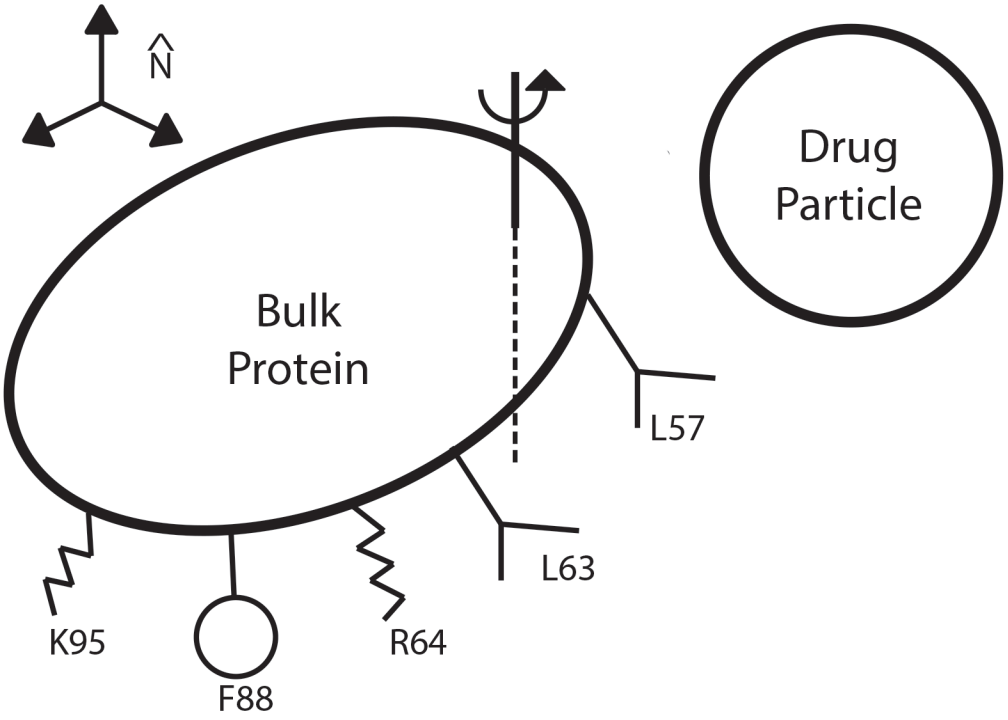

Figure 2.3. System model showing the bulk protein with active-site side chain groups.

perienced at each joint. A derivation of such a recursive formulation can be seen in Appendix A.

2.4 Forces and Energy

The system is driven by the point-to-point potential interactions between charged points. These interactions are calculated between points on the receptor proteins and the points on the drug particle. Also considered are interactions amongst the points on the receptor protein when those points are separated by at least three joints.

The potential energy is modeled using a combination of Coulomb and Lennard-Jones terms:

$$\Phi_{ij}(r) \;=\; \frac{c_i c_j}{4\pi\epsilon_0\epsilon_r r_{ij}} \;+\; 4\epsilon_{ij}\Big[\Big(\frac{\sigma_{ij}}{r_{ij}}\Big)^{12} \;-\; \Big(\frac{\sigma_{ij}}{r_{ij}}\Big)^{6}\Big]$$

$$\epsilon_{ij} \;=\; \sqrt{\epsilon_i \epsilon_j} \tag{2.17}$$

$$\sigma_{ij} \;=\; \frac{\sigma_i + \sigma_j}{2}$$

where Φ denotes a potential energy quantity, r is the distance between the i^{th} and j^{th} points, c is the point charge, ϵ_0 is the permittivity of free space, ϵ_r is the relative permittivity of the medium, and ϵ and σ are Lennard-Jones constants. The subscripts i and j are iterated over all the points in the model. Equation (2.17) can also be written as:

$$\Phi_{ij}(r) \;=\; \frac{A_{ij}}{r_{ij}^{12}} \;-\; \frac{B_{ij}}{r_{ij}^{6}} \;+\; \frac{C_{ij}}{r_{ij}} \tag{2.18}$$

where:

$$A_{ij} \;=\; 4\epsilon_{ij}\sigma_{ij}^{12}$$

$$B_{ij} \;=\; 4\epsilon_{ij}\sigma_{ij}^{6} \tag{2.19}$$

$$C_{ij} \;=\; \frac{c_i c_j}{4\pi\epsilon_0\epsilon_r}$$

The force arising from these potential interactions are then found by taking the derivative of the energy expression:

$$\Gamma_p(r) \;=\; \frac{d\Phi}{dr}\;\bar{r} \tag{2.20}$$

where $\bar{r}$ denotes the unit vector along which the resultant force acts. For higher resolution biopolymer models, a four-term [85] or six-term potential expression [86] is more appropriate. Since the coarse grain model used here removes the degrees of freedom associated with bond angle and bond length and since the dihedral bonds associated

with backbone alpha carbons are contained within the bulk protein, both of those expressions simplify to the expression given by Eqn. (2.17). These more complex potential expressions arise from the matching of simulation data with experimental vibrational analysis [86].

The theoretical drug particle and bulk protein body are large enough to satisfy the continuum assumption, so viscous friction is appropriate. This is determined by calculating the Knudsen number for the bodies:

$$Kn \;=\; \frac{\lambda}{L} \tag{2.21}$$

where λ denotes the mean free path of the medium and L is the characteristic length of the body. Damping terms were calculated using Stoke's Law:

$$\beta \;=\; 6\pi\eta R v \tag{2.22}$$

where η denotes the dynamic viscosity of the medium, R is the radius of the body, and v is the flow velocity relative to the body. This friction is linearly distributed among the protein components as a function of relative mass. For atomistic models, especially when the degrees of freedom associated with bond length and bond angle are retained, a more appropriate friction model for the small molecular components is a temperature controlling thermostat that redistributes energy to enforce a constant temperature constraint. Temperature control and thermostats are further discussed in Chapter 3.

Energy conservation and dynamic consistency are confirmed by summing the instantaneous kinetic and potential energies and subtracting the thermostat work:

$$T \;+\; \mathbf{\Phi} \;-\; W \;=\; constant \tag{2.23}$$

where T denotes the kinetic energy, $\mathbf{\Phi}$ is the total potential energy, and work done by friction is calculated from:

$$W = \int_{t1}^{t2} \sum_{i=1}^{N} \Gamma_{damping,i} \cdot \dot{q}_i \, dt \tag{2.24}$$

The *constant* on the right hand side of Eqn. (2.23) shows that for all time, the total energy of the system must be a constant.

2.5 Hardware and Simulation Parameters

Simulation files were coded using C++ programming language using the Eigen library [87]. Numerical integration was performed on a DELL PowerEdge 2900 III Server with two quad-core, 2.0 GHz processors running Linux 12.04.5 LTS operating system. The integration was performed by a Kutta-Merson algorithm [88] with adaptive integration step sizes for reducing the numerical error. Relative and absolute errors were 10^{-7} and 10^{-8} respectively. Data comes from systems printing data every 100 fs. Final simulation time was 6.4 ns. A unit system of (zg, nm, ns) was selected to keep the order of terms in the side chain equations of motion near unity. The value of a_2 selected was 10^{-3}. Table D.1 in Appendix D lists the values of the parameters used.

GP1 structure and initial conditions were estimated from x-ray crystallography data [89, 90]. Since the primary sequence is known, the actual mass and estimated inertia can be determined from the known primary sequence and estimated structure. For calculating the drag coefficient, the bulk protein was approximated as a sphere with radius of 225 nm. The drug particle has a radius of 1.5 nm and a mass of 0.4 zg. Figure 2.4 shows a plot of the drug particle interacting with the EBOV GP1 receptor.

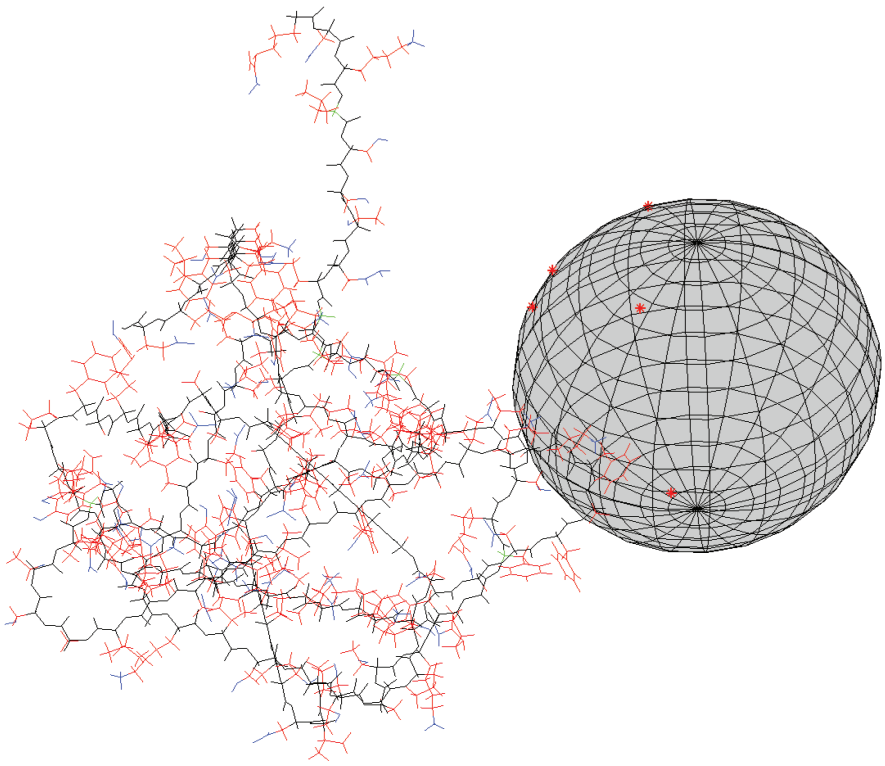

Figure 2.4. Plot of drug particle interacting with EBOV GP1 receptor.

2.6 Results Analysis

Figure 2.5 shows the time evolution of six of the generalized coordinates. These coordinates correspond to the joint angles of the bulk protein and the base link of each of the five side chains. The darker line denotes the unscaled system and the lighter line denotes the scaled system. It can be seen that while some variance is present, the scaled and unscaled results are largely consistent. The standard deviations in absolute differences for plots (a)-(f) are 0.0013, 0.0108, 0.0128, 0.0082, 0.0833, and 0.3923 radians, respectively.

Figure 2.6 shows the position of the drug particle center of mass defined in the inertial frame. Similarly to the data in Fig. 2.5, some variance is present but the

results are largely consistent. Standard deviations in absolute differences for plots (a)-(c) are 0.6629, 0.2726, and 1.6545 Å, respectively.

Figures 2.7 and 2.8 show the system energies of the unscaled and scaled systems, respectively. The dark flat line denotes the total system energy, the lighter line with large negative values is the potential energy, the lighter line with small positive values is the kinetic energy, and the dashed line is the work done by friction. It should be noted that there is a change in the vertical scale between the two figures. As potential energy is the antiderivative of the potential force and the potential force is scaled by a_2, it is therefore necessary to scale the potential energy as well. Because of this change in scale, the kinetic energy profile is much more visible in Fig. 2.8 due to the smaller resolution.

Figure 2.9 shows the potential energy of both the unscaled and scaled systems plotted together for comparison. The darker line denotes the unscaled system and the lighter line denotes the scaled system. It can be seen both the scaled and unscaled systems evolve into similar low-potential states. The change in potential energy is converted to kinetic energy and then dissipated by the friction. It can be seen that tracking the work done by friction confirms the dynamic consistency of the model as given by Eqn. (2.23). This time evolution is consistent with theories of mechanics stating that systems evolve into low-potential states coinciding with equilibrium positions. Note that the scaled system potential energy has been multiplied by a_2^{-1} to bring both data sets into the same scale. The standard deviation in absolute difference is $1.547 * 10^{-25}$ J.

2.7 Computation Time Reduction

The unscaled system took 11481 minutes to integrate while the scaled system took 1636 minutes. This reduction in computation time was approximately 86%. It

was observed that the unscaled system was integrated at sub-femtosecond time steps. It was also observed that the scaled system could be integrated at 100 fs without consistent reduction of the time step. For much of the time evolution, the scaled system could be integrated at even larger time steps ($\geq$1 ps) providing even greater computation time reduction. However, these savings must be balanced against the loss of resolution. This increase in integration time step reduced the number of total integration steps calculated, significantly reducing the overall computation time required to generate the data. These results were consistent with previous studies on continuum systems [76, 77, 79].

2.8 Conclusions

This chapter presents an investigation of the continuum regime multiscale method applied to a finely coarse grained biopolymer system. The multiscale analysis shows that a portion of the dissipative and active forces cancel, meaning that these canceled forces can be removed from the dynamics while still accurately predicting overall behavior. The selected model was chosen because previous works had not yet investigated the application of this multiscale analysis to such finely grained biopolymer systems.

Results show that application of the multiscale method reduced the computation time required to generate the time evolution by 86%. While some variance exists between the results of the scaled and unscaled system, this variance can be mitigated through selection of the a_2 scaling factor. It was also seen that, compared to the unscaled system, the scaled system could be integrated at a time step an order of magnitude larger. These results show that the presented multiscale method can significantly reduce the computation time required for fine grained biopolymer systems.

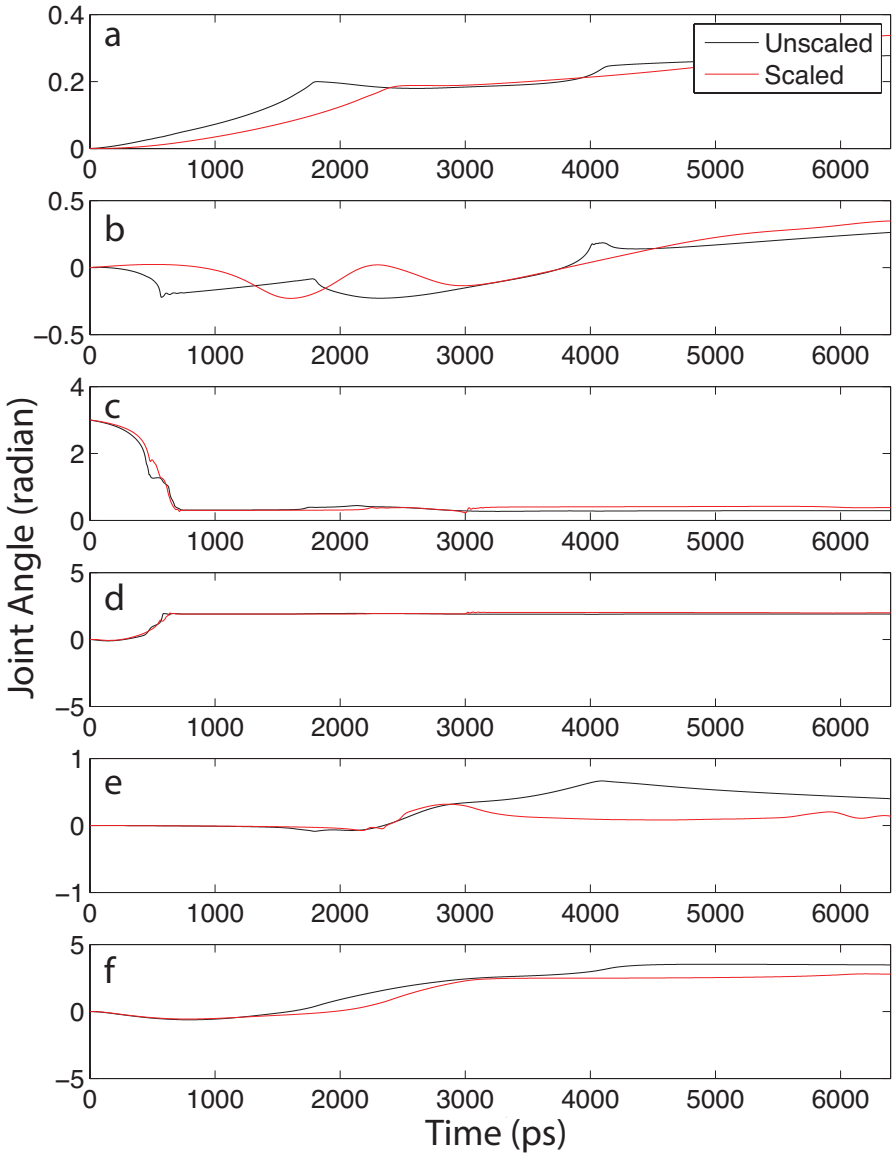

Figure 2.5. Bulk protein and side chain base link joint angles. (a) bulk protein, (b) L57, (c) L63, (d) R64, (e) F88, (f) K95.

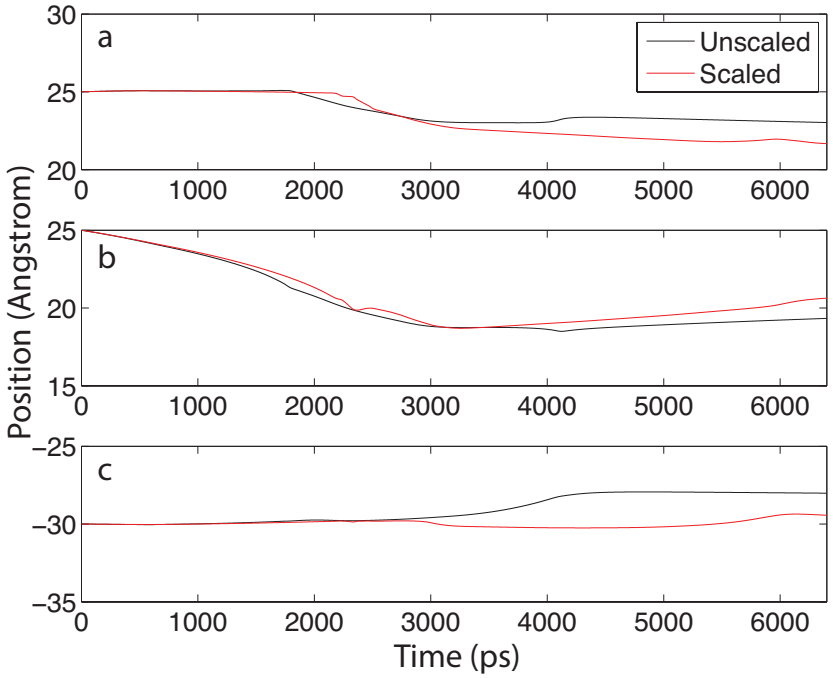

Figure 2.6. Position of drug particle in inertial frame. (a) $\hat{N}_1$ direction, (b) $\hat{N}_2$, (c) $\hat{N}_3$.

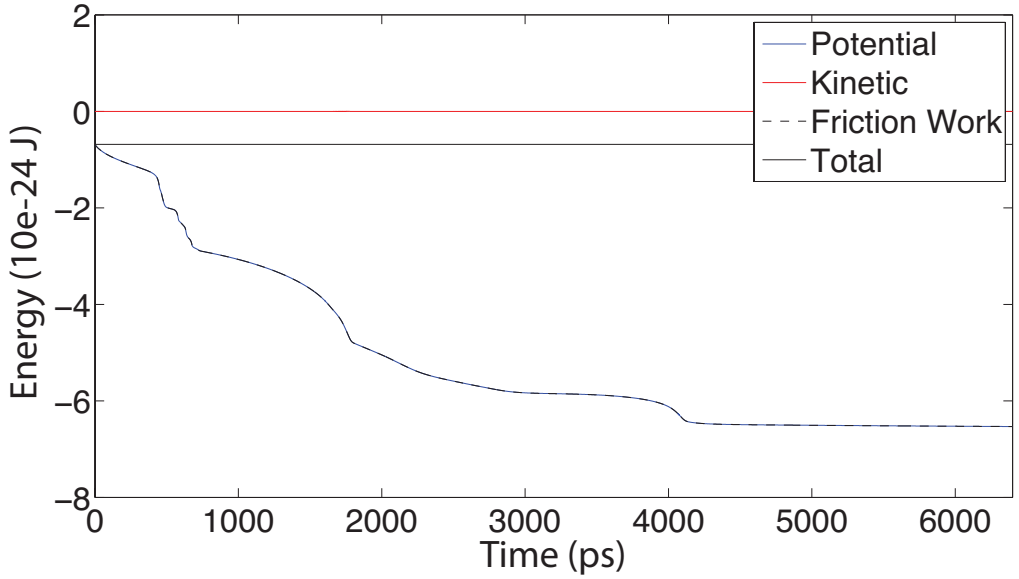

Figure 2.7. Energy of unscaled system.

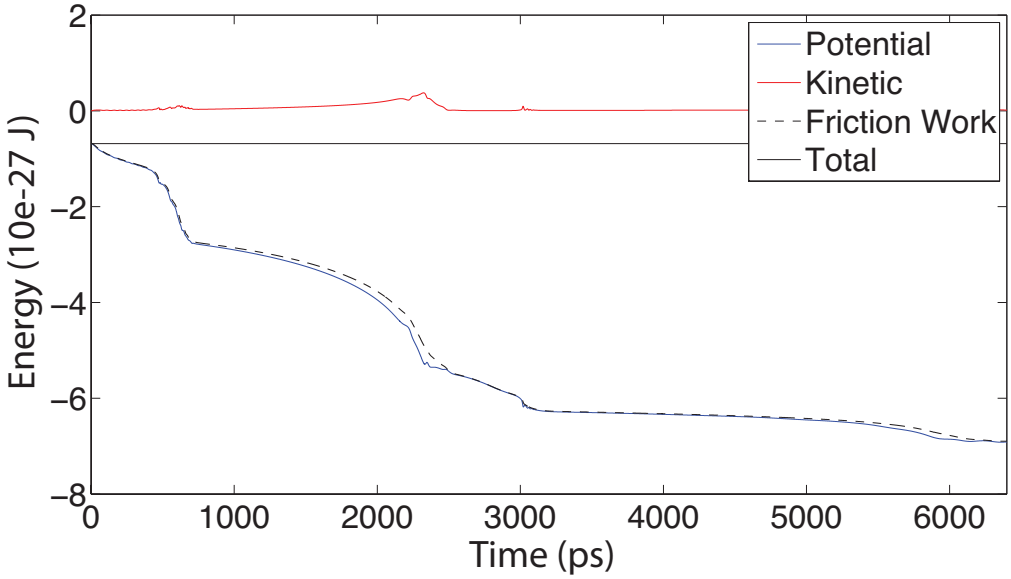

Figure 2.8. Energy of scaled system.

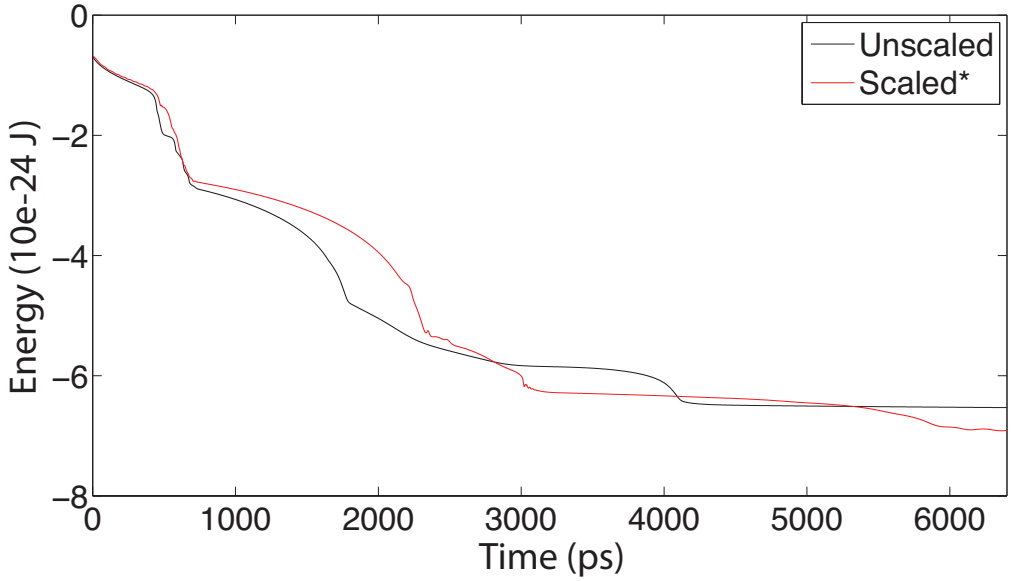

Figure 2.9. Comparison of potential energy from unscaled and scaled systems. Note that the scaled system potential has been multiplied by a_2^{-1}.

CHAPTER 3

MODIFICATION TO THE NÒSE-HOOVER THERMOSTAT

3.1 Introduction

This chapter discusses temperature control in atomistic simulations and presents a modification to an established temperature control approach. Various constraints were assumed in the initial derivations connecting macroscale thermodynamic properties to discrete atomistic quantities [91]. Common sets of constraints are defined by ensembles. These constraints include constant temperature, constant pressure, constant volume, and a constant number of bodies, among others. While a constraint on the volume or number of bodies requires only the consideration of the researcher, constraints on temperature and pressure requires mechanisms for calculating and controlling these quantities.

The constant temperature constraint appears in the NVT (constant-volume, constant-temperature), NPT (constant-pressure, constant temperature), and NST (constant-stress, constant-temperature) ensembles. Methods for controlling system temperature are collectively called thermostats. Of the numerous formulations available [92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102], thermostats can be broadly classified based upon how they measure and control temperature: by kinetic energy using generalized speeds or by system configuration using generalized positions. Some of these formulations reassign or scale the velocities between integration steps [92, 95], but those methods are not deterministic, time-reversible, or dynamically consistent. As an alternative to kinetic energy-based thermostats, configurational thermostats periodically modify the system's configuration to model random temperature effects

[98]. However, these formulations are also not deterministic, time-reversible, or dynamically consistent. A characteristic of these formulations is a mean temperature converging to a steady state value consistent with a target. However, temperature fluctuations periodically appear in the response, even after this steady state has been reached.

A comparison of four thermostats found the Nóse-Hoover thermostat, initially formulated by Nóse [96] and later extended by Hoover [97], produced the most favorable time averaged temperature and phase space distributions [103]. The Nóse-Hoover thermostat is a popular formulation and has been widely applied to systems at various time and length scales [104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120]. The mechanism used is a feedback-based force modifying the generalized momentum of the system. As the mechanism is force based, there are no discontinuities in generalized positions or speeds and the system is dynamically consistent. Similar to the other formulations, Nóse-Hoover response is characterized by a mean temperature converging to a target temperature with oscillations in the actual system temperature consistently appearing with time. Furthermore, this thermostat has been shown to produce ergodic results, meaning that given sufficient simulation time, all points within the phase space will be periodically occupied. Ergodic systems can therefore be modeled probabilistically. The original formulation is a type of single variable thermostat, which have been shown to lead to non-ergodic results for small systems [121, 122, 123]. As such, it is often used in chains to ensure a more ergodic result [124].

Applications and modifications of this thermostat are far ranging. Hoover's continued work with collaborators commonly used harmonic oscillators to characterize the thermostat [104, 109, 112, 114] and showed that deterministic thermostats can be applied to non-equilibrium problems [125]. A combination of the Nóse-Hoover

thermostat and the NEIMO formulation showed this approach gave rise to a canonical distribution and stable dynamics for a simulation of two amorphous polymers [108]. An application of the Nóse-Hoover thermostat with a coarse grained simulation of RNA was used to convert the generalized thermostat forces to spatial forces acting at kinematic joints [118]. Works studying carbon nanotubes have used the thermostat in the testing mechanical response, heat conduction, and flow of fluids through the tubes [113, 115, 116, 117]. Other applications include biopolymers [106, 107, 111], heat conduction [119, 120], Lie algebras [105], and quantum systems [110].

Nóse-Hoover response is characterized by a time-averaged system temperature converging to a target but with short term oscillations consistently appearing that do not diminish with time. There are user specified parameters associated with the thermostat, and while modifying them does affect response it will not prevent these oscillations. As nanoscale simulations are often integrated at the femtosecond scale, short-term temperature oscillations may be interpreted as near-instantaneous fluctuations in local temperature. In some fields this oscillatory response is acceptable. However, these temperature oscillations are coupled with an oscillatory control history that acts as a forcing function on system response. From a multibody dynamics perspective, the oscillatory control history acts more as a driving force on the system than a constraint and may become more significant when the simulation is integrated at large time steps. A constraint should not cause motion of the system, it should only define a subspace of possible motions while other forces move the system around in this subspace. This system can then evolve in a manner more consistent with the theory that systems naturally evolve toward a low-energy equilibrium state, eventually minimizing the system energy. From this perspective, a control law maintaining an on-target instantaneous temperature while minimizing fluctuations would better enforce the temperature constraint. Thermal disturbances may be implemented through

some stochastic Brownian motion-like force or time-variant non-equilibrium target temperatures.

Efforts to reduce oscillations in temperature response have been pursued. In [99], two configurational thermostats were formulated similarly to [97] and found reductions in temperature response oscillation compared to [96] and [98]. As these are configurational thermostats, they share the same problems of non-time-reversibility and state modifying mentioned before.

The modification of the Nóse-Hoover thermostat presented in this chapter is intended to reduce the oscillations in temperature response. The system selected to test this proposed formulation is a molten solution of sodium and potassium nitrate salts. Included in this molten solution is a large cluster, or nanoparticle, of silicon dioxide molecules. This silicon dioxide nanoparticle is assumed to have a negative surface charge that will introduce a potential gradient to the salt solution. These nanoparticle-salt solutions are commonly used as heat transfer fluids in energy storage systems as the inclusion of nanoparticles has been shown to significantly increase the heat capacity of the fluid [126, 127]. It is theorized this increase in heat capacity arises from the aggregation and organization of ions around the nanoparticles as observed under electron microscopy.

Since the salt solution is molten, the weak ionic bonds have broken, allowing the salt molecules to dissociate into negatively-charged anions and positively-charged cations. These ions are all modeled as particles, including the trigonal planar nitrate anions. Each of these particles can translate in three dimensions, but no rotations are considered. It was assumed that the approximation of the nitrate anions as particles would not affect the testing of the proposed formulation efficacy.

A comparison of temperature response shows that the response oscillations are significantly reduced in magnitude, resulting in a more strictly enforced constant

temperature constraint. This tight control of temperature occurred for systems both with and without stochastic forces. An interesting result of the modified thermostat is the swift evolution of the system to a low potential state. It is expected that this result is due to the lack of an oscillatory thermostat force consistently disrupting the low potential positions.

3.2 Nóse-Hoover Formulation

Consider the equation of motion for a simple particle model:

$$M(\mathbf{q})\ddot{\mathbf{q}} \; + \; \mathbf{\Gamma}_{NH}(\mathbf{q}, \dot{\mathbf{q}}) \; = \; \mathbf{\Gamma}(\mathbf{q}) \tag{3.1}$$

where $M(\mathbf{q}) \in \mathbb{R}^{NxN}$ denotes the mass matrix where N is the number of degrees of freedom, $\mathbf{q} \in \mathbb{R}^{Nx1}$ is a vector containing the generalized coordinates with time derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, $\mathbf{\Gamma}(\mathbf{q}) \in \mathbb{R}^{Nx1}$ is a vector of forces, and the subscript NH associates the quantity with the Nóse-Hoover thermostat.

This thermostat connects the modeled system to a thermal reservoir capable of donating or receiving energy from the system as needed to reduce the error in temperature. The Nóse-Hoover thermostat is implemented by extending the dynamics to include an additional generalized coordinate (ζ) and speed ($\dot{\zeta}$). The new speed variable is defined as the scaled error in instantaneous kinetic energy:

$$\dot{\zeta} \; = \; \Big(\dot{\mathbf{q}}^T M \dot{\mathbf{q}} \; - \; N k_B T_d\Big) \frac{1}{Q} \tag{3.2}$$

where k_B denotes the Boltzmann constant, T_d is the absolute target system temperature, and Q is a mass-like term associated with the thermal reservoir. For reference, ζ and $\dot{\zeta}$ have the units s^{-1} and s^{-2}, respectively. It is of note that $\dot{\zeta}$ may be either

positive or negative, depending on the error. It is also of note that Eqn. (3.2) comes from the connection between kinetic energy and temperature:

$$T \;=\; \frac{1}{2} N k_B T_d \tag{3.3}$$

where T denotes kinetic energy. Note that the $\frac{1}{2}$ in Eqn. (3.3) has been combined with Q in Eqn. (3.2). As the target temperature is user specified, it may be defined as time variant for non-equilibrium studies. The error $\dot{\zeta}$ is then integrated and used in feedback to modify the generalized momenta:

$$\mathbf{\Gamma}_{NH} \;=\; -\zeta M \dot{\mathbf{q}} \tag{3.4}$$

The thermostat control law in Eqn. (3.4) is a force similar to friction and resembles a classical controller using integrated state feedback.

While this simple implementation can produce ergodic results for many systems, it fails to produce ergodic results for small or stiff systems. This exception comes from the assumption made during formulation that the trajectory average can be taken into the phase space average [124]. For these systems, a chain of Nóse-Hoover thermostats can implemented:

$$M\ddot{\mathbf{q}} \;-\; \zeta_1 M\dot{\mathbf{q}} = \mathbf{\Gamma}$$

$$\begin{aligned}
\dot{\zeta}_1 \;&=\; \Big(\dot{\mathbf{q}}^T M\dot{\mathbf{q}} \;-\; N k_B T_d\Big)\frac{1}{Q_1} \;-\; \zeta_1\zeta_2 \\
\dot{\zeta}_j \;&=\; \Big(Q_{j-1}\zeta_{j-1}^2 \;-\; k_B T_d\Big)\frac{1}{Q_j} \;-\; \zeta_j\zeta_{j+1} \\
\dot{\zeta}_k \;&=\; \Big(Q_{k-1}\zeta_{k-1}^2 \;-\; k_B T_d\Big)\frac{1}{Q_k}
\end{aligned} \tag{3.5}$$

where j denotes an iteration over a set of k thermostats. This set of thermostat chains can produce ergodic results for small or stiff systems, but still has the aforementioned problem of periodic, non-damping oscillations in response.

To reduce oscillations in temperature response arising from the standard thermostat, the control law seen in Eq. (3.4) is modified to include the $\dot{\zeta}$ term:

$$\mathbf{\Gamma}_{MNH} \;=\; -\big(\zeta \;+\; k_d\dot{\zeta}\big)M\dot{\mathbf{q}} \tag{3.6}$$

where k_d is a gain associated with the derivative term. This control law is constructed similar to classical controllers with higher order states used to diminish oscillations. The independent gain k_d is included to allow tuning of the thermostat response.

3.3 Model Description and Equations of Motion

As mentioned before, the system selected for testing the proposed modified Nóse-Hoover thermostat is a molten blend of sodium and potassium nitrate salts. Figure 3.1 shows the mechanical model for this system. The equation of motion is similar to that given by Eqn. (3.1):

$$\begin{gathered} M(\mathbf{q})\ddot{\mathbf{q}} \;+\; \mathbf{\Gamma}_{MNH}(\mathbf{q},\dot{\mathbf{q}}) \;=\; \mathbf{\Gamma}(\mathbf{q}) \\ \mathbf{\Gamma}(\mathbf{q}) \;=\; \mathbf{\Gamma}_p(\mathbf{q}) \;+\; \mathbf{\Gamma}_v(\mathbf{q}) \;+\; \mathbf{\Gamma}_s \end{gathered} \tag{3.7}$$

where the subscripts MNH denotes the modified Nóse-Hoover thermostat, p denotes potential forces, v denotes volume constraint forces, and s denotes time-variant stochastic forces.

Solving the equations of motion for a set of linear differential equations such as Eqn. (3.7) often involves summing all the known and calculated quantities and inverting the mass matrix. As this system is a particle model, the mass matrix is diagonal and thus quickly inverted. Since the mass matrix is constant, this inversion may be performed in preprocessing. However, mass matrix inversion is avoidable as the vector of known quantities may be iterated over and simply divided by the respective mass term to solve for the generalized acceleration. While parallelization

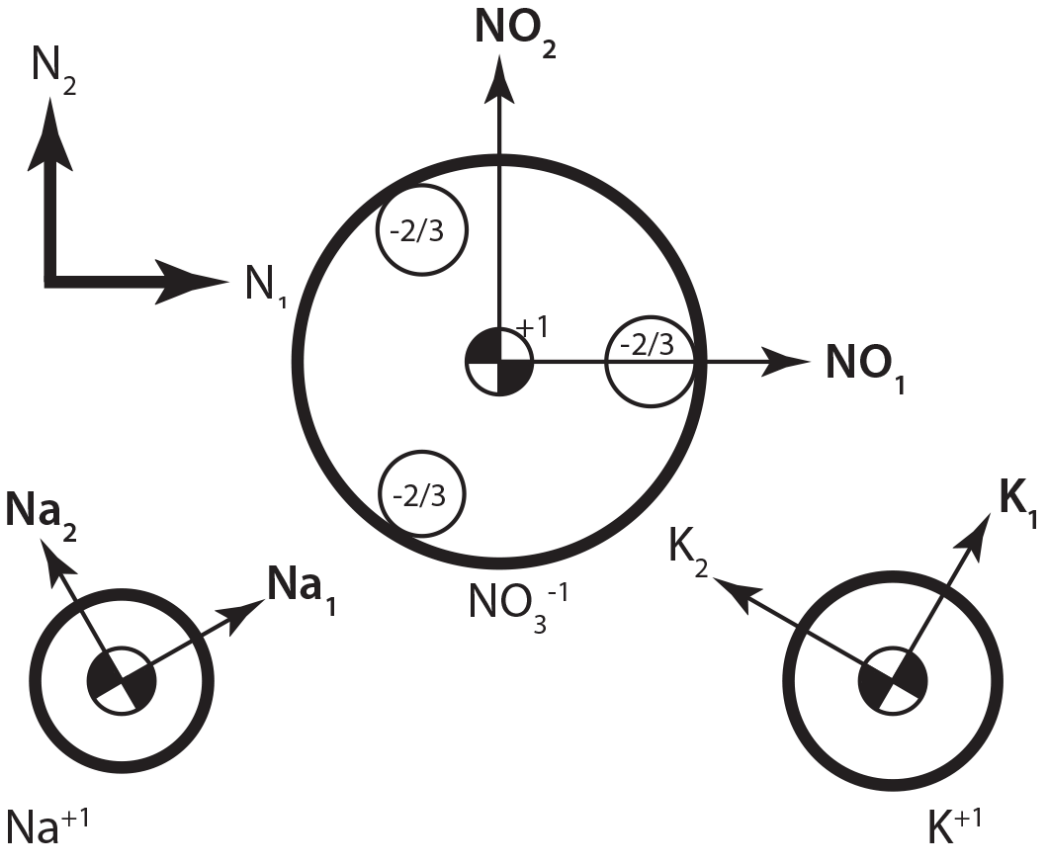

Figure 3.1. Mechanical model of the sodium, potassium, and nitrate ions represented by spherical particles.

was not used to generate the results shown in this chapter, these problems are highly parallelizable as the equation of motion for each degree of freedom is independent from all other equations, meaning the entire set of equations can be solved concurrently.

3.4 Forces and Energy

Active forces in atomistic simulations are dominated by body-to-body potential interactions. These potential interactions are modeled as a combination of Coulomb and Lennard-Jones terms similar to method described in Chapter 2:

$$\Phi_{p,ij}(r_{ij}) \;=\; \frac{A_{ij}}{r_{ij}^{12}} \;-\; \frac{B_{ij}}{r_{ij}^{6}} \;+\; \frac{C_{ij}}{r_{ij}} \tag{3.8}$$

where the force arising from a potential interaction is found by taking the derivative:

$$\Gamma_p(r) \;=\; \frac{d\Phi_p}{dr}\bar{r} \tag{3.9}$$

The Lennard-Jones forces provide a small attractive force at large distances and a large repulsive term to model collisions. These potential interactions are calculated body-to-body with a specified threshold distance. Any interaction with a distance beyond this threshold is omitted to facilitate faster computations. The threshold is user-specified such that errors in potential energy calculation are minimized. The volume constraint is enforced by an uncharged potential interaction between each particle and the volume wall. This potential contains only a Lennard-Jones term in which the constants are defined by the body:

$$\begin{aligned} \Phi_{v,i}(r_i) \;&=\; \frac{A_i}{r_i^{12}} \;-\; \frac{B_i}{r_i^{6}} \\ \Gamma_v(r) \;&=\; \frac{d\Phi_v}{dr} \end{aligned} \tag{3.10}$$

Potential forces arising from gravity were not considered because, at the atomistic scale, these force are approximately ten orders of magnitude smaller than the Coulomb and Lennard-Jones terms. While these forces can have significant effects over long time periods, they will not have a significant effect for shorter simulations.

Stochastic forces are modeled to represent thermal noise and interactions between modeled and non-modeled bodies. The method used to implement these stochastic forces is similar to an implementation of Brownian motion previously used for motor proteins and optical tweezers [128, 129]. As Brownian motion is a continuum phenomena, these stochastic forces are not explicitly considered as such. The random force acting on a particular body may be defined, for example, as:

$$\Gamma_s \;=\; B_1(t)\;\widehat{\mathbf{N}}_1 \;+\; B_2(t)\;\widehat{\mathbf{N}}_2 \;+\; B_3(t)\;\widehat{\mathbf{N}}_3 \tag{3.11}$$

where the $B_i(t)$ terms represent forces produced by randomly fluctuating thermal noise and non-modeled collisions. Each component of the random force is treated independently as a normally distributed random variable. They have the following expectations, $E[\cdot]$, or weighted average values:

$$E\left[\ B_i(t)\ \right]\ =\ \left\langle\ B_i(t)\ \right\rangle\ =\ 0 \tag{3.12}$$

and are governed by a fluctuation-dissipation relation:

$$E\left[\ B_i(t_1)\ B_j(t_2)\ \right] = 2\ \beta\ k_B\ T\ \delta(t_1 - t_2)\delta_{i,j} \tag{3.13}$$

where β is a drag coefficient, k_B the Boltzmann constant, and T the instantaneous absolute system temperature. Note that in an implementation of Brownian motion, the β term is often taken from Stoke's law. For the work detailed in this chapter, the value of β was selected such that the magnitude of the Γ_s term approximated a collision. The collection of these random forces comprise $\mathbf{\Gamma}_s$. The randomly generated variables are updated at each integration step and held constant for that single step.

As a means of confirming energy conservation, many works utilizing thermostats present quasi-Hamiltonians including potential and kinetic terms associated with the thermal reservoir [99, 121, 123, 119, 120, 130, 131, 132, 133]. In this study, the work done by the thermostat on the system is calculated by defining the derivative of work at each time step and integrating:

$$W_{NH}\ =\ \int_{t1}^{t2}\sum_{i=1}^{N}\Gamma_{NHi}\cdot\dot{q}_i\ dt\ =\ \int_{t1}^{t2}-\zeta\dot{\mathbf{q}}^T M\dot{\mathbf{q}}\ dt \tag{3.14}$$

$$W_{MNH}\ =\ \int_{t1}^{t2}\sum_{i=1}^{N}\Gamma_{MNHi}\cdot\dot{q}_i\ dt\ =\ \int_{t1}^{t2}-\left(\zeta\ +\ k_d\dot{\zeta}\right)\dot{\mathbf{q}}^T M\dot{\mathbf{q}}dt \tag{3.15}$$

where W is the work done. Energy conservation is then confirmed by summing the instantaneous kinetic and potential energies and subtracting the work:

$$T\ +\ \mathbf{\Phi}\ -\ W\ =\ constant \tag{3.16}$$

where $\mathbf{\Phi}$ is the total potential energy including both the interaction and volume constraint potentials. For a particle model, the kinetic energy can be easily calculated from:

$$T = \frac{1}{2}\dot{\mathbf{q}}^T M \dot{\mathbf{q}} \tag{3.17}$$

The *constant* on the right hand side of Eqn. (3.16) shows that for all time, the total energy of the system must be a constant. Note that the work done by the thermostat is also the change in thermal reservoir energy.

3.5 Hardware and Simulation Parameters

Simulation files were coded in the C++ programming language using the Eigen library [87]. Numerical integration was performed on a DELL PowerEdge 2900 III Server with two quad-core, 2.0 GHz processors running Ubuntu 12.04 LTS operating system. The integration was performed by a Kutta-Merson algorithm [88] with adaptive integration step sizes for reducing the numerical error. Relative and absolute errors were 10^{-4} and 10^{-5}, respectively. Data comes from systems with 100 ions printing data every 100 fs. A unit system of (zg, nm, ns) was selected to keep the order of terms in the equations of motion near unity. Table D.2 in Appendix D lists the values of the parameters used.

Figures 3.2 and 3.4 show the initial and final positions of the modified thermostat system, respectively. For comparison, 3.3 shows the final positions of the original thermostat system. The size of the ions are enlarged in these figures for easier viewing. It can be seen by comparing Figs. 3.4 and 3.3 that the modified thermostat formulation leads to aggregation of cations on the negatively charged nanoparticle surface. As will be discussed later, this aggregation of cations is consistent with a reduction in the potential energy of the modified thermostat system energy.

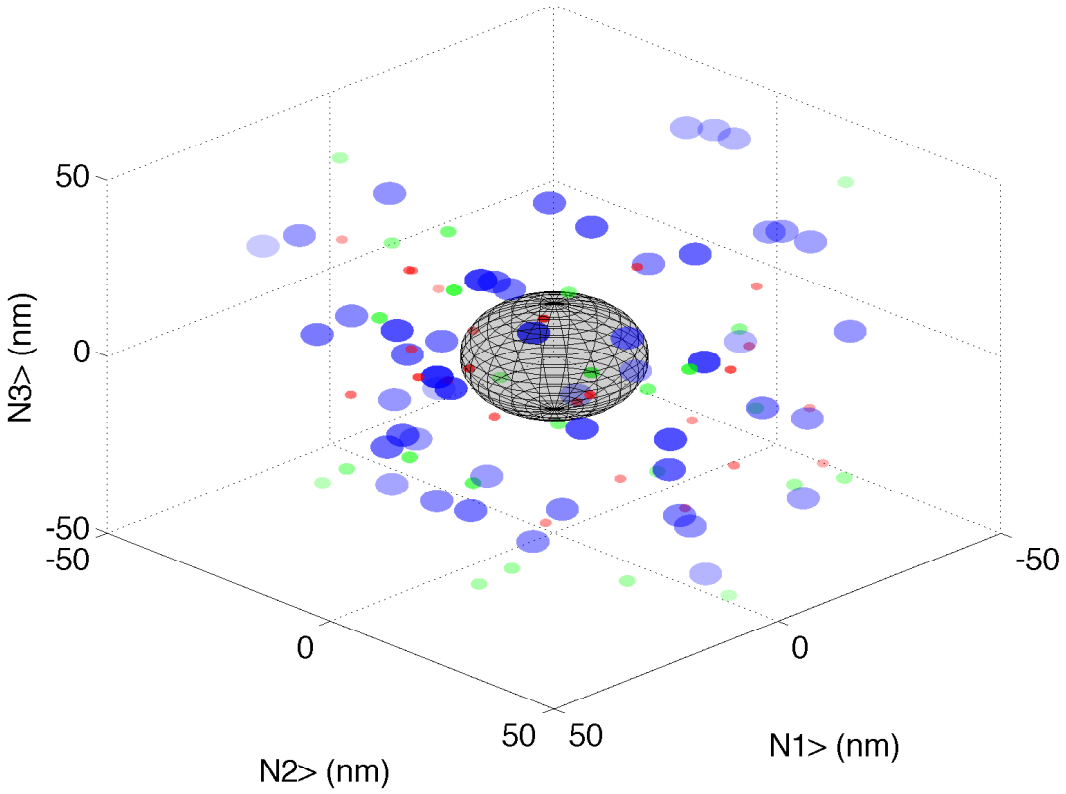

Figure 3.2. Initial positions. Cations (small) and anions (medium) are shown here interacting with a silicon dioxide nanoparticle (large). Ionic radii are enlarged for viewing.

Table D.2 shows the values of the simulation parameters. Many of these quantities are shown with an iterative subscript. The m_i mass terms are used in constructing the diagonal mass matrix M. The c_i, ϵ_i, and σ_i terms are used to define the Coulomb and Lennard-Jones constants in Eqns. (3.8) - (3.10). Specific values used are based upon which two ions' interaction is currently being evaluated. The Lennard-Jones constants for the sodium and potassium cations were taken from [134]. The Lennard-Jones constants for the nitrate anions, denoted †, were approximated. It is assumed that this approximation did not affect the testing of the modified formulation effi-

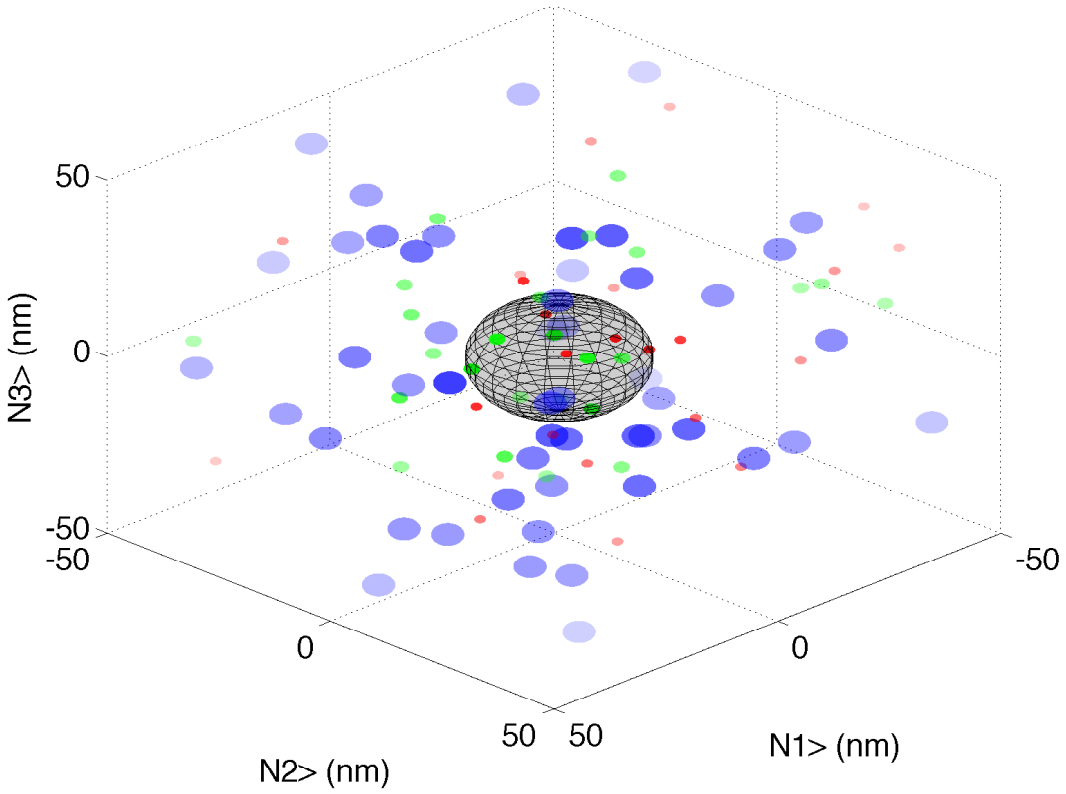

Figure 3.3. Final positions for system using the original Nóse-Hoover thermostat.

cacy. Relative permittivity is taken as an average of the quantities for sodium and potassium nitrate.

3.6 Results Analysis

Figures 3.5 and 3.6 show the temperature response of the standard and modified thermostats, respectively. The subplots show fixed and stepped target temperature profiles. Stochastic forces are disabled to isolate the performance of the thermostat. Figures 3.7 and 3.8 show data from similar experiments, but with Brownian motion enabled to show the thermostat capacity to control these disturbances. It can be seen

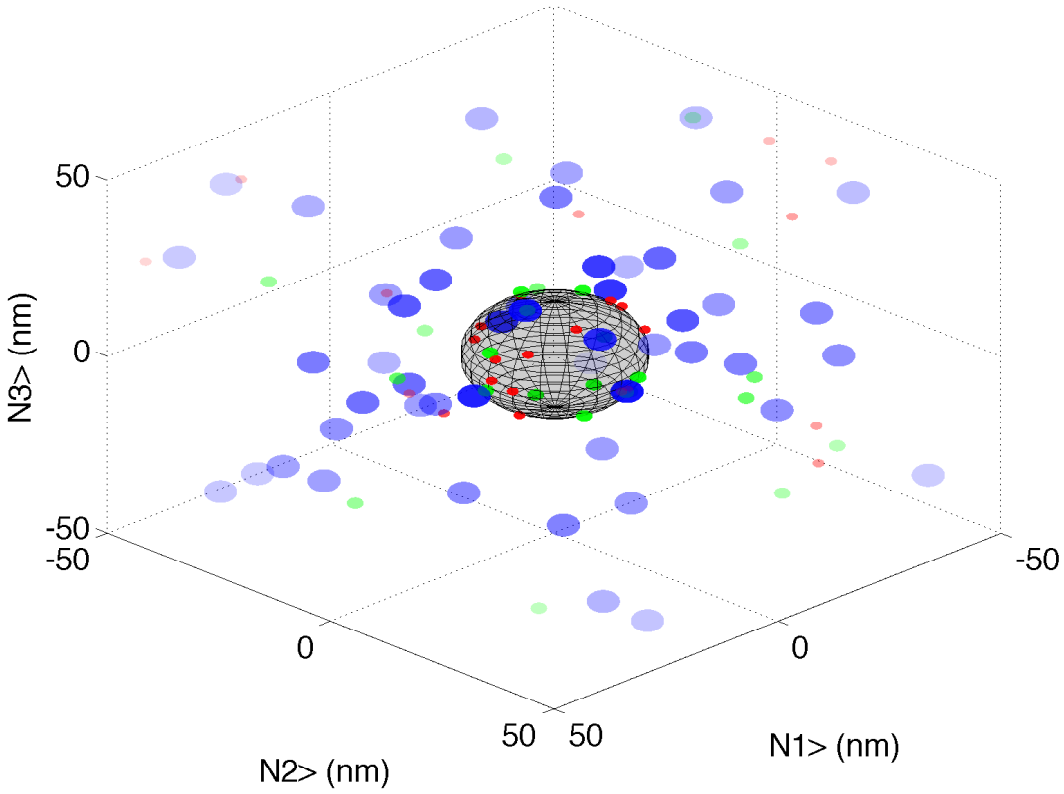

Figure 3.4. Final positions for system using the modified Nóse-Hoover thermostat.

the standard thermostat response is oscillatory with large oscillations. The modified thermostat response is also oscillatory, but the magnitude of those oscillations is significantly reduced.

For these simulations, O(Q) = 1 and O(k_d) = 10^{-2} where O(x) is the order of magnitude of x. It is of note that in the case of the stepped target temperature, the increase in system temperature is wholly in response to the thermostat. This increase in kinetic energy can be tracked by calculating thermostat work. There was no additional velocity scaling or momentum balance required.

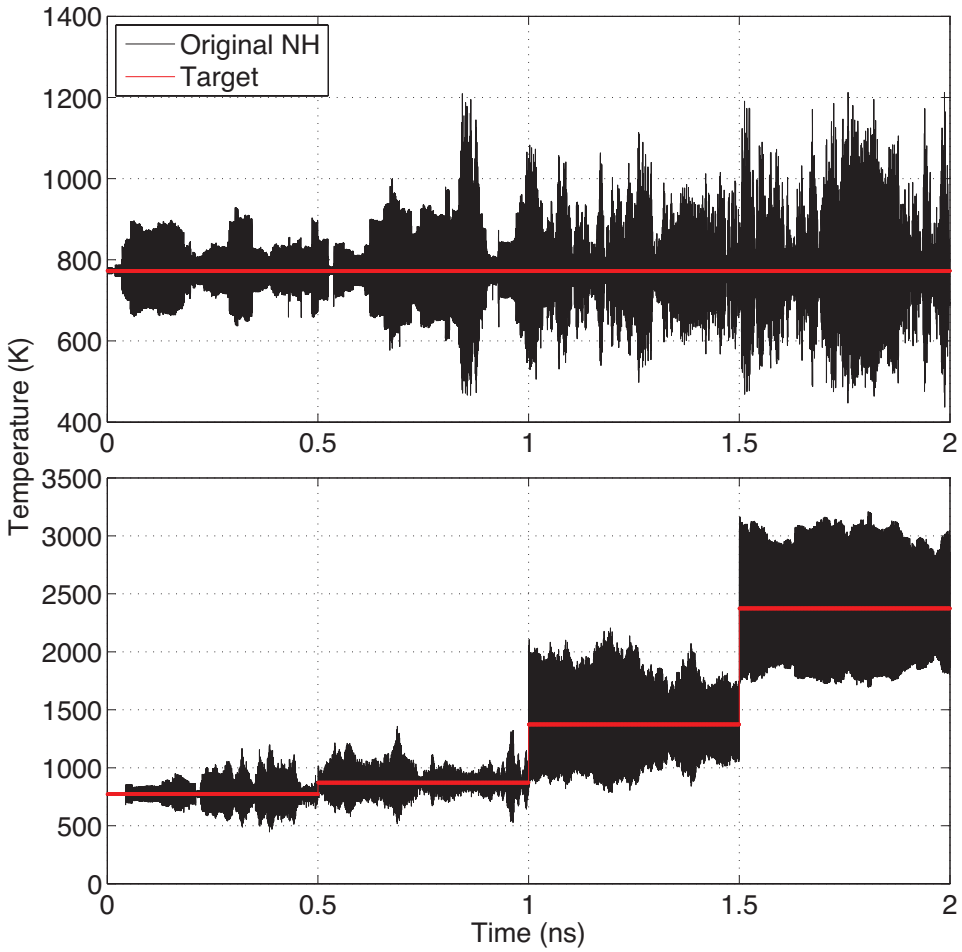

Figure 3.5. Temperature response of standard thermostat showing fixed and stepped target temperature profiles. Stochastic forces disabled.

Figures 3.6 and 3.8 show the modified thermostat significantly reduces the temperature oscillations associated with the standard formulation. For the data sets shown in Fig. 3.5 and 3.6, the standard deviation in temperature response is 112.2 and 0.6 for the standard and modified thermostats, respectively, for the fixed target temperature data sets. While the inclusion of stochastic forces affects time evolution, no significant change in overall behavior is observed. The modified thermostat is able to tightly maintain the constant temperature constraint regardless of the presence of stochastic forces. These figures also show that both thermostats respond well to

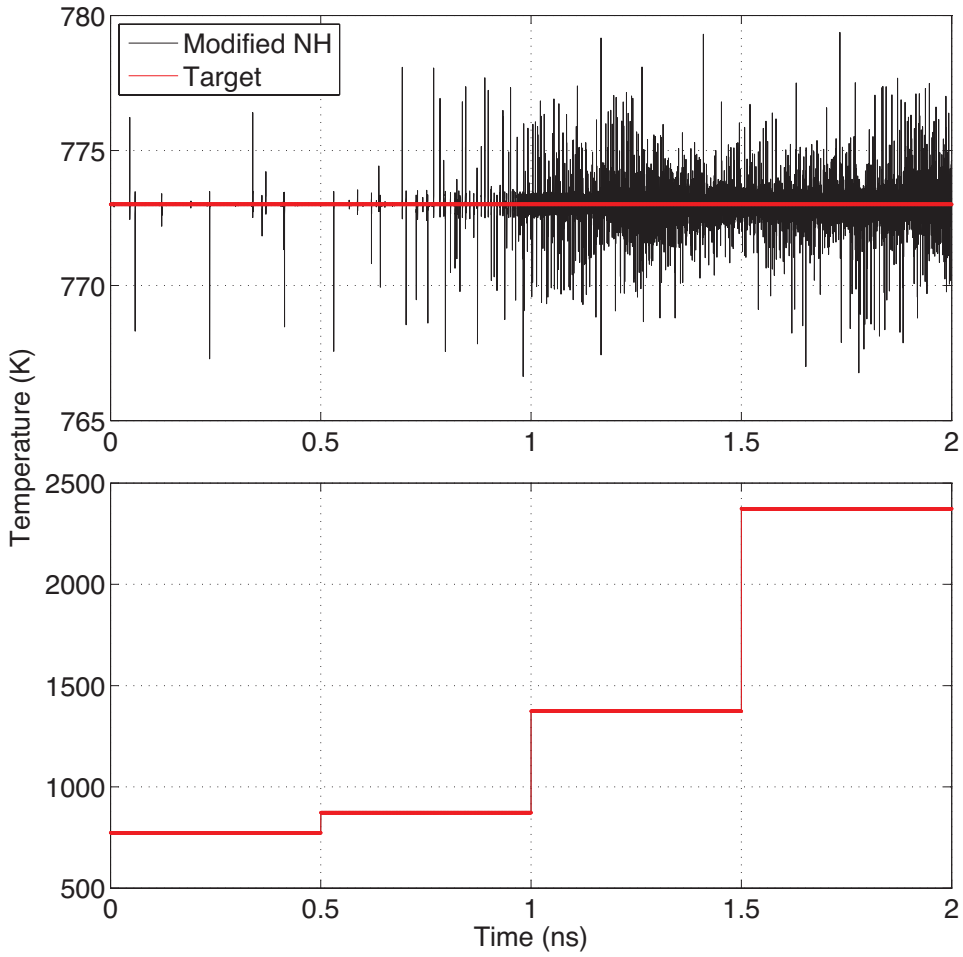

Figure 3.6. Temperature response of modified thermostat showing fixed and stepped target temperature profiles. Stochastic forces disabled.

instantaneous changes in target temperature, producing short-term mean temperatures consistent with the target. It can also be seen that oscillation magnitude is proportional to target temperature.

Figure 3.9 shows the effects of varying the modified thermostat mass Q and derivative gain k_d on temperature response. The magnitude of Q increases by orders of magnitude from Figs. 3.9a - 3.9d. As the value of Q is increased, the frequency of the response oscillations is decreased and the relative time required to approach a bounded steady state increases.

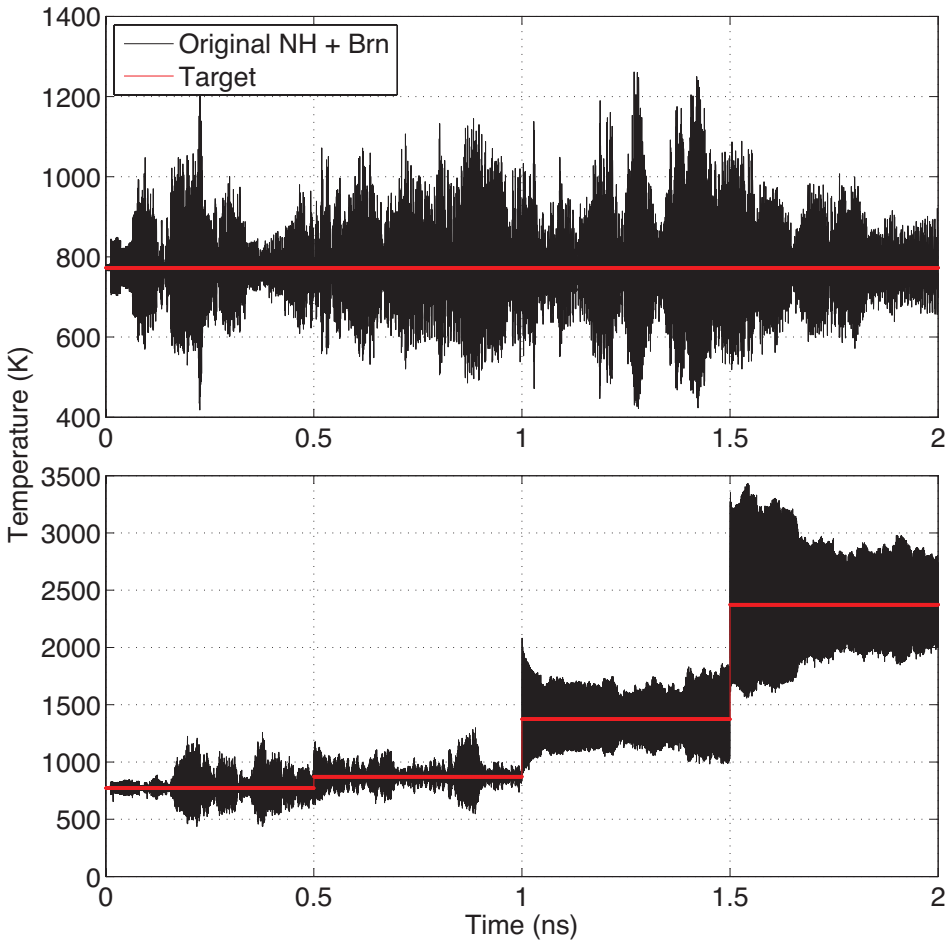

Figure 3.7. Stochastic forces enabled.

The solid dark line indicates $O(k_d) = 10^{-5}$, the smallest derivative gain used in the experiment. It can be seen that a low k_d gain leads to a highly oscillatory response. The highest derivative gain is indicated by the dashed lighter line and creates the least oscillatory response. The dataset shown by the dashed lighter line in Fig. 3.9a has the same Q and k_d gains as the simulation data shown in Figs. 3.5 - 3.8. It can be seen that as k_d approaches zero, the modified thermostat becomes the standard. These results show the Nóse-Hoover response behavior shares a degree of similarity with that of a classic spring-mass-damper system.

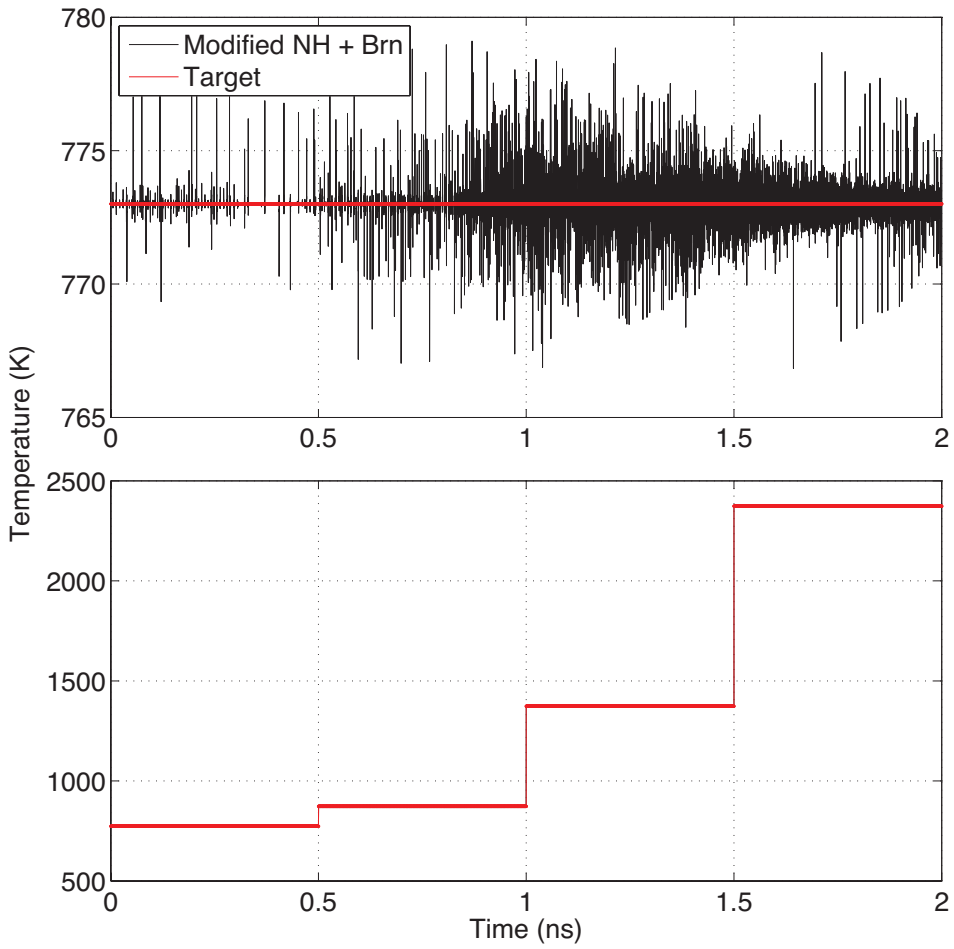

Figure 3.8. Stochastic forces enabled.

Figure 3.10 shows the effects of varying Q and k_d on the simulation time. Simulation time using the standard thermostat is included in Fig. 3.10 for comparison and is shown as the lowest point on the log scale. Unfortunately, the inclusion of the $\dot{\zeta}$ term in the modified control law adversely affects the time required to the run the simulation. It can be seen that lower Q and higher k_d values generally trend toward longer simulation times. Low Q values lead to high frequency response oscillations. Large k_d values create a "stiff" system with large thermostat terms and large resultant accelerations requiring small integration time steps to fully capture.

Considering the trends shown in Figs. 3.9 and 3.10, investigators may select gains for favorable responses with acceptable simulation times. Actual values of ideal gains will be greatly dependent on the nature and scale of the model.

Figure 3.11 shows simulation energy over time. Included is the kinetic energy, potential energy, work done by the thermostat, and total as per Eqn. (3.16). Figures 3.11(a) and 3.11(b) show systems using the standard and modified thermostats, respectively. The kinetic and potential energies are shown by the solid lighter lines. The kinetic energy is the oscillatory and relative smooth lines seen at the top of Figs. 3.11a and 3.11b, respectively. The work done by the thermostat is calculated and shown by the dashed line. The dark solid line, denoted as Total, shows that Eqns. (3.14) - (3.16) accurately measure the energy shared with the reservoir and that total energy is conserved. The data in this plot comes from a system with stochastic forces disabled as these forces would introduce energy not originating from the reservoir. The work done by these stochastic forces could be easily calculated if needed. Similarly to Figs. 3.5 - 3.8, $\mathrm{O}(Q) = 1$ and $\mathrm{O}(k_d) = 10^{-2}$.

Figure 3.11(c) shows a comparison of the system energies from Figs. 3.11(a) and 3.11(b) plotted on the same scale. Note that this quantity is the sum of kinetic and potential energy. It may be seen in Fig. 3.11(c) that the modified thermostat allows the system to evolve into a low potential state significantly faster than the standard. While the modified control law increases simulation time, there is a significant reduction in integration steps required to reach this low-potential state, allowing overall savings in time.

3.7 Conclusions

This chapter presented an investigation of a modified form of the popular Nóse-Hoover thermostat. This modification was intended to reduce the temperature re-

sponse oscillations that consistently appear when using the standard Nóse-Hoover formulation. The results showed that this modification significantly reduced the magnitude of these oscillations. For a study of 100 bodies at a fixed target temperature, the standard deviation in temperature error was 112.2 and 0.6 for the standard and modified thermostats, respectively.

Data plots also showed the interesting result of a swift system evolution to a low-potential state. This result is consistent with the idea that dynamic systems evolve into a low-potential equilibrium state. In the plots of final state, it can be seen that this low potential state is consistent with an accumulation of cations on the surface of the negatively charged silicon dioxide cluster. This result highlights the possibility of this modified thermostat being used to find low-potential states of other dissimilar systems.

This modified thermostat has a set of user specified parameters that may be tuned to control temperature response. These parameters should be selected with care as poor choices can lead to poor results - "stiff" systems with long simulation runtimes or "loose" systems with minimal reduction of temperature oscillations.

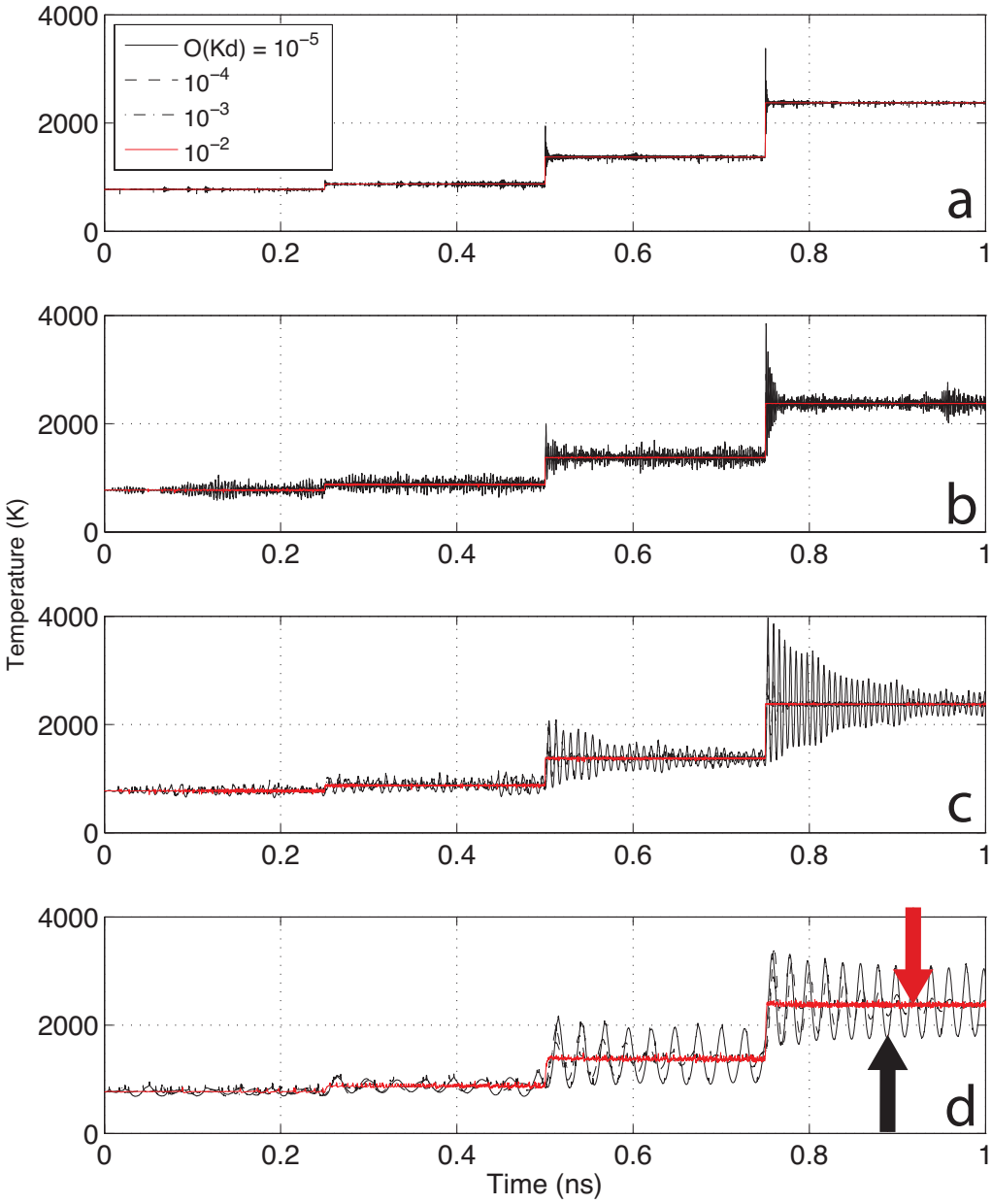

Figure 3.9. Effect of varying gains on response. (a) O(Q) = 1; (b) O(Q) = 10; (c) O(Q) = 100; (d) O(Q) = 1000. Insert shows the order of the derivative gain.

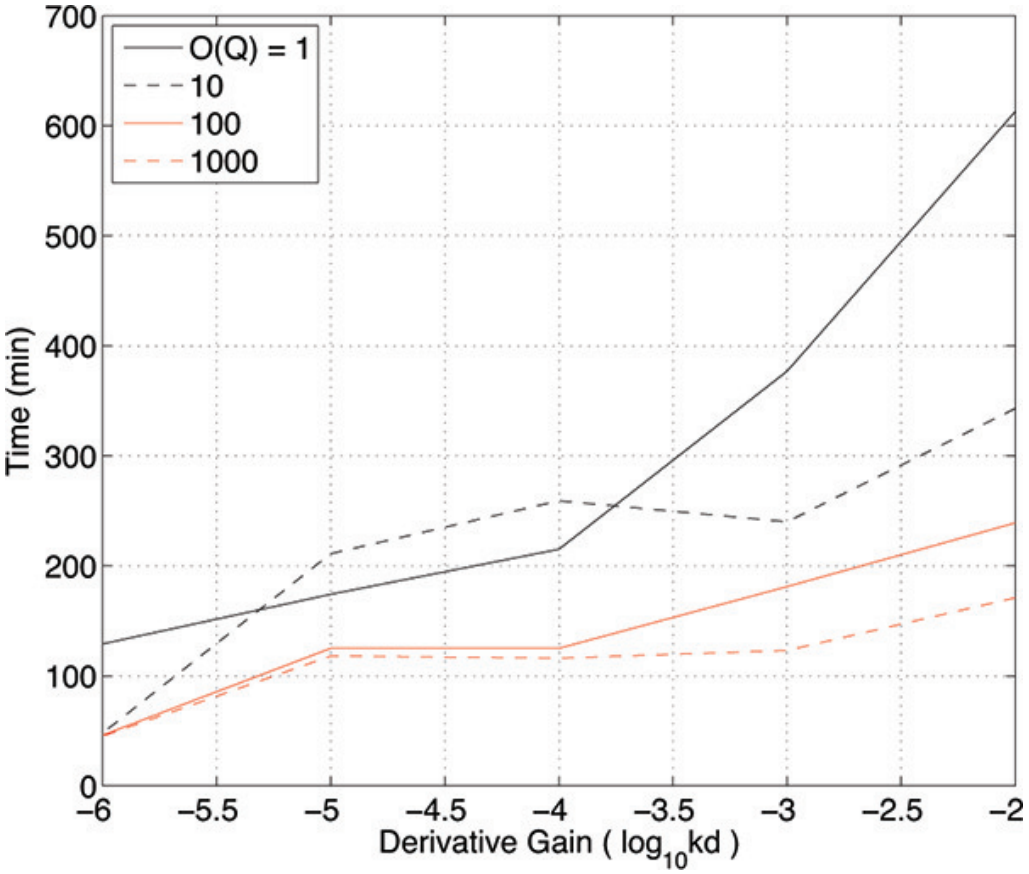

Figure 3.10. Effect of varying gains on simulation time.

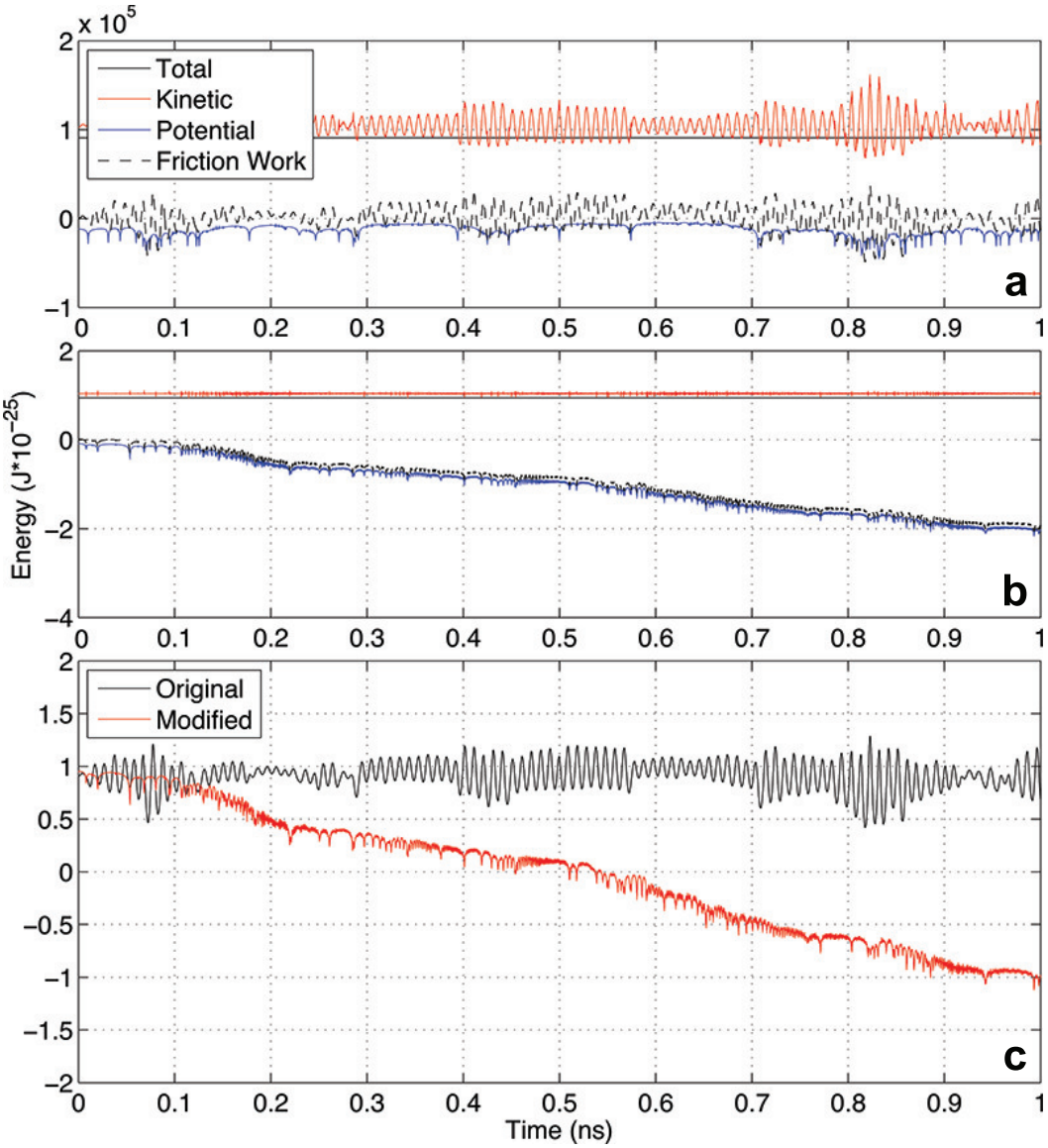

Figure 3.11. System energy and thermostat work. (a) standard thermostat; (b) modified thermostat; (c) comparison of total system energy from (a) and (b).

CHAPTER 4

ATOMISTIC REGIME MULTISCALE METHOD

4.1 Introduction

This chapter presents an extension to the established multiscale method that is applicable to temperature controlled atomistic systems. As mentioned before, a key component in the derivation of the continuum multiscale method is the large disproportionality between the mass and viscous damping terms often seen at that scale. This disproportinality is leveraged to eliminate portions of the active and dissipative forces from the dynamics, significantly reducing the computation time required to generate time histories. As atomistic simulations also suffer from burdensome computation times, a similar formulation for the atomistic regime was desired.

Since viscous damping is a continuum approximation of atomistic interactions, it is only appropriate for bodies large enough to satisfy the continuum assumption. The role of viscous friction in continuum analysis is to dissipate the kinetic energy of the modeled bodies into the surrounding medium. When that continuum medium is replaced with discrete bodies, that dissipation of energy then becomes a redistribution of energy amongst the modeled bodies. A useful replacement for viscous friction for atomistic bodies is one of the thermostats described in Chapter 3. To satisfy the multiscale analysis, the thermostat selected should be force based with terms large enough to define a significant disproportinoality in the equations of motion. Attempts at scaling using the original Nóse-Hoover formulation were ineffective as the thermostat coordinate ζ was shown to not be large enough to justify the scaling [135]. However, the modified formulation seen in Chapter 3 introduced the thermostat

speed $\dot{\zeta}$ into the control law which was shown to be large enough to justify the scaling [136]. This modified Nóse-Hoover thermostat was therefore selected to form the new atomistic multiscale formulation.

However, simply replacing viscous friction with a thermostat does not guarantee that the multiscale analysis is justified. Defining the system at a characteristic time scale traditionally used for atomistic studies yields a multiscale analysis result suggesting that no scaling may be used - it is only when the system is defined at a larger time scale that the analysis suggests scaling is appropriate. These larger time scales correlate to the higher tiers of resolution and coarse graining used by other multiscale approaches [65, 67]. By defining all bodies at these larger time scales, the disproportionality between terms can be leveraged to justify a scaling of the large forces that bring all terms in the equations of motion closer to the order of the mass.

The mixture selected for simulation is a eutectic blend of molten sodium and potassium nitrate. A target system temperature T_d of 773 K was selected such that the salt mixture is molten. These salts are therefore ionized into sodium and potassium cations and nitrate anions, wholly constituting the solvent and defining its properties. All ions are modeled as spherical bodies, including the trigonal planar nitrate anions. As the four atoms in a nitrate anion remain bonded at the simulated temperature, they are therefore considered a rigid body. Each body has three degrees of freedom associated with translation. Potential forces all act along vectors passing through centers of mass, so no torques are created. Since there are no torques, rotations are not considered. It was assumed that modeling the ions as spheres and eliminating the degrees of freedom associated with rotations and the internal bonds of the nitrate anions would not significantly affect the testing of the atomistic multiscale formulation efficacy.

A negatively charged cluster of several silicon dioxide molecules is included to create a potential gradient. As the selected target temperature is well below the melting point of silicon dioxide, this cluster of molecules will not ionize and dissociate and is thus treated as a single spherical rigid body.

4.2 Disproportionality Assumptions

Consider the equation of motion for a particle model using the modified Nóse-Hoover thermostat presented in Chapter 3:

$$M\ddot{\mathbf{q}} \ + \ (\ \zeta \ + \ k_d\dot{\zeta}\ )M\dot{\mathbf{q}} \ = \ \mathbf{\Gamma}(\mathbf{q}) \tag{4.1}$$

where $M \in \mathbb{R}^{N\times N}$ denotes the symmetrical mass matrix where N is the degrees of freedom, $\mathbf{q} \in \mathbb{R}^{N\times 1}$ is a vector containing the generalized coordinates with time derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ denoting the generalized speeds and accelerations, respectively, ζ and $\dot{\zeta}$ are the scalar thermostat terms with gain k_d, and $\mathbf{\Gamma} \in \mathbb{R}^{N\times 1}$ is the generalized active forces. Note that since this is a particle model, the mass matrix is not a function of the states. Recall that ζ and $\dot{\zeta}$ are the new generalized coordinate and speed associated with the thermostat where $\dot{\zeta}$ is calculated as the scaled error in instantaneous kinetic energy:

$$\dot{\zeta} \ = \ \Big(\dot{\mathbf{q}}^T M\dot{\mathbf{q}} \ - \ Nk_BT_d\Big)\frac{1}{Q} \tag{4.2}$$

In the established multiscale method, a small parameter ϵ was defined by the ratio of the damping term to the mass. Since the damping term has been replaced by the thermostat, this small parameter may now be defined as:

$$\begin{gathered} \epsilon \ = \ \frac{m}{\beta} \ \longrightarrow \\ \epsilon \ = \ \frac{m}{k_d\dot{\zeta}m} \ = \ \frac{1}{k_d\dot{\zeta}} \end{gathered} \tag{4.3}$$

	selected time scale	$O(\frac{1}{\zeta})$	$O(\frac{1}{k_d\dot{\zeta}})$	number of scaling factors
Case I	ps	$< 10^0$	10^0	0
Case II	ns	10^0	10^{-3}	1
Case III	μs	10^{-3}	10^{-6}	2

Table 4.1. Equations of motion disproportionalities.

where m is used to denote a mass extracted from the mass matrix. It is of note that, unlike the definition of ϵ used in the established multiscale method, this new small parameter is independent of the mass. In order to develop the multiscale analysis, it must be shown that this parameter ϵ is significantly small.

The magnitude of the small parameter ϵ can be manipulated through the selection of the characteristic time scale. The result then is that as the atomistic bodies are observed from increasingly larger time scales the disproportionality between the terms in Eqn. (4.1) becomes more extreme. Depending on the time scale selected, and the resulting magnitude of the thermostat terms, three cases are possible. These cases are summarized in Table 4.1. These three cases produce three distinct results from the multiscale analysis: no scaling; one scaling factor; and two scaling factors.

Case I was encountered when the system was defined at the picosecond scale. This time scale is within the range traditionally selected for atomistic simulations as it keeps all terms in the equations of motion close in magnitude to one. In this case, both the ζ and $\dot{\zeta}$ terms are not significantly large. This results in a small thermostat term compared to the mass term and active forces. Since no significant disprorportionality exists, multiscale analysis shows that scaling is not justified.

Case II was encountered when the characteristic time scale *was increased from picoseconds to nanoseconds*. This time scale is useful for simulations of sub-micron sized bodies such as nanoparticles and coarse-grained protein polymers. Selection of this time scale yielded large $\dot{\zeta}$ terms but still relatively small ζ terms. Since a single large disproportionality exists, a single small parameter ϵ may be defined. The resulting multiscale analysis produces a scaled equation of motion with a single scaling factor.

Case III was encountered when the characteristic time scale *was increased from picoseconds to microseconds*. Because of the extreme time scale selected, both the ζ and $\dot{\zeta}$ terms have grown significantly large. Since $\dot{\zeta}$ is several orders of magnitude larger than ζ, two disproportionalities exist and two small parameters, ϵ_1 and ϵ_2 may be defined. The resulting multiscale analysis is similar to that discussed in [79] and results in a scaled equation of motion using two scaling factors.

Since Case III was only encountered when the system was defined at an extreme characteristic time scale, Case II was investigated and is assumed for the remainder of this chapter. Let us therefore assume that the ratios defined for Case II in Table 4.1 are true. By making these assumptions, an atomistic model may be simulated for longer time periods in reasonable real time. These assumptions will be later confirmed by the results to justify the scaling procedure.

4.3 Multiscale Derivation

Consider again the equation of motion for a particle model using the modified Nóse-Hoover thermostat given by Eqn. (4.1). If the Case II conditions in Table 4.1

are true, then the ζ term is approximately on the same order as the mass and the $\dot{\zeta}$ term is significantly larger. Normalizing Eqn. (4.1) by the $k_d\dot{\zeta}M$ term yields:

$$\begin{aligned} \frac{M^{-1}M}{k_d\dot{\zeta}}\ddot{\mathbf{q}} \;+\; \frac{M^{-1}\zeta M}{k_d\dot{\zeta}}\dot{\mathbf{q}} \;+\; \dot{\mathbf{q}} \;&=\; \frac{M^{-1}\,\mathbf{\Gamma}(\mathbf{q})}{k_d\dot{\zeta}} \\ \frac{1}{k_d\dot{\zeta}}\ddot{\mathbf{q}} \;+\; \frac{\zeta}{k_d\dot{\zeta}}\dot{\mathbf{q}} \;+\; \dot{\mathbf{q}} \;&=\; \frac{M^{-1}\,\mathbf{\Gamma}(\mathbf{q})}{k_d\dot{\zeta}} \end{aligned} \tag{4.4}$$

The $k_d\dot{\zeta}$ term can therefore be used to define a small parameter ϵ:

$$\frac{1}{k_d\dot{\zeta}} \;=\; \epsilon\;(1\;ns) \tag{4.5}$$

such that ϵ is unitless and small, according to Table 4.1 and the (1 ns) is a constant. Note that since Case II was assumed, a characteristic time scale of (ns) was selected for this work. Rearranging Eqn. (4.4) and introducing ϵ yields:

$$\begin{aligned} \mathbf{0} \;&=\; \epsilon\;(1\;ns)\;\ddot{\mathbf{q}} \;+\; \epsilon\;(1\;ns)\;\zeta\dot{\mathbf{q}} \;+\; \dot{\mathbf{q}} \;-\; \frac{M^{-1}\,\mathbf{\Gamma}(\mathbf{q})}{k_d\dot{\zeta}} \\ \mathbf{0} \;&=\; \epsilon\,\ddot{\bar{\mathbf{q}}} \;+\; \epsilon\,\zeta\dot{\bar{\mathbf{q}}} \;+\; \dot{\mathbf{q}} \;-\; \frac{M^{-1}\,\mathbf{\Gamma}(\mathbf{q})}{k_d\dot{\zeta}} \end{aligned} \tag{4.6}$$

According to the method of multiple scales [51], the small parameter ϵ can be used to decompose time into different scales according to $T_i \;=\; \epsilon^i t$. The time derivatives $\mathbf{q}$ and $\ddot{\mathbf{q}}$ are then expanded into series:

$$\dot{\mathbf{q}} \;=\; \frac{d\mathbf{q}}{dt} \;=\; \epsilon^0\frac{\partial\mathbf{q}}{\partial T_0} \;+\; \epsilon^1\frac{\partial\mathbf{q}}{\partial T_1} \;+\; \epsilon^2\frac{\partial\mathbf{q}}{\partial T_2} \;+\; \cdots \tag{4.7}$$

$$\ddot{\bar{\mathbf{q}}} \;=\; \frac{d^2\bar{\mathbf{q}}}{dt^2} \;=\; \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\epsilon^i\epsilon^j\frac{\partial^2\bar{\mathbf{q}}}{\partial T_i\;\partial T_j} \tag{4.8}$$

Substituting these expansions into Eqn. (4.6) and collecting terms by increasing order of ϵ yields:

$$\mathbf{0} \;=\; \epsilon^0\Big(\;\frac{\partial\mathbf{q}}{\partial T_0} - \frac{M^{-1}\,\mathbf{\Gamma}(\mathbf{q})}{k_d\dot{\zeta}}\;\Big) \;+\; \epsilon^1\Big(\;\frac{\partial^2\bar{\mathbf{q}}}{\partial T_0^2} \;+\; \zeta\frac{\partial\bar{\mathbf{q}}}{\partial T_0} \;+\; \frac{\partial\mathbf{q}}{\partial T_1}\;\Big) \;+\; \cdots \tag{4.9}$$

Knowing that ϵ is small and $\epsilon^0 = 1$, and assuming that the ϵ^1 acceleration terms are not excessively large, it can be seen that the first term on the right hand side of Eqn. (4.9) must largely cancel such that the whole expression equals zero. That first term contains the generalized active forces and the $\dot{\zeta}$ term of the modified thermostat. Separating those forces into large and small parts yields:

$$\frac{\partial \mathbf{q}}{\partial T_0} \; - \; \frac{M^{-1}\,\mathbf{\Gamma}(\mathbf{q})}{k_d \dot{\zeta}} \; = \; (a_1 + a_2)\Big(\frac{\partial \mathbf{q}}{\partial T_0} \; - \; \frac{M^{-1}\,\mathbf{\Gamma}(\mathbf{q})}{k_d \dot{\zeta}} \; \Big) \tag{4.10}$$

where the scalars $a_1 + a_2 = 1$ and $a_1 \gg a_2$. Substituting this definition into the equation of motion yields:

$$\mathbf{0} \; = \; M\ddot{\mathbf{q}} \; + \; \zeta M\dot{\mathbf{q}} \; + \; (a_1 + a_2)\Big(\; k_d \dot{\zeta} M\dot{\mathbf{q}} \; - \; \mathbf{\Gamma}(\mathbf{q}) \; \Big) \tag{4.11}$$

If a portion of the generalized active and thermostat forces cancel, those canceled forces can be eliminated from the model without affecting the dynamics. Assuming the bulk of these forces - those associated with the a_1 term - cancel such that Eqn. (4.9) is true, removing them from the model and rearranging yields the scaled temperature controlled equation of motion:

$$M\ddot{\mathbf{q}} \; + \; (\; \zeta + a_2 \; k_d \dot{\zeta} \;) M\dot{\mathbf{q}} \; = \; a_2 \; \mathbf{\Gamma}(\mathbf{q}) \tag{4.12}$$

This scaled equation of motion suggests that for systems satisfying the initial assumptions, specifically that the $\dot{\zeta}$ term of the modified thermostat is significantly large, the active forces and $\dot{\zeta}$ thermostat term may be scaled by the small parameter a_2. Note that a_2 is a user-specified scaling factor that removes a significant portion of the forces from the calculations. The practical effect of this scaling is an elimination of many of the high-frequency vibrations that require small integration time steps to resolve but do not contribute to the overall time evolution. By removing these vibrations, the integration time step may be increased - usually by an order of magnitude - significantly reducing the real computation time required to capture the time evolution.

The actual value of a_2 can be adjusted to minimize relative differences between the scaled and unscaled results, but usually $a_2 \geq \epsilon$ is an approximate starting point.

In the established multiscale formulation, the damping term is a constant and the disproportionality between the mass and damping terms is therefore also a constant. However, in the atomistic multiscale formulation presented above, the time variant $\dot{\zeta}$ term is used. This time variant disproportionality raises the possibility of time variant scaling. In this chapter however, the a_2 term is assumed to be constant.

4.4 Model Description and Equations of Motion

An ionic particle model was selected with canonical ensemble constraints placed on the number of bodies, volume, and temperature. The temperature constraint is enforced using the modified Nóse-Hoover thermostat as previously discussed. The equation of motion for this system is the multiscale temperature controlled equation formulated in Sec. 4.3 and written in the general form:

$$M\ddot{\mathbf{q}} \; + \; (\; \zeta + a_2 k_d \dot{\zeta} \;) M\dot{\mathbf{q}} \; = \; a_2 \, \mathbf{\Gamma}(\mathbf{q}) \tag{4.13}$$

The active forces in this model are:

$$\mathbf{\Gamma}(\mathbf{q}) \; = \; \mathbf{\Gamma}_{Pot}(\mathbf{q}) \; + \; \mathbf{\Gamma}_{Vol}(\mathbf{q}) \tag{4.14}$$

where $\mathbf{\Gamma}_{Pot}(\mathbf{q})$ denotes forces arising from potential interactions and $\mathbf{\Gamma}_{Vol}(\mathbf{q})$ denotes potential-like volume constraint forces. It can be seen that the modified Nóse-Hoover thermostat force is included on the left-hand-side of Eqn. (4.13).

To test the efficacy of the proposed multiscale formulation, the multiscale temperature controlled system is compared against an unscaled temperature controlled system with the equation of motion:

$$M\ddot{\mathbf{q}} \; + \; (\; \zeta + k_d \dot{\zeta} \;) M\dot{\mathbf{q}} \; = \; \mathbf{\Gamma}(\mathbf{q}) \tag{4.15}$$

and an unscaled system using the standard Nóse-Hoover thermostat with equation of motion:

$$M\ddot{\mathbf{q}} \ + \ \zeta M\dot{\mathbf{q}} \ = \ \mathbf{\Gamma}(\mathbf{q}) \tag{4.16}$$

4.5 Forces and Energy

Active forces in atomistic simulations are dominated by body-to-body potential interactions. These potential interactions are modeled as a combination of Coulomb and Lennard-Jones terms similar to the methods described in Chapters 2 and 3:

$$\Phi_{p,ij}(r_{ij}) \ = \ \frac{A}{r_{ij}^{12}} \ - \ \frac{B}{r_{ij}^{6}} \ + \ \frac{C_{ij}}{r_{ij}} \tag{4.17}$$

where the force arising from a potential interaction is found by taking the derivative:

$$\Gamma_p(r) \ = \ \frac{d\Phi_p}{dr}\bar{r} \tag{4.18}$$

The Lennard-Jones forces provide a small attractive force at large distances and a large repulsive term to model collisions. These potential interactions are calculated body-to-body with a specified threshold distance. Any interaction with a distance beyond this threshold is skipped to facilitate faster computations. The threshold is user-specified such that errors in potential energy calculation are minimized. The volume constraint is enforced by an uncharged potential interaction between each particle and the volume wall. This potential contains only a Lennard-Jones term in which the constants are defined by the body:

$$\begin{aligned} \Phi_{v,i}(r_i) \ &= \ \frac{A_i}{r_i^{12}} \ - \ \frac{B_i}{r_i^{6}} \\ \Gamma_v(r) \ &= \ \frac{d\Phi_v}{dr}\bar{r} \end{aligned} \tag{4.19}$$

Potential forces arising from gravity were not considered because, at the atomistic scale, these force are approximately ten orders of magnitude smaller than the Coulomb

and Lennard-Jones terms. While these forces can have significant effects over long time periods, they will not have a significant effect for shorter simulations.

To ensure energy conservation and dynamic consistency, the work done by the thermostat forces must be calculated. The derivative of work is defined at each time step and integrated:

$$W_{MNH} \;=\; \int_{t1}^{t2} \sum_{i=1}^{N} \Gamma_{MNHi} \cdot \dot{q}_i dt \;=\; \int_{t1}^{t2} -\big(\zeta \;+\; k_d\dot{\zeta}\big)\dot{\mathbf{q}}^T M \dot{\mathbf{q}} dt \tag{4.20}$$

where W is the work done. Energy conservation is then confirmed by summing the instantaneous kinetic and potential energies and subtracting the work done:

$$T \;+\; \mathbf{\Phi} \;-\; W \;=\; constant \tag{4.21}$$

where $\mathbf{\Phi}$ is the total potential energy including both the interaction and volume constraint potentials.

4.6 Determining Appropriate Q and k_d Values

As mentioned before, the value of Q is user-specified, but because of its interaction with $\dot{\zeta}$ according to Eqn. (4.2), it also affects the multiscale analysis presented in Sec. 4.3. Similarly, the value of k_d is user-specified and also plays a critical role in the derivation. Selection of these values then will greatly affect the efficacy of the proposed method.

The role of Q in thermostat response is similar to that of the mass in a classic spring-mass-damper system. Much like the relationship $\omega_n = \sqrt{k/m}$, as Q is increased, the frequency of oscillation in temperature response will similarly decrease. The value of Q was selected considering only the unscaled standard Nóse-Hoover system given by Eqn. (4.16) such that the period of temperature oscillation was approximately 7 ps and approximately three full periods could be observed in a 20 ps

time frame. Selection of larger values of Q will result in longer oscillation periods, but those oscillations grow in magnitude and are therefore not ideal for systems with significant initial errors in temperature.

The role of k_d and $\dot{\zeta}$ is similar to that of the damper in a spring-mass-damper system. Vary large Q values will diminish the effects of the $\dot{\zeta}$ "damping" term, requiring a larger k_d value to ensure temperature convergence in reasonable time. The value of k_d was therefore selected using the previously selected Q value and considering only the unscaled modified Nóse-Hoover system given by Eqn. (4.15). Small values of k_d lead to a lightly damped system requiring more time to reach target temperature. However, large k_d values create a "stiff" system resulting in fast convergence but ultimately poor computational performance: the thermostat forces become overly large, creating artificial vibrations in the motion of the bodies and smaller integration step sizes.

Selection of Q and k_d then follows this procedure: considering both the effects on temperature oscillation and desired initial errors, select Q for only the unscaled standard Nóse-Hoover system. Then, implement the unscaled modified Nóse-Hoover formulation and find an appropriate k_d that promotes convergence without negatively affecting computation time. With Q and k_d selected, consider that the magnitude of the disproportionality in the equations of motion will vary as the observed time scale is changed. The user can then define the time scale at which to observe the system and calculate the small parameter ϵ from short unscaled simulations. Once ϵ has been determined, an appropriate a_2 scaling factor can found with the useful initial guess $a_2 \approx \epsilon$.

4.7 Hardware and Simulation Parameters

Simulation files were coded in the C++ programming language using the CUDA [24] and Eigen [87] libraries. Parallel processing was used to concurrently resolve the numerous point-to-point potential interactions. Numerical integration was performed on a computation server with Intel Xeon E5-2680 v3 processors running Ubuntu 16.04 operating system. Parallelization was performed on an NVIDIA GTX 980 Ti GPU. The integration was performed by a Kutta-Merson algorithm [88] with adaptive integration step sizes for reducing the numerical error. Relative and absolute errors were 10^{-6} and 10^{-7}, respectively. A unit system of (10 yg, nm, ns) was selected to keep the mass of all ions in the 10^0 - 10^1 range. Table D.2 in Appendix D lists the values of the parameters used.

Initial positions are user-specified and can be seen in Fig. 4.1. Final positions of particles produced by the time evolution can be seen in Fig. 4.2. As time progresses, these particles disperse to form a more uniform distribution. Initial velocity directions are randomly generated but constrained by magnitude to ensure an error in initial system temperature such that convergence may be observed.

Presented data comes from systems with 1025 bodies. As the mixture is a eutectic blend of the salts, 256 of these bodies are sodium cations, 256 are potassium cations, and 512 are nitrate anions, plus the single large cluster of silicon dioxide. The user-specified parameters are the mass-like term associated with the thermostat Q, the derivative gain k_d, and, for the scaled system, the scaling factor a_2. The following values were selected: $Q = 6000$ (10)yg nm^2; $k_d = 0.3$; $a_2 = 10^{-3}$. The model volume is a cube of approximately 157 nm^3. The silicon dioxide cluster has a diameter of 4.2 nm.

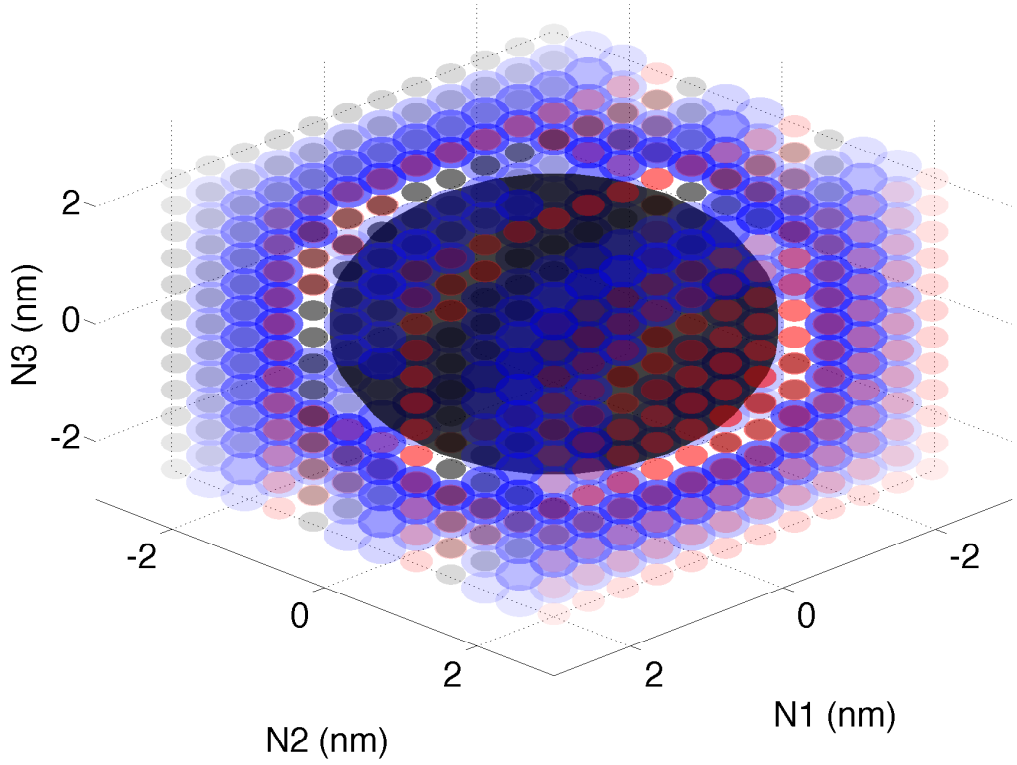

Figure 4.1. Initial positions. Smaller bodies are sodium and potassium cations. Larger bodies are nitrate anions. The large central body is the silicon dioxide cluster.

4.8 Results Analysis

Figure 4.3 shows a comparison of the temperature responses produced by the three simulations. The open circles are the system using the original Nóse-Hoover thermostat, the dashed line is the unscaled system, and the solid line is the scaled system.

It can be seen that the original Nóse-Hoover thermostat leads to a non-attenuating oscillatory response, which is expected. The frequency of these oscillations is regulated by the mass-like term Q in Eqn. (4.2). The unscaled modified thermostat significantly reduces the magnitude of these oscillations. The unscaled modified ther-

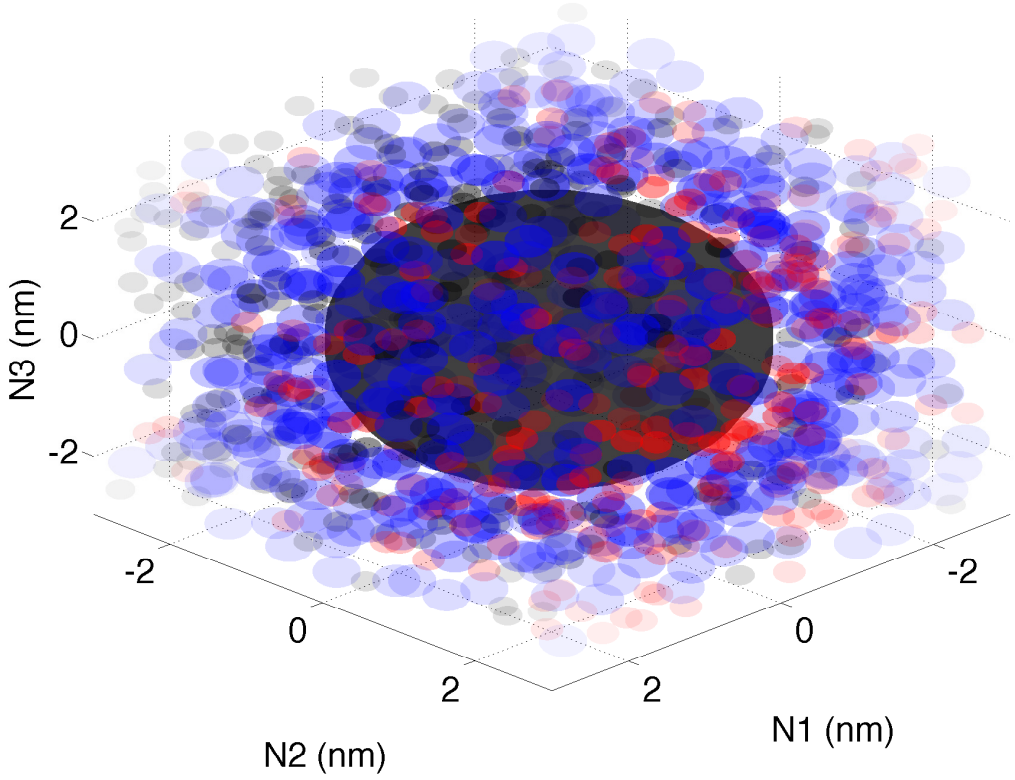

Figure 4.2. Final positions.

mostat also results in near-instant convergence of temperature to the target, and this temperature constraint is tightly maintained for the duration of the simulation. For comparison, the original formulation exhibits temperature errors of approximately $\pm 100\ K$ while the unscaled system exhibits errors of approximately $\pm 3\ K$. These results are consistent with those presented in Chapter 3. The scaled modified thermostat exhibits some oscillation initially, but these oscillations do attenuate with time, causing the instantaneous temperature to eventually settle around the target. Initial oscillations are approximately $\pm 50\ K$. Using this atomistic multiscale formu-

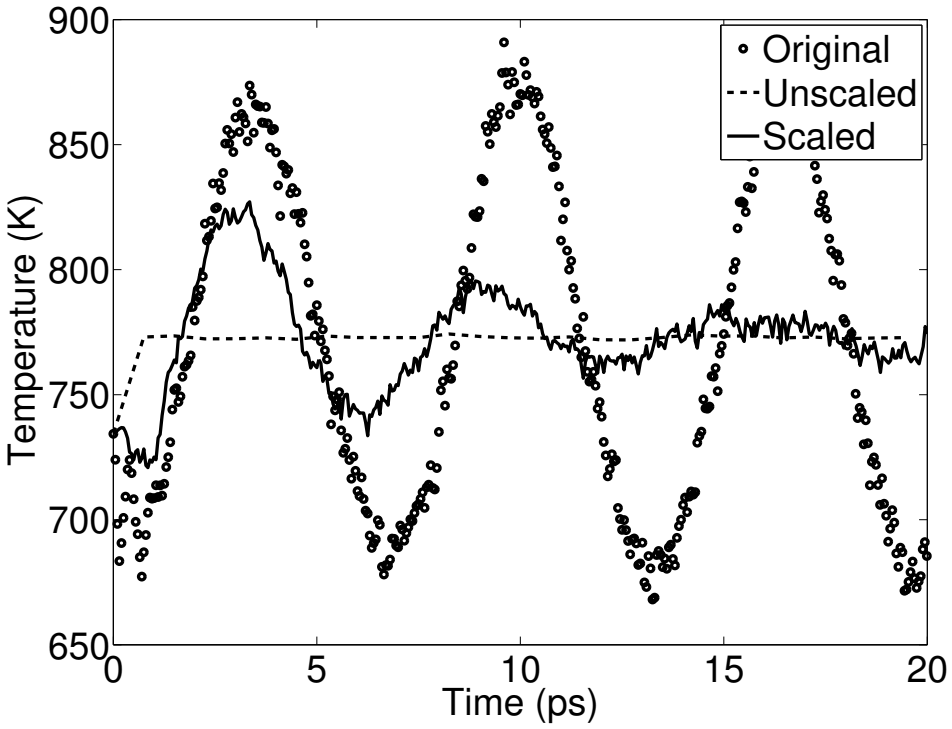

Figure 4.3. Temperature response for the three simulations.

lation, the constant temperature constraint is loosened during the transient state but maintained in the steady state.

Figures 4.4 - 4.6 show the system energies from the original thermostat, the unscaled modified thermostat, and the scaled modified thermostat systems, respectively. In these plots, the open circles are the potential energy, the dots are the kinetic energy, the solid line is the work done by the thermostat, and the dashed line is the total. It can be seen that Eqns. (4.20) and (4.21) successfully tracked the work done by the thermostat forces and that total energy is conserved.

It is of note that the total energy in Fig. 4.6 is increased compared to Figs. 4.4 and 4.5. This increase in total energy for the scaled system is not erroneous but an expected result of scaling the potential forces. Consider again the definition

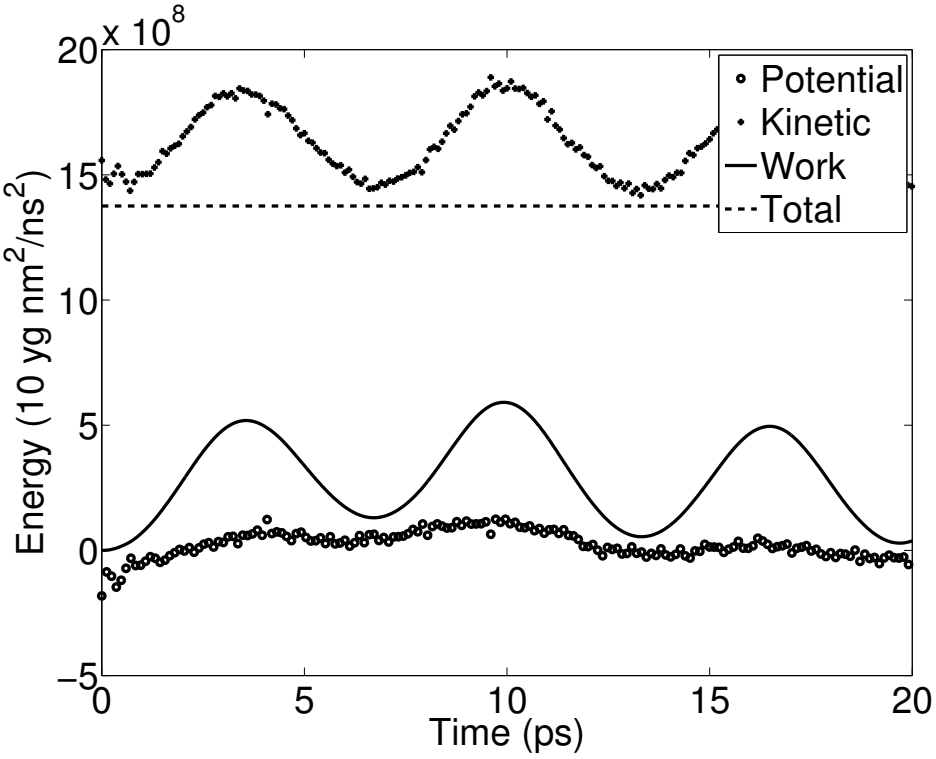

Figure 4.4. System energy using the original thermostat.

of potential energy and potential force given by Eqns. (4.17) - (4.19). It can be seen in these equations that potential energy is the antiderivative of potential force. Therefore, if the potential forces are scaled by the factor a_2, the potential energy must also be scaled by this same factor. This means that the potential energy of the system is reduced by a_2. It can be seen in Figs. 4.4 and 4.5 that the initial value of potential energy is negative. Hence, by scaling this negative value, the total calculated energy of the scaled system increases.

Phase plots of the thermostat variables ζ and $\dot{\zeta}$ are useful for justifying the disproportionality assumptions made in Sec. 4.2. From these plots we can observe the magnitudes of these variables at the initial conditions and as the system moves

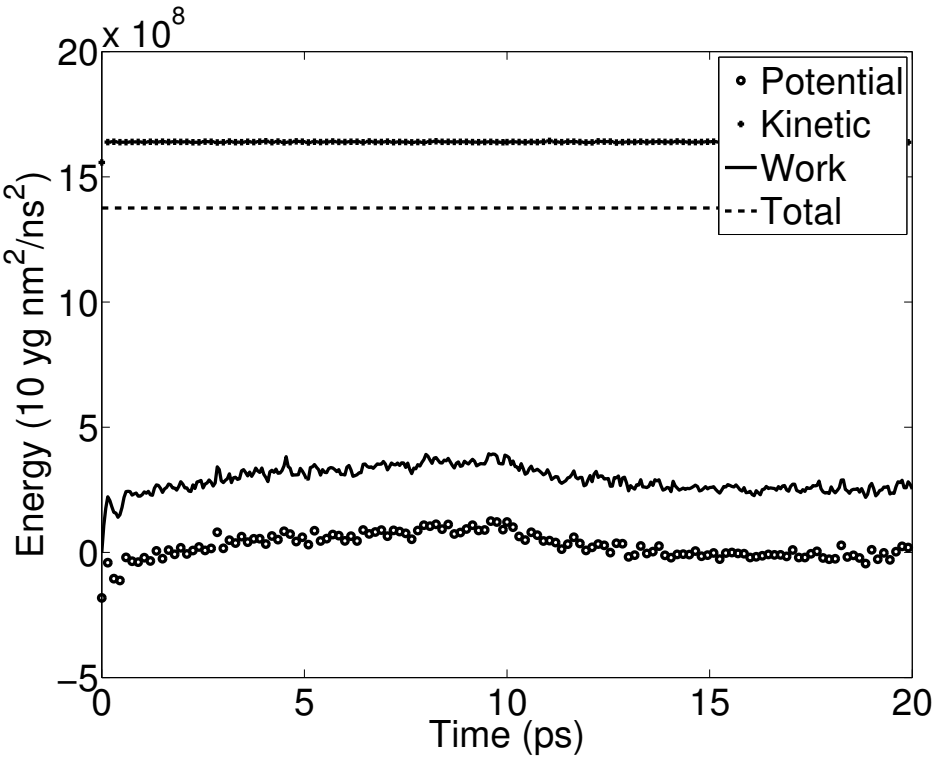

Figure 4.5. System energy using the unscaled modified thermostat.

towards an equilibrium. Data near the origin imply that the temperature constraint is tightly enforced.

Figure 4.7 shows the phase plots for the three simulations. The open circles are the system using the original Nóse-Hoover thermostat, the dashed line is the unscaled system, and the solid line is the scaled system. It can be seen that the original Nóse-Hoover thermostat produces a phase plot constrained within a limit cycle with small disturbances. This limit cycle coincides with the temperature oscillations seen in Fig. 4.3. As the temperature oscillations do not damp with time, the phase plot remains constrained within this limit cycle even for longer time evolutions. The unscaled modified thermostat produces a phase plot that quickly moves toward the origin. This fast convergence to the origin coincides with the fast convergence of temperature to

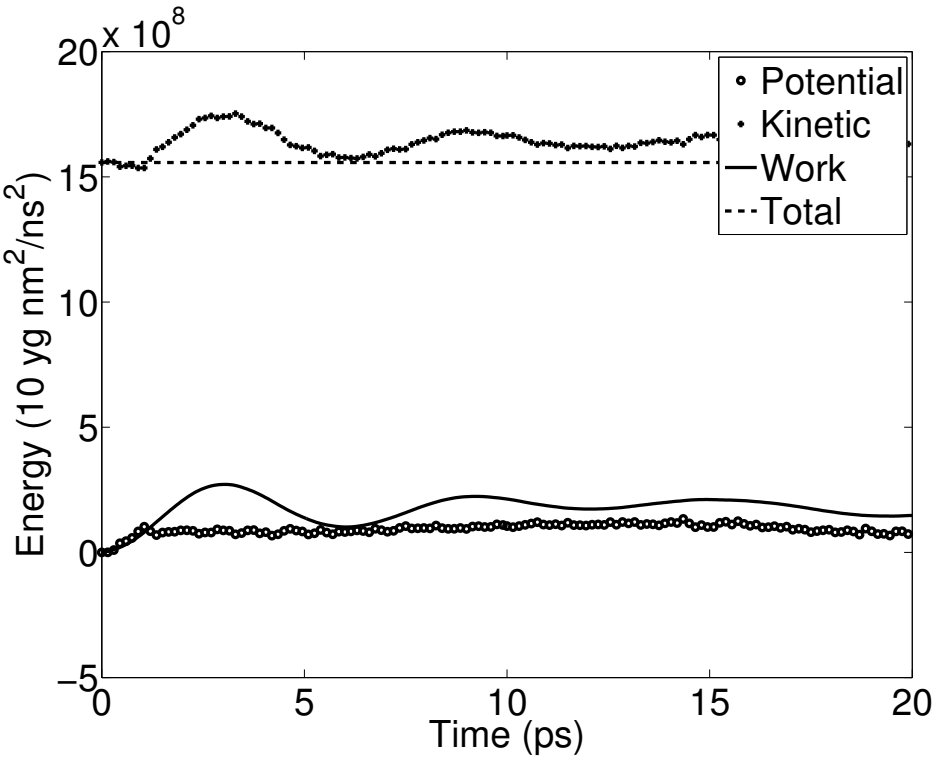

Figure 4.6. System energy using the scaled modified thermostat.

the target seen in Fig. 4.3. The phase plot for the scaled modified thermostat is still oscillatory, but decays towards the origin. The decaying oscillatory portion of this response coincides with the transient portion of the temperature response seen in Fig. 4.3. It is of note that both the unscaled and scaled systems never arrive and remain at the origin. As the systems are driven by inter-particle potential interactions, these plots will continually hover around the origin, but at significantly smaller values than those of the original thermostat limit cycle.

As mentioned, the phase plots are useful for justifying the disproportionality assumptions defined by Case II in Table 4.1. Note that these assumption should hold for both the unscaled and scaled systems. It can be seen in Fig. 4.7 that this

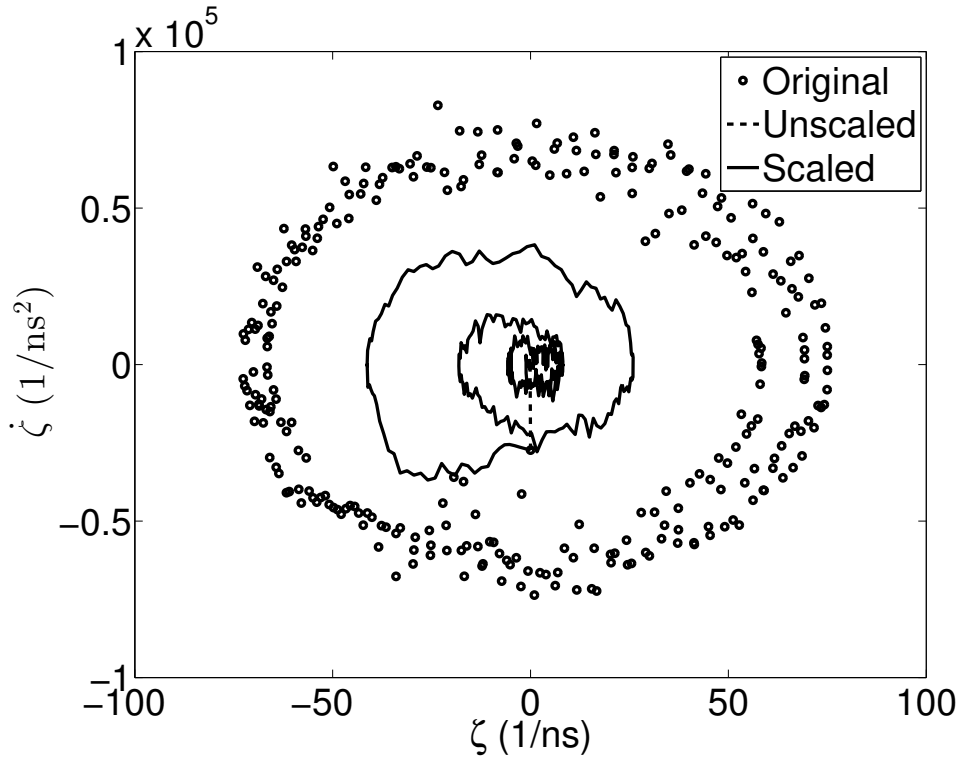

Figure 4.7. Phase plots for all three simulated systems.

assumption holds for scaled system, even as it oscillates around the origin. However, given the axes used, the behavior of the unscaled system cannot be easily observed.

Figure 4.8 shows the phase plot for only the unscaled system. It can be seen that the assumption defined by Case II in Table 4.1 holds for the unscaled system as well. However, these phase plots do highlight a significant difference between the existing continuum multiscale formulation and the atomistic formulation presented here: while the continuum equation of motion has a constant disproportionality between terms, the disproportionality in the atomistic model is time variant, and - at the characteristic time scale selected - may not hold for all time. Note that this condition where the assumptions are sometimes satisfied occurs within a small range of characteristic scales. Were a slightly larger time scale selected, the Case II assump-

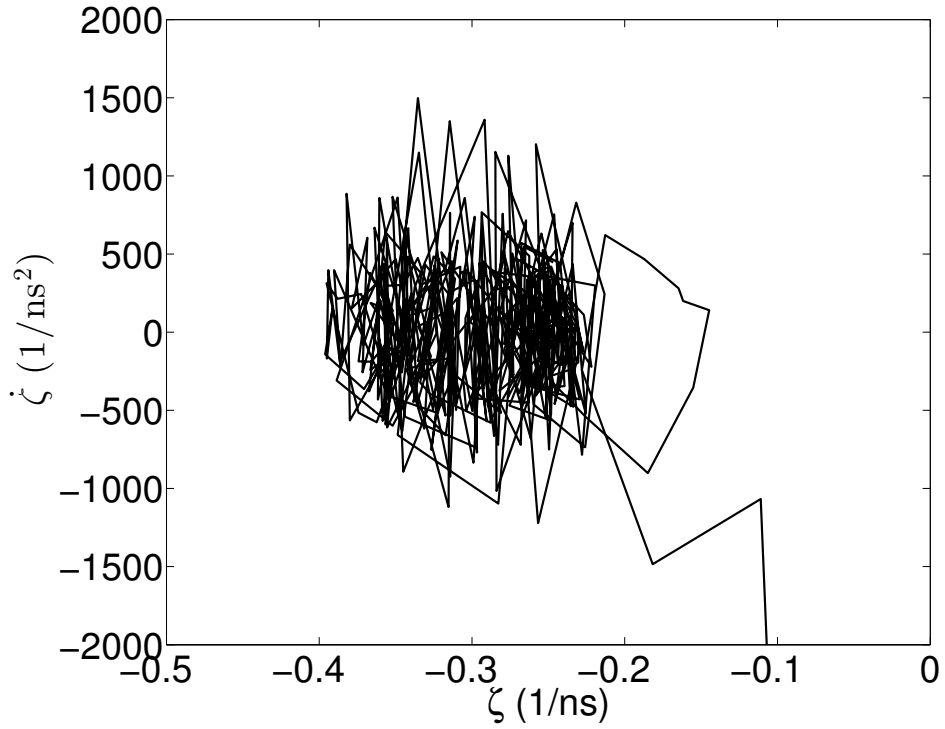

Figure 4.8. Phase plot for the unscaled modified thermostat.

tions would always hold. These assumptions would continue to hold as that scale was further increased until a scale was reached where the Case III assumptions begin to be satisfied. Also, given that the disproportionalities defined in Table 4.1 establish an approximate starting point for a_2, the ideal value of a_2 will change with time as well. This time variant disproportionality in the equations of motion is further discussed in Chapter 5.

4.9 Computation Time Reduction

The time required to perform the numerical integration for the original, unscaled, and scaled systems was approximately 65 hours, 64 hours, and 49 hours, respectively. The difference between the original and unscaled systems is not consid-

ered significant and attributed to differences in initial conditions and time evolutions. This equivalent computation time is consistent with the results of [136] for the user-specified thermostat values selected. The computation time reduction achieved by the multiscale thermostat formulation was 24%. It was observed that the scaled case integrated at time steps about a half an order larger than the other cases. The main mechanism for these savings was therefore a decrease in the total number of integration steps required to generate equivalent time evolutions.

While a 24% reduction in computation time is a significant contribution, this reduction may be further enhanced. First, it is known that increasing system complexity leads to a reduction in scaling effectiveness [77, 79, 80]. Smaller systems then might therefore achieve greater computational savings. Second, the Lennard-Jones potential used in this work establishes an upper limit on possible savings. The potential force defined by Eqns. (4.17) - (4.19) contains a Lennard-Jones power-twelve term that models a repulsive force. This repulsive term increases with decreasing distance and is responsible for bringing the relative velocity between two particles to zero to model a collision. This force will grow, even when scaled, to a magnitude necessary to elicit this change in the particles' trajectories. It was observed that this repulsive term quickly increases during collisions, forcing smaller integration steps. If these collision interactions could be resolved without these large, spiky repulsive forces, integration step sizes could be further increased providing even greater computational savings.

4.10 Conclusions

This chapter presented an extension to the continuum regime multiscale method making it applicable to atomistic bodies. The goal of this reformulation was to allow simulations consisting of atomistic bodies to be integrable at larger time steps, reduc-

ing the computation time required to generate time evolutions by reducing the total number of integration steps. This reformulation replaced the viscous friction used in the continuum derivation with a force-based temperature controlling thermostat to define a large disproportionality in the equations of motion. This disproportionality justifies the removal of a large portion of the active and thermostat forces that would otherwise cancel.

The scaled system results were compared to those of two unscaled systems. It was shown that the proposed atomistic multiscale formulation reduced the computation time required by 24%. Temperature response showed a loosening of the constant temperature constraint during the transient phase of temperature change. However, response oscillations did diminish and convergence was achieved in approximately 20 picoseconds. This 24% reduction in computation time allows for temperature controlled atomistic systems to be simulated in significantly less real time while exhibiting fast convergence to a target temperature.

This work also suggested that a time variant scaling factor may be needed to address the time variant disproportionalities in the equations of motion. Chapter 5 presents an extension of the continuum regime multiscale method to account for forces whose magnitude changes significantly over the time history.

CHAPTER 5

ADAPTIVE MULTISCALE METHOD

5.1 Introduction

This chapter presents a second extension to the established multiscale method that accounts for forces whose magnitude changes significantly across the time history. Recall that a key component in the derivation of this multiscale method is the disproportionality between the viscous damping and mass terms. This disproportionality is used to identify a portion of the active and damping forces that cancel which can be eliminated from the dynamics without significantly affecting the long-term results. In previous works, the magnitude of the active forces - and thus the portion of forces eliminated - was assumed to be a constant. However, potential functions often decay exponentially with increasing distance leading to resultant force magnitudes that can vary greatly across the time history [137, 40, 41, 42, 43, 138].

It is well established that small forces can have a significant effect on larger scale phenomena over relatively long time frames [139, 140, 141], so it is critical that small contributing forces not be scaled along with the large forces. As the scaling process reduces the magnitude of the scaled forces by several orders, if this process is applied to a small force that is not canceled by other forces, any effect that force has on the time evolution could be greatly diminished. In order for scaled system results to match those of the unscaled system, it is therefore necessary that small potential forces be allowed to affect the system dynamics without inappropriate scaling.

In this chapter, it is assumed that some of the active forces, such as those produced by conformational springs and Brownian motion, result in vibratory motion

and do not contribute significantly to the final states. Meanwhile, other forces such as those produced by long-range potential interactions, act constantly over a long time frame to elicit more significant effects on final states. An adaptive scaling approach is therefore proposed to remove portions of the large, non-contributing active and dissipative forces while retaining the small, contributing potential forces. When these potential forces grow to a significant enough magnitude, they are similarly scaled along with the large active and dissipative forces.

The specific model selected for testing the efficacy of this proposed adaptive scaling method is a 400 nm nanoparticle conjugated with Glycoprotein 1b (GP1b) proteins interacting with von Willebrand factor (VWF) proteins. This system is similar to that seen in [23]. The GP1b protein is a component of the greater GP1b-IX-V complex found on the surface of thrombocytes, or platelets. This protein is well researched and known to bind to VWF proteins expressed on damaged sub-endothelial arterial cells [142, 143, 144, 145, 146]. These ligand-receptor interactions are currently showing great promise as a mechanism for targeted delivery and tissue regeneration [23, 147, 148].

5.2 Adaptive Scaling

Recall that in Chapter 2, the multiscale formulation was presented for a rigid multibody model satisfying the continuum assumption such that viscous damping was applied to the bodies. After defining a small parameter ϵ and using that small parameter to expand the time derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ into series, a new equation of motion was given by Eqn. (2.10) which is repeated here:

$$\mathbf{0} \;=\; \epsilon^0 \left(D(\mathbf{q}) \, \frac{\partial \mathbf{q}}{\partial T_0} \;-\; \frac{\mathbf{\Gamma}_{others}}{\beta} \right) \;+\; \epsilon^1 \Big(\overline{M}(\mathbf{q}) \frac{\partial^2 \mathbf{q}}{\partial T_0^2} \;+\; \overline{C}_{00} \;+\; \cdots \Big) \;+\; \cdots \tag{5.1}$$

In Chapter 2 it was assumed that a significant portion of the forces in the first term on the right side canceled, justifying the scaling of these forces. It was also assumed that the forces were significantly large compared to the acceleration term. These assumptions led to the scaled equation of motion given by Eqn. (2.14) which is repeated here:

$$M(\mathbf{q})\ \ddot{\mathbf{q}}\ +\ C(\mathbf{q},\dot{\mathbf{q}})\ +\ a_2\ \beta\ D(\mathbf{q})\ \dot{\mathbf{q}}\ =\ a_2\ \mathbf{\Gamma}_{others} \tag{5.2}$$

Consider now the case where the potential forces are several orders of magnitude smaller than the other forces, a disproportionality similar in scale to the small parameter ϵ. In this case, the $\mathbf{\Gamma}_p$ term contributes little to Eqn. (5.1) equaling zero and does not cancel any significant portion of the other forces. Scaling this potential term is therefore not justified by the multiscale analysis. Rearranging Eqn. (5.1) to reflect this consideration yields:

$$\mathbf{0}\ =\ \epsilon^0\ \Big(D(\mathbf{q})\ \frac{\partial \mathbf{q}}{\partial T_0}\ -\ \frac{\mathbf{\Gamma}_{others}}{\beta}\Big)+\ \epsilon^1\Big(\frac{\mathbf{\Gamma}_p}{\epsilon^1\beta}\ +\ \overline{M}(\mathbf{q})\frac{\partial^2 \mathbf{q}}{\partial T_0^2}\ +\ \overline{C}_{00}\ +\ \cdots\Big)\ +\ \cdots \tag{5.3}$$

If it is assumed that the accelerations are not significantly large, it can be seen that the first term on the right side of Eqn. (5.3) must largely cancel such that the whole equation is equal to zero. The assumption that the accelerations are not significantly large can then be imposed by scaling the forces contained within that first term. Similarly to the procedure in Chapter 2, this is accomplished by splitting these forces into small and large parts:

$$\mathbf{0}\ =\ (a_1\ +\ a_2)\ \Big(D(\mathbf{q})\ \frac{\partial \mathbf{q}}{\partial T_0}\ -\ \frac{\mathbf{\Gamma}_{others}}{\beta}\Big)+\ \epsilon^1\Big(\frac{\mathbf{\Gamma}_p}{\epsilon^1\beta}\ +\ \overline{M}(\mathbf{q})\frac{\partial^2 \mathbf{q}}{\partial T_0^2}\ +\ \overline{C}_{00}\ +\ \cdots\Big)\ +\ \cdots \tag{5.4}$$

where it is assumed that the forces associated with the a_1 term cancel. Eliminating the forces associated with the a_1 term leads to a new scaled equation of motion:

$$M(\mathbf{q})\ \ddot{\mathbf{q}}\ +\ C(\mathbf{q},\dot{\mathbf{q}})\ +\ a_2\beta\ D(\mathbf{q})\ \dot{\mathbf{q}}\ =\ a_2\mathbf{\Gamma}_{others}\ +\ \mathbf{\Gamma}_p \tag{5.5}$$

For potential forces whose magnitude changes significantly over the time history, both Eqns. (5.2) and (5.5) may be justified at various points. These two cases can be simultaneously represented by the adaptively scaled equation of motion:

$$M(\mathbf{q})\ \ddot{\mathbf{q}}\ +\ C(\mathbf{q},\dot{\mathbf{q}})\ +\ a_2\ \beta\ D(\mathbf{q})\ \dot{\mathbf{q}}\ =\ a_2\ \mathbf{\Gamma}_{others}\ +\ a_2^*\ \mathbf{\Gamma}_p \tag{5.6}$$

where the a_2^* term denotes a time-variant scaling factor. In this chapter, it is only considered whether scaling the potential forces *is* or *is not* justified, leading to an adaptive scaling scheme similar to a bang-bang controller:

$$\begin{gathered} if\ \frac{\mathbf{\Gamma}_p}{\beta} > \alpha\ \longrightarrow a_2^*\ =\ a_2 \\ else\ a_2^*\ =\ 1 \end{gathered} \tag{5.7}$$

where α is a user-specified threshold. Note that $a_2^* = 1$ implies that the potential forces are not scaled. In this work, this threshold is identified with respect to the magnitudes of the other forces as measured in short-time unscaled cases.

5.3 Model Description and Equation of Motion

A rigid multibody model connected by kinematic joints was selected to test the proposed method. This model represents a coarse grained approximation of the GP1b-conjugated nanoparticle discussed above. The sequence and structure of the GP1b protein are well documented [149]. This protein is modeled as four rigid bodies connected by spherical joints. Figure 5.1 shows the mechanical model of this system.

This model consists of a floating base with numerous kinematic chains attached. Each kinematic joint is spherical allowing three degrees of rotation. The equations of motion for this system are the set of adaptively scaled equations given by Eqn. (5.6). The forward dynamics are solved using a recursive numerical approach similar to [26] in which quantities are defined using 6-dimensional spatial notation. Mass

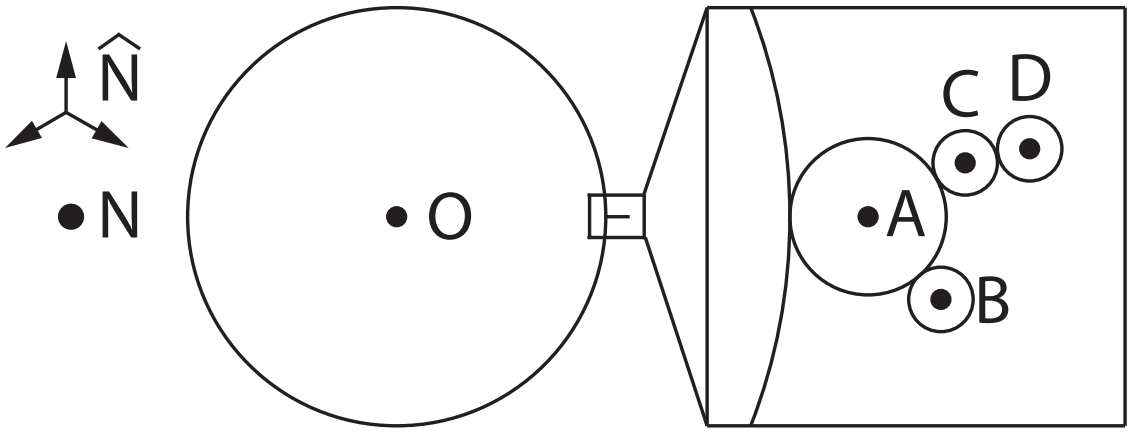

Figure 5.1. Model of the conjugated nanoparticle. Nanoparticle center of mass at point O. Bodies A, B, C, and D are the four sub-bodies of the coarse grained GP1b protein.

matrices and vectors of known quantities are modified through back substitution to reflect the effective inertia and forces experienced at each joint. This back substitution eliminates the unknown reaction forces, allowing for accelerations to be calculated for each joint in a succinct, computationally efficient manner. Appendix A provides a thorough derivation and discussion of this approach.

For each spherical joint, Euler parameters are used to define the joint rotation such that the singularities associated with other methods are avoided. When solving for the accelerations at a particular spherical joint, the equation of motion, which nominally has the dimensions 3×1, is transformed into the 4×1 vector space corresponding to the Euler parameters. Appendix B provides a discussion of Euler parameters, including how they can be combined with the recursive equations of motion.

Once an equation of motion is transformed into the 4×1 Euler parameter vector space, the 4×4 mass matrix is singular and non-invertible. These equations must then be reduced back the minimal form such that the mass matrix is invertible. While

many methods exist for accomplishing this reduction, an online constraint embedding technique [150] was used to enforce the Euler parameter normality constraints:

$$\mathbf{e}^T\mathbf{e} \;=\; 1 \tag{5.8}$$

$$\dot{\mathbf{e}}^T\dot{\mathbf{e}} \;=\; 0 \tag{5.9}$$

where $\mathbf{e}$ denotes the 4×1 set of Euler parameters and $\dot{\mathbf{e}}$ is the time derivatives. This technique reduces the size of the equation of motion by eliminating the degree of freedom associated with a specified dependent Euler parameter. The reduced mass matrix is inverted and used to solve for the three independent accelerations. In this manner, the constraints given by Eqns. (5.8) and (5.9) are defined implicitly through numerical integration. Appendix C provides a discussion of this online constraint embedding technique.

The ideal dependent Euler parameter is best identified by mass matrix condition number to ensure minimal error during inversion. As the condition number is a function of the parameters and the parameters are time variant, the specified dependent parameter must also change throughout the time history. However, each time this variable changes, it results in some error due to renormalization of the Euler parameters and speeds. An efficient approach for selecting the dependent variable while minimizing error is presented in [151].

5.4 Forces and Energy

Four forces are implemented in the model:

$$\mathbf{\Gamma} \;=\; \mathbf{\Gamma}_p \;+\; \mathbf{\Gamma}_d \;+\; \mathbf{\Gamma}_k \;+\; \mathbf{\Gamma}_b \tag{5.10}$$

where $\mathbf{\Gamma}_p$ denotes potential forces arising from potential interactions, $\mathbf{\Gamma}_d$ is the damping forces modeling friction, $\mathbf{\Gamma}_k$ is the conformational forces arising from springs, and

$\mathbf{\Gamma}_b$ are the stochastic Brownian motion forces modeling thermal noise and random fluctuations in the medium.

As mentioned in previous chapters, the driving forces in micro and nanoscale dynamical systems arises from inter-atomic potential interactions and are often modeled as a combination of Coulomb and Lennard-Jones terms. These forces are found by taking the derivative of the potential energy expressions:

$$\Phi_{ij}(r_{ij}) \;=\; \frac{A}{r_{ij}^{12}} \;-\; \frac{B}{r_{ij}^{6}} \;+\; \frac{C}{r_{ij}} \tag{5.11}$$

$$\Gamma_p(r) \;=\; \frac{d\Phi}{dr}\bar{r} \tag{5.12}$$

In this work, explicit atoms are not modeled and an approximate potential function was generated for each of the coarse grained bodies:

$$\Phi_{ij}(r_{ij}) \;=\; \frac{A^*}{r_{ij}^{12}} \;-\; \frac{B^*}{r_{ij}^{6}} \;+\; \frac{C^*}{r_{ij}} \tag{5.13}$$

where A^*, B^*, and C^* are the modified Lennard-Jones and Coulomb terms, respectively. These approximate terms model the potential for all of the atoms within the body. The resultant potential forces act through the bodies' centers of mass. As this work is not a docking study, it was assumed that these approximations in potential would not affect the testing of the proposed adaptive multiscale formulation's efficacy.

Numerous potential functions exist for the various levels of coarse grained biopolymers, but all decay at long range [137, 40, 41, 85, 42, 43, 86, 138]. Because of this decay, it is possible that the magnitude of the resultant potential force could change significantly across the time history. Considering this, it is expected that another potential expression in place of Eqn. (5.13) would also be compatible with the proposed formulation.

All bodies within the model are large enough to satisfy the continuum assumption. These bodies are assumed to be spherical such that Stokes law may be used to

calculate the β drag terms. Coarse grained biopolymers approximate secondary and tertiary structures - alpha helices and beta sheets for example - and are often not spherical. Any drag model may be selected as appropriate for the modeled system and will still satisfy the continuum multiscale method as long as that damping term is significantly large enough compared to the mass term. This damping term is used to define the forces arising from friction:

$$\mathbf{\Gamma}_d \;=\; -\beta D(\mathbf{q})\dot{\mathbf{q}} \tag{5.14}$$

Torsional springs are considered for each of the kinematic joints to model the bending stiffness of the biopolymer. The values of these springs depend on the specific geometry and chemical bonding at that joint. The forces arising from these torsional springs comprise $\mathbf{\Gamma}_k$.

Brownian motion is included to model stochastic forces generated through random motion and thermal noise in the medium. The method used to implement Brownian motion is similar to that used in [76]. The random force acting on a particular body may be defined as:

$$\mathbf{\Gamma}_b \;=\; B_1(t)\widehat{\mathbf{N}}_1 \;+\; B_2(t)\widehat{\mathbf{N}}_2 \;+\; B_3(t)\widehat{\mathbf{N}}_3 \tag{5.15}$$

where the B_i term represents the stochastic force. Each component of the random forces is treated independently as a uniformly distributed random variable. They have the following expectations $E[\cdot]$, or weighted average values:

$$E[\; B_i(t)\;] \;=\; \langle\; B_i(t)\;\rangle \;=\; 0 \tag{5.16}$$

and are governed by a fluctuation-dissipation relation:

$$E[\; B_i(t_1)\; B_j(t_2)\;] \;=\; 2\;\beta\;k_B T\;\delta(t_1 - t_2)\delta_{ij} \tag{5.17}$$

where k_B denotes the Boltzmann constant and T is the absolute system temperature. The collection of these random forces comprise $\mathbf{\Gamma}_b$. The randomly generated variables are updated after a specified period of integration steps and held constant for each period.

To ensure energy conservation and dynamic consistency, the kinetic and potential energies are often totaled with the work done by dissipative and stochastic forces to show a constant across the time evolution. However, an interesting result in [152] was a shift in total system energy produced by scaling potential forces. As potential energy is the anti-derivative of potential forces, any scaling of the forces must also scale the energy. This was not a problem in [152] as the scaling was a constant. However, this scheme cannot be used in this work as adaptive scaling of the potential forces would result in discontinuities in the potential energy calculated.

As an alternative confirmation of dynamic consistency, the work done by all the forces can be calculated and compared with the kinetic energy to show a constant across the time evolution:

$$T \; - \; W_d \; - \; W_k \; - \; W_p \; - \; W_b \; = \; constant \tag{5.18}$$

where T denotes the kinetic energy and W is work. The work is calculated through numerical integration of the total instantaneous change in work $\dot{W}$, or the power produced by the forces. The work done by the damping force for example is therefore:

$$W_d \; = \; \int_{t_1}^{t_2} \sum_{i=1}^{bodies} \mathbf{f}_{d,i} \cdot \boldsymbol{\vartheta}_i \tag{5.19}$$

where $\mathbf{f}$ and $\boldsymbol{\vartheta}$ denote the spatial definitions of force and velocity, respectively, i iterates over all bodies, and the $[\cdot]$ operator is the dot product. Appendix A further discusses the spatial quantities used in this investigation.

5.5 Hardware and Simulation Parameters

A generalized forward dynamics toolbox was constructed to simulate the conjugated nanoparticle. This toolbox was written first in MATLAB and then in the C++ programming language using the CUDA library [24] for parallel processing on NVIDIA GPUs. The MATLAB toolbox performs the calculations serially on the CPU and was used in development to confirm the accuracy of the overall process. The C++ toolbox was used to generate the simulation data presented in this chapter. A model description defined in preprocessing specified the number of bodies, joint kinematics, and mass and inertial properties. This model description was then passed to the derivative file along with a vector of current states. A model-specific subprocess returned vectors of forces for each body. Using the provided model, forces, and states, the generalized accelerations were calculated using the recursive algorithm, Euler parameters for spherical joints, and online constraint embedding processes mentioned above. This forward dynamics process is described in fine detail in the appendices.

The forward dynamics and numerical integration were solved wholly on the GPU. Because of the independence of calculations, the point-to-point potential interactions between all bodies were resolved concurrently. Kinematic calculations along each protein chain were also performed concurrently. All linear algebra calculations were performed using an optimum number of blocks and threads to maximize GPU bandwidth. Because parallel processing was used, this otherwise burdensome computational process was resolved in significantly less real time. The production of this dynamics toolbox was a significant development and can be reused in later studies.

Numerical integration was performed on an HP Z230 Workstation with an Intel Xeon E3-1225 v3 processor with 12 GB of RAM running Ubuntu 16.04 operating system. Parallelization was performed on an NVIDIA Quadro P1000 4GB GPU. The integration was performed by a Kutta-Merson algorithm [88] with adaptive integration

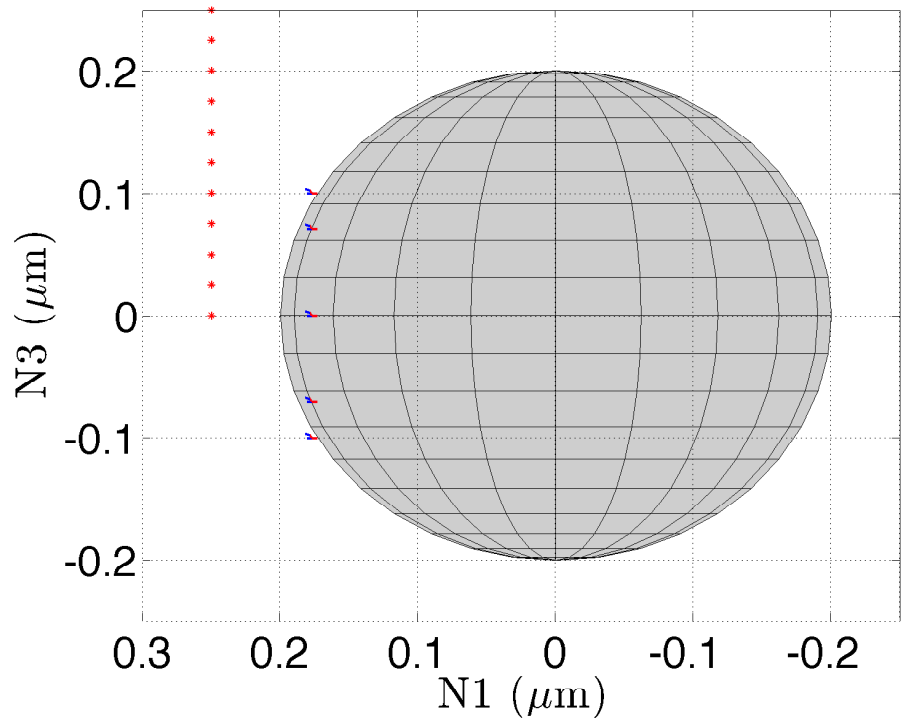

Figure 5.2. Initial positions.

step sizes for reducing the numerical error. Relative and absolute errors were 10^{-5} and 10^{-6}, respectively.

Eight GP1b proteins were attached to the nanoparticle surface in a circular arrangement. While exact conjugation densities of surface ligands are difficult to calculate, this density of ligands is an extrapolation of the methodology in [23]. Twenty VWF proteins were modeled as fixed points on the $(\widehat{\mathbf{N}}_2, \widehat{\mathbf{N}}_3)$ plane. Potential interactions were calculated between these VWF proteins and all protein bodies.

The mass and inertial properties of the nanoparticle were also extrapolated from [23]. The mass of the proteins were calculated from totaling the mass of all the residues in the primary sequence of each coarse grained body. The inertia properties were calculating using the determined mass and a size estimated from x-ray crystallography

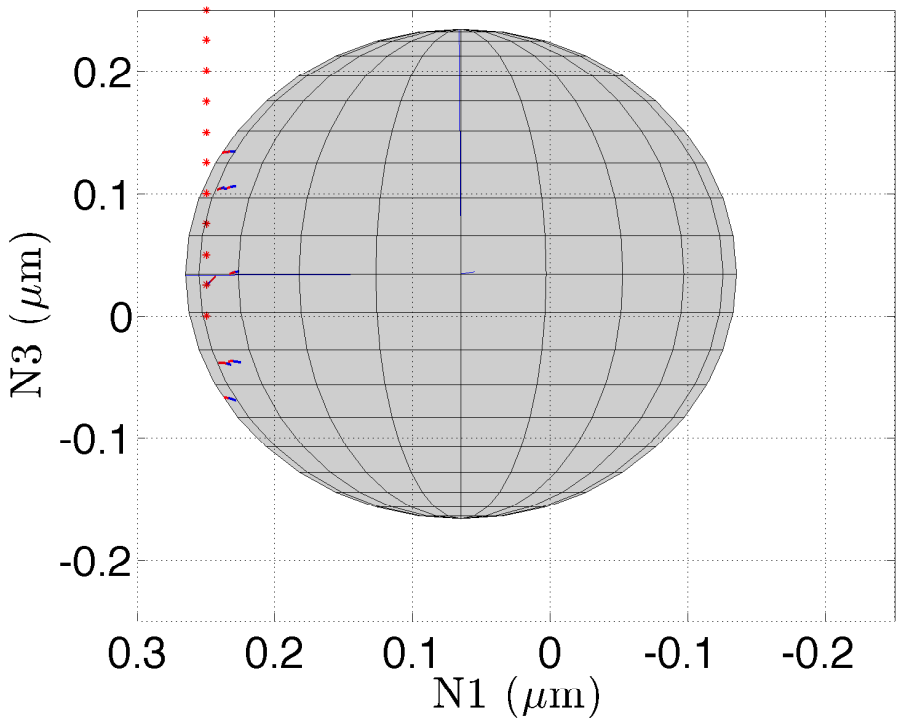

Figure 5.3. Final positions of the adaptively scaled simulation.

data from an online research aggregate database [149]. A unit system of (fg, μm, μs) was selected. This unit system keeps the mass of the nanoparticle in the 10^0 - 10^1 range. The time and distance scales were selected for convenient observation. Tables D.3 and D.4 in Appendix D list the values of the parameters used.

For the conformational spring constants, the specific structure at each kinematic joint was considered and approximated from available sources. For example, bodies C and D together model a helical structure. The joint between them models the bending of this helix by allowing some relative motion but the that torsional spring is significantly stiffer than the others. Body B meanwhile approximates a chain of residues bound with di-sulfide bonds, so a relatively looser spring was used.

Initial nanoparticle position was 250 nm from the $(\widehat{\mathbf{N}}_2, \widehat{\mathbf{N}}_3)$ plane containing the VWF proteins. The nanoparticle has a radius of 200 nm. Because the GP1b proteins are attached in a circular arrangement, the nanoparticle must traverse approximately 65 nm for the protein to closely interact with the VWF receptors. Initial nanoparticle orientation is such that the GP1b proteins are directed towards the VWF receptors in the $(\widehat{\mathbf{N}}_2, \widehat{\mathbf{N}}_3)$ plane. The initial positions can be seen in Fig. 5.2. Final positions produced by the adaptively scaled system can be seen in Fig. 5.3. A flow velocity was defined as 1 mm/s in the $\widehat{\mathbf{N}}_3$ direction.

5.6 Results Analysis

Three simulations were performed on unscaled, constantly scaled, and adaptively scaled systems. The constantly scaled system was simulated to show the need for the adaptive approach. Figure 5.4 shows plots of the q_1 coordinate over time for the three cases. This coordinate tracks the motion of the nanoparticle in the $\widehat{\mathbf{N}}_1$ direction. The adaptively scaled case is given by the solid line, the constantly scaled case is given by the dashed line, and the unscaled case is given by the dotted line. Since the VWF receptor proteins are in the positive $\widehat{\mathbf{N}}_1$ direction relative to the nanoparticle, the q_1 coordinate will increase as the nanoparticle approaches the VWF receptors. It can be seen that in constantly scaled case, the scaling of the small potential forces reduces the work done by those forces and no significant displacement of the nanoparticle is observed. It can be seen that for both the scaled and unscaled cases, the nanoparticle moves toward the VWF receptors and eventually docks. While the two trajectories are similar, there is some noticeable variance. It can be seen that the unscaled case requires more time to reach an equivalent position. The standard deviation between these two trajectories is approximately 5 nm.

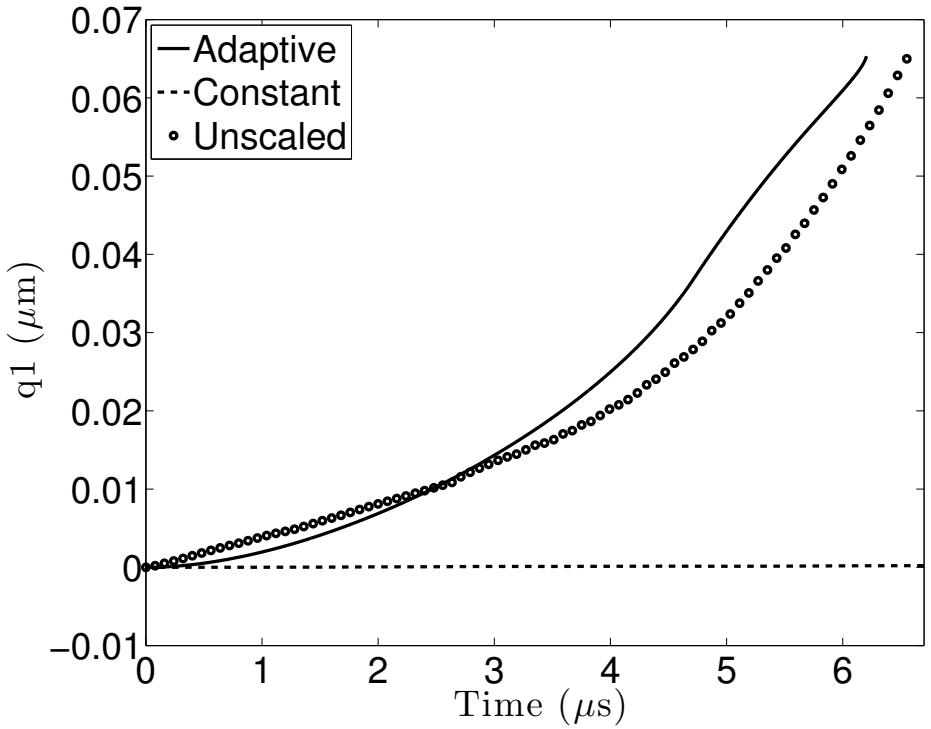

Figure 5.4. Nanoparticle position in $\widehat{\mathbf{N}}_1$ direction for the adaptively scaled, constantly scaled, and unscaled cases.

Figures 5.5 and 5.6 show the system energies from the adaptively scaled and unscaled cases, respectively. The quantities plotted include the kinetic energy and the work done by the various forces. Recall that since an adaptive scaling method is applied to the potential forces, the potential energy cannot be used to confirm energy conservation. Scaling of the potential forces yields a discontinuous potential energy, so the work done is calculated instead.

The energy quantities are separated into two plots to conveniently show the time evolutions of the large and small quantities. Since the works done by stochastic and damping forces are significantly larger than the other quantities, it is impossible to observe the trajectories of the small quantities when all are plotted together. The work done by potential forces is included in the plots of large quantities for comparison.

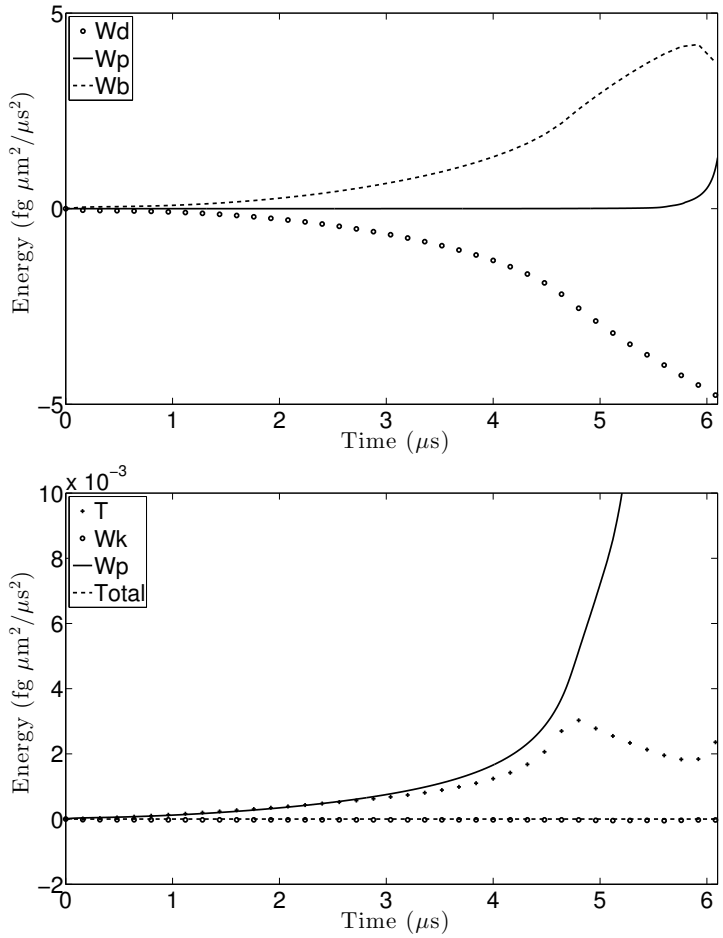

Figure 5.5. Adaptively scaled system energy. W denotes the work done by friction (subscript d), potential (p), Brownian (b), and conformational (k) forces. T denotes kinetic energy.

These plots confirm that Eqn. (5.18) can be used to check for energetic consistency, even when the adaptive scaling method is implemented.

Figure 5.7 shows a plot of the forces over time in the adaptively scaled system. The quantities plotted are the total magnitude of each of these forces found through summation across all bodies. The damping forces are given by the dashed line, the conformational forces are the small dots, the potential forces are the solid line, and the Brownian motion forces are the open dots. It can be seen that the damping forces

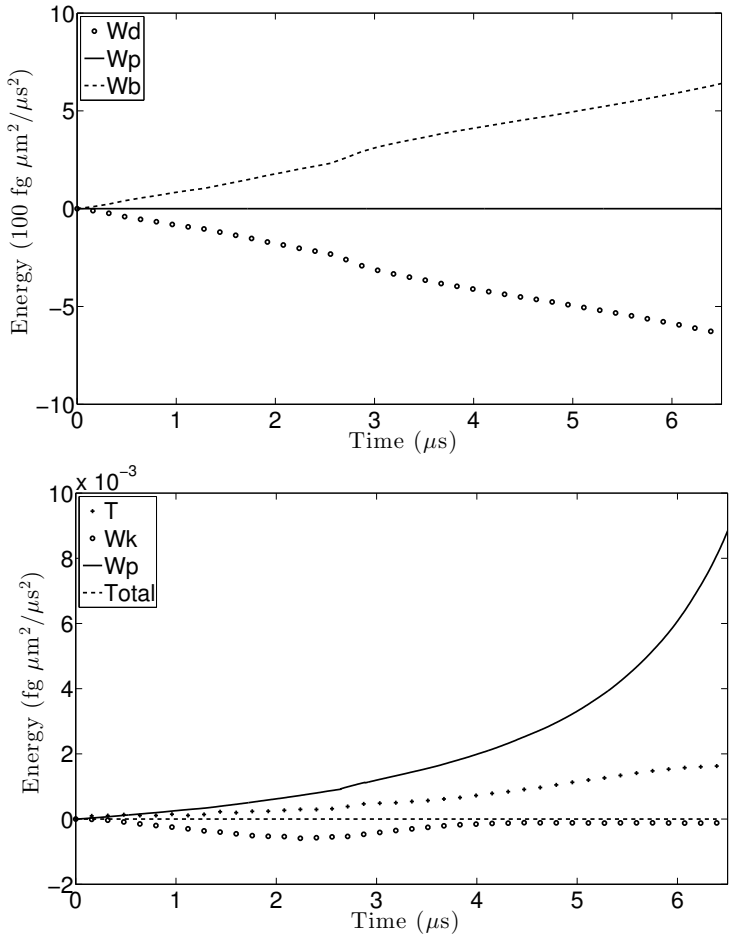

Figure 5.6. Unscaled system energy. W denotes the work done by friction (subscript d), potential (p), Brownian (b), and conformational (k) forces. T denotes kinetic energy.

are relatively large at initial conditions and increase with nanoparticle velocity. It can also be seen that the potential forces increase exponentially with decreasing distance between the nanoparticle and the VWF receptors. Note that magnitude of the stochastic forces is a constant. Recall from the discussion above on the implementation of the stochastic forces that sets of uniformly distributed variables are generated and normalized to specify the unit vectors for forces of constant magnitude. The dashed vertical line is included to show the time where the adaptive scaling term a_2^*

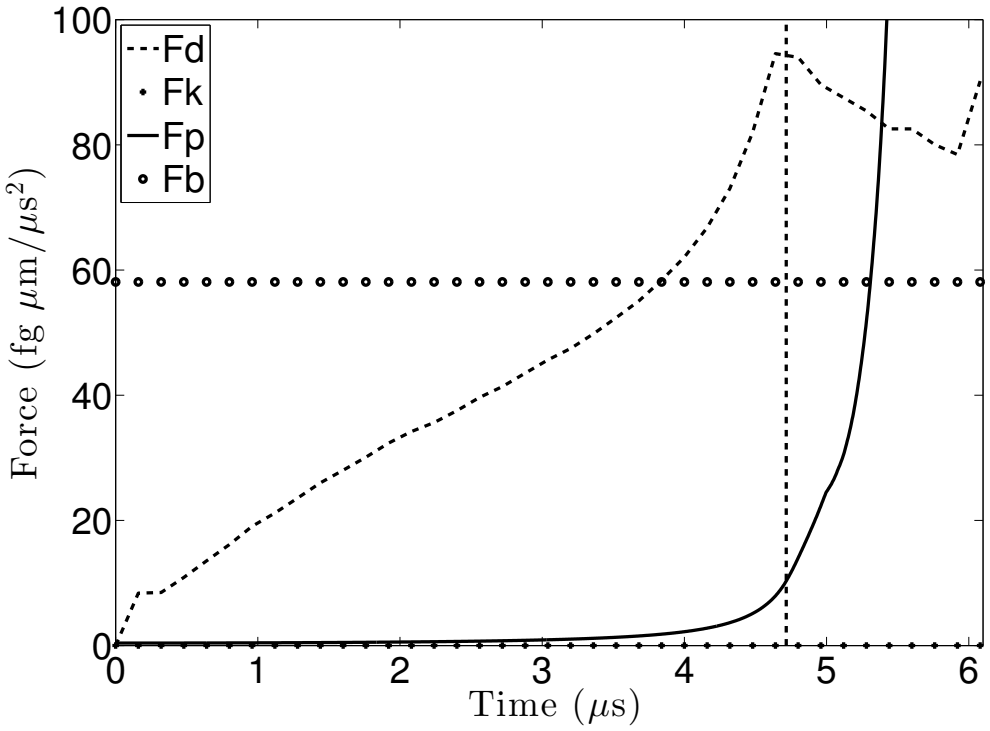

Figure 5.7. Force magnitudes from the scaled system. Forces shown are damping (subscript d), conformational (k), potential (p), and Brownian (b). Dashed vertical line denotes time when adaptive scaling is activated.

is activated by the control law given by Eqn. (5.7). Recall that this adaptive scaling is activated once the value of F_p exceeds a user-specified threshold. There is a noticeable change in the damping force F_d following this change. This response will be addressed later and correlated to other data.

Figure 5.8 shows a plot of the adaptive scaling factor a_2^* over time. It can be seen that the adaptive scaling is activated at approximately 4.7 μs. This time corresponds to the vertical dashed line seen in Fig. 5.7. While the the adaptive scaling algorithm allows for the scaling to deactivate, the potential forces remained large throughout the remainder of the simulation.

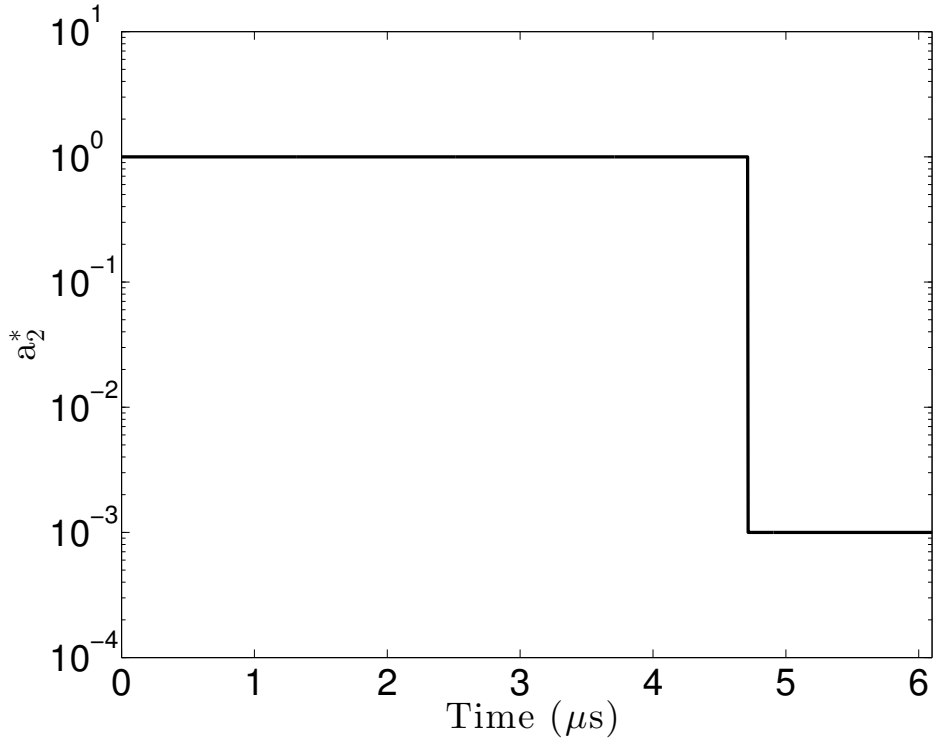

Figure 5.8. Adaptive scaling factor a_2^* over time.

Figures 5.9 and 5.10 are including to further observe the effects of system scaling. Figure 5.9 shows plots of the kinetic energy for the adaptively scaled and unscaled cases. The adaptively scaled case is given by the solid line and the unscaled case is given by the dotted line. The dashed vertical line denotes the time when the adaptive scaling is activated. As in Fig. 5.7, a noticeable change occurs once the adaptive scaling has activated. The scaling of the potential forces brings it into the same scaled order of the damping and stochastic forces. This means that there is a discontinuity in the magnitude of the potential forces experienced by the system. It can be seen that prior to the scaling activating, the kinetic energy was swiftly increasing, corresponding to an increase in nanoparticle velocity. By diminishing the potential forces, the damping forces dissipate some kinetic energy, slowing the nanoparticle. This can

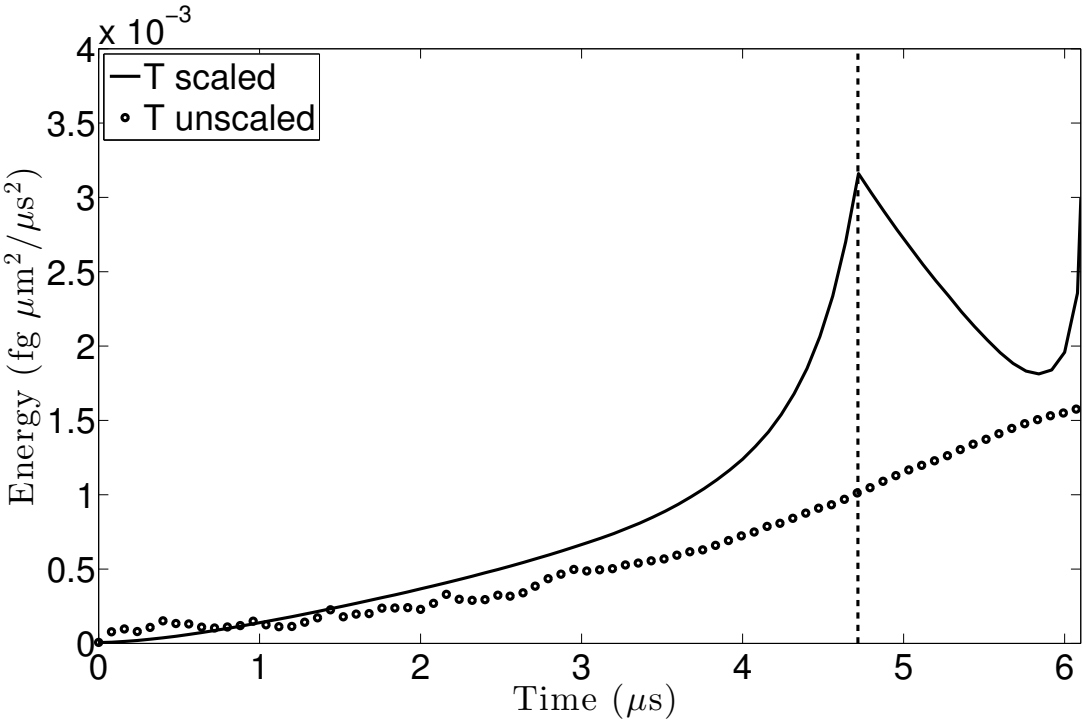

Figure 5.9. Comparison of kinetic energy from the adaptively scaled and unscaled simulations. Solid line is the adaptively scaled case and the dashed line is the unscaled case. Dashed vertical line denotes time when adaptive scaling is activated.

be seen by the reduction in kinetic energy immediately following the activation of the scaling. Even when scaled, the potential forces still do work on the nanoparticle bringing it closer to the VWF proteins. It can be seen that after the initial reduction in kinetic energy, the nanoparticle accelerates again once the potential forces have grown. By comparison, the kinetic energy of the unscaled system experiences a relatively smooth trajectory. While some variance exists between these two data sets, especially near the point of scaling activation, the trajectories and final values are similar. The dip in kinetic energy after scaling activation can be considered then a correction toward a more accurate estimation of the time evolution.

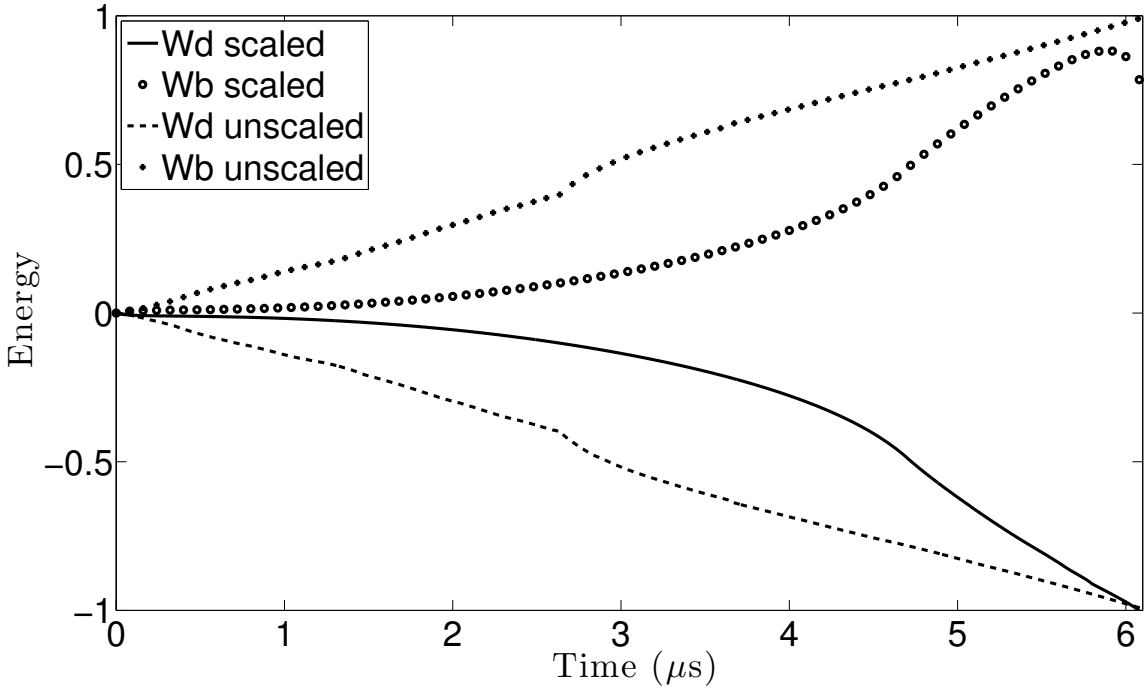

Figure 5.10. Comparison of work done by damping (subscript d) and Brownian (b) forces from the adaptively scaled and unscaled simulations.

Figure 5.10 shows plots of the work done by damping and Brownian forces for the scaled and unscaled systems. The work done by damping are denoted by the solid and dashed lines for the scaled and unscaled cases, respectively, and the work done by Brownian forces are denoted by the open and closed dots for the scaled and unscaled cases, respectively. For Fig. 5.10, these quantities have been normalized and scaled to show the similarity in time histories. These quantities do however have significantly different actual values, as can be seen in Figs. 5.5 and 5.6. This large difference arises from the scaling of the damping and Brownian forces. Since the forces are diminished in magnitude, they do less work on the system. A key outcome of applying the multiscale analysis is a reduction in forces, and thus a reduction in large

resultant accelerations. The difference between the two profiles then is a visualization of the work eliminated by the scaling procedure.

5.7 Computation Time Reduction

The time required to perform the numerical integration for the scaled and unscaled systems was approximately 837 minutes and 32310 minutes, respectively. The computation time reduction achieved by the adaptive multiscale formulation was approximately 97.4%. This computation time reduction is consistent with the results seen in [76]. It was observed that the scaled system would integrate with step sizes approximately in the 100 - 50 ps range. The unscaled system would integrate with step sizes approximately in the 1 - 0.5 ps range. The main mechanism for the computational savings was therefore a decrease in the total number of integration steps required to generate the time evolutions.

5.8 Conclusions

This chapter presented a second extension to the continuum multiscale formulation to accounted for active forces whose magnitude changes significantly over the duration of the simulation. By not scaling the potential forces initially, when the magnitudes were small, the forces were allowed to contribute to the time evolution through consistent application of small forces. Once these forces grew to a significant enough magnitude, they were scaled in a manner consistent with the other forces. This formulation resulted in a significant reduction in computation time by allowing for integration step sizes approximately two orders of magnitude larger to be implemented.

The adaptive control scheme used was similar to a bang-bang controller. This method, though effective, is a relatively simply implementation. The work detailed here can be further expanded to allow the value of a_2^* to vary across a range of values depending on the current magnitude of forces. This further work may produce adaptive scaling schemes that reduce the error in overall time history by better accounting for the changes in force cancelation.

5.9 Future Work

This dissertation presented three developments in the field of micro and nanoscale dynamical simulation. Each of these developments have succeeded in their respective goals, but more work can always been done in applying these developments to new systems and the refining of methods. The modified Nóse-Hoover thermostat has only been applied to the models of molten nitrate salts shown in Chapters 3 and 4. Future works can apply this thermostat and the atomistic multiscale analysis to other systems, possibly other phases of matter, to investigate the effects on calculated thermodynamic quantities.

The atomistic multiscale formulation, though effective, has much room for improvement. The current result of 24% computation time reduction is limited by the repulsive Lennard-Jones term used to model collisions between particles. This nonlinear force term grows, even when scaled, to a magnitude large enough to elicit a change in relative velocity between two particles. When this force is scaled, the distance between the two particles reduces, leading to larger forces. The accelerations resultant from these collisions is approximately the same for both the scaled and unscaled case, leading to similarly small integration step sizes during those collision events. A future work is investigating how those collisions can be resolved by replacing the repulsive terms with an analytical contact-impact analysis [153].

The adaptive scaling results shown in this chapter are similar to those of the unscaled case, but variances in trajectories can be observed. It is also seen in the plots of kinetic energy and forces that the activation of the scaling does result in a correction period in which the nanoparticle first loses energy, then regains it. It is possible then that the specified scaling threshold is too high and should activate earlier to prevent this abrupt correction period. Future works will also investigate the application of a more complex adaptive scaling algorithm, such as an initial activation of the scaling factor and then further changes of the scaling factor along a continuous range of values. Continuous control laws using hyperbolic functions could also be used. While future works will continue to expand these methods, the investigations detailed in this dissertation show that each of these new methods are useful in the development of micro and nanoscale dynamical simulations.

APPENDIX A

RECURSIVE FORWARD DYNAMICS ALGORITHM

A.1 Recursive Formulation Introduction

This appendix presents the derivation of a recursive formulation of the Newton-Euler equations of motion. As an alternative to solving the equations of motion in symbolic form using linear algebra and the inversion of a mass matrix, recursive approaches are often numerical, avoiding symbolic solutions altogether. This recursive formulation uses back substitution to eliminate the unknown reaction forces until a point is reached when a set of generalized accelerations for a single body can be calculated. This set of known accelerations is then propagated along the kinematic link and successive accelerations can thus be calculated.

To see this concept visually, consider the triangularization of a dense matrix:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \longrightarrow \begin{bmatrix} \alpha_{11} & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} \tag{A.1}$$

where $A_{ij} \neq \alpha_{ij}$. Note that the specific values of the entries in A changed, but the indices and axes did not. Similarly, the inverse of the two matrices are identical. If A were included in a linear set of non-homogeneous differential equations:

$$A\ddot{\mathbf{q}} = \mathbf{b} \tag{A.2}$$

where $\ddot{\mathbf{q}}$ and $\mathbf{b} \in \mathbb{R}^{3x1}$, either of the forms given in Eqn. (A.1) would satisfy the equation for the same $\ddot{\mathbf{q}}$ and $\mathbf{b}$. The triangularized form of A implies that there is some way to restructure the matrix A such that the first unknown term in $\ddot{\mathbf{q}}$ is independent of the others. This term could be calculated by expanding Eqn. (A.2) into a series of scalar equations and isolating it. This first term could then be used to solve for the second, which could then be used to solve for the third. Similarly for the recursive forward dynamics algorithm, the inertias and forces are modified

through back substitution until the generalized accelerations of the first body can be calculated independent of the other joints.

The derivation presented here was used in the study detailed in Ch. 5. This derivation is intended as an intuitive and easy-to-follow introduction to the concept of recursive algorithms. This recursive algorithm is by no means unique. Similar formulations may be seen in [154, 155, 156, 157, 158, 26, 27, 159, 160], among many others. Similar recursive formulations have also been developed for the inverse and hybrid dynamics problems. These algorithms allows for joints with multiple degrees of freedom - and thus multiple generalized accelerations - to be resolved simultaneously, even when a rotation is combined with a translation. However, this derivation is simplified to account only for rotations. The assumptions associated with this simplification are highlighted in the derivation and the resulting algorithm can be easily modified for the general case.

This derivation makes use of spatial algebra which is the collection of translational and rotational equations into a single set of expressions. For example, a general spatial force vector would contain both forces and moments:

$$\mathbf{f} = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \tag{A.3}$$

This arrangement allows for the interdependence of the translational and rotational variables to be easily accommodated. By using this notation, a single set of equations of motion is defined for each body. Note that in this derivation, the spatial reaction and spatial external forces are denoted by $\mathbf{r}$ and $\mathbf{g}$, respectively.

Also used in this derivation is the representation of the vector cross product operation as matrix multiplication using skew symmetric matrices. This representa-

tion is useful in constructing transformation matrices which for rotational joints are constant and can be defined in preprocessing. Consider two general vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{a} = a_1\hat{\mathbf{i}} + a_2\hat{\mathbf{j}} + a_3\hat{\mathbf{k}} \tag{A.4}$$

$$\mathbf{b} = b_1\hat{\mathbf{i}} + b_2\hat{\mathbf{j}} + b_3\hat{\mathbf{k}} \tag{A.5}$$

If a third vector $\mathbf{c}$ is defined as:

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} \tag{A.6}$$

then $\mathbf{c}$ can also be expressed as

$$\mathbf{c} = (\mathbf{a}\times)\ \mathbf{b} \tag{A.7}$$

where $(\mathbf{a}\times)$ can be used to denote the skew symmetric form of $\mathbf{a}$. This form is sometimes also seen as $\tilde{\mathbf{a}}$. For the general vector $\mathbf{a}$, the skew symmetric form is:

$$\mathbf{a}\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{A.8}$$

For example, the skew symmetric form of the position vector $\mathbf{P}_{\mathrm{AB}}$ is $\mathbf{P}_{\mathrm{AB}}\times$.

For the following derivation, consider the system shown in Fig. A.1. Body A is a floating base with center of mass at point A and frame $\hat{\mathbf{A}}$ attached. Bodies B and C are kinematically connected by rotational joints at points B and C and have their own respective body frames attached. The expanded view shows the spatial reaction forces, denoted by $\mathbf{r}_i$ where i is the body. For each kinematic relation, there will be a parent and a child body, sometimes denoted by λ and i, respectively, in recursive derivations. For example, body A is the parent of B; body B is therefore the child of body A. Note that the points B and C are not the centers of mass for these bodies. These points denote the kinematic joint between the parent and child bodies and are sometimes called "handles" in recursive derivations.

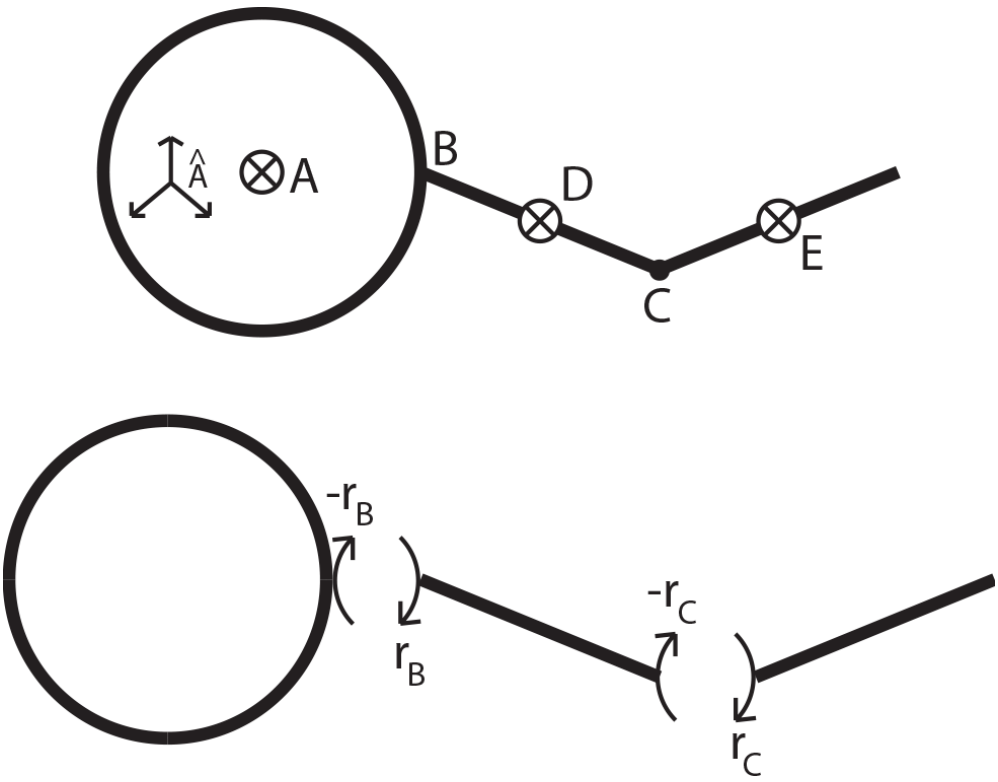

Figure A.1. Kinematic chain with floating base parent body. Free body diagrams show spatial reaction forces.

A.2 Spatial Velocity and Acceleration

This section defines the spatial velocities and accelerations of each body. A vector of known quantities $\mathbf{d}_i$ and matrix ϕ_i must be defined for each body after the ungrounded parent. Both $\mathbf{d}_i$ and ϕ_i are defined not in the i frame, but in that of its parent link. For a rotational joint, the matrix ϕ_i is a constant and may be defined in the preprocessing model description.

A.2.1 Body A

The spatial velocity and acceleration of body A is defined as:

$$\boldsymbol{\vartheta}_A = \begin{bmatrix} \mathbf{V}_{\mathrm{A}} \\ {}^N\boldsymbol{\omega}^A \end{bmatrix} \tag{A.9}$$

$$\dot{\boldsymbol{\vartheta}}_A \;=\; \begin{bmatrix} \dot{\mathbf{V}}_{\mathrm{A}} \\ {}^{N}\dot{\boldsymbol{\omega}}^{A} \end{bmatrix} \tag{A.10}$$

Note that since body A is unconstrained with respect to the ground, the quantities given by Eqns. (A.9) and (A.10) can be fully populated.

A.2.2 Body B

The spatial velocity of the middle body B is constructed from the linear velocity of point B and angular velocity of body B:

$$\mathbf{V}_{\mathrm{B}} \;=\; \mathbf{V}_{\mathrm{A}} \;+\; {}^{\mathrm{A}}\mathbf{V}_{\mathrm{AB}} \;+\; {}^{N}\boldsymbol{\omega}^{A} \times \mathbf{P}_{\mathrm{AB}} \tag{A.11}$$

$${}^{N}\boldsymbol{\omega}^{B} \;=\; {}^{N}\boldsymbol{\omega}^{A} \;+\; {}^{A}\boldsymbol{\omega}^{B} \tag{A.12}$$

Note that the system shown in Fig. A.1 has only rotational joints. While this method can be further generalized to allow for prismatic joints, this derivation is simplified to account for only rotational joints. Therefore it is assumed that all slip velocities are zero (${}^{i}V_{ij} = 0$). Stacking these quantities and collecting terms yields:

$$\boldsymbol{\vartheta}_B \;=\; \begin{bmatrix} \mathbf{V}_{\mathrm{B}} \\ {}^{N}\boldsymbol{\omega}^{B} \end{bmatrix} \;=\; \begin{bmatrix} I_3 & -\mathbf{P}_{\mathrm{AB}}\times \\ 0 & I_3 \end{bmatrix} \begin{bmatrix} \mathbf{V}_{\mathrm{A}} \\ {}^{N}\boldsymbol{\omega}^{A} \end{bmatrix} \;+\; \begin{bmatrix} 0 \\ {}^{A}\boldsymbol{\omega}^{B} \end{bmatrix} \tag{A.13}$$

$$\boldsymbol{\vartheta}_B \;=\; \phi_B \boldsymbol{\vartheta}_A \;+\; s_B \dot{\mathbf{q}}_B$$

where $\dot{\mathbf{q}}_B \in \mathbb{R}^{n\times 1}$ denotes the joint velocities where n is the degrees of freedom for that joint and $s_B \in \mathbb{R}^{6\times n}$ is a vector defining the AB joint space. For example, the quantity s for joint i allowing a single rotation around a shared Y-axis is:

$$s_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{A.14}$$

Note that for joints with multiple degrees of freedom, the quantity s is a matrix and not a vector.

The spatial acceleration of body B is constructed from the linear acceleration of point B and the angular acceleration of body B:

$$\dot{\mathbf{V}}_{\mathrm{B}} \;=\; \dot{\mathbf{V}}_{\mathrm{A}} \;+\; {}^{\mathrm{A}}\dot{\mathbf{V}}_{\mathrm{AB}} \;+\; {}^{N}\dot{\boldsymbol{\omega}}^{A}\times\mathbf{P}_{\mathrm{AB}} + \;2\;{}^{N}\boldsymbol{\omega}^{A}\times{}^{\mathrm{A}}\mathbf{V}_{\mathrm{AB}} \;+\; {}^{N}\boldsymbol{\omega}^{A}\times({}^{N}\boldsymbol{\omega}^{A}\times\mathbf{P}_{\mathrm{AB}}) \quad \text{(A.15)}$$

$$ {}^{N}\dot{\boldsymbol{\omega}}^{B} \;=\; {}^{N}\dot{\boldsymbol{\omega}}^{A} \;+\; {}^{A}\dot{\boldsymbol{\omega}}^{B} \;+\; {}^{N}\boldsymbol{\omega}^{A} \times\; {}^{A}\boldsymbol{\omega}^{B} \quad \text{(A.16)}$$

where, since it is assumed that ${}^{\mathrm{A}}\mathbf{V}_{\mathrm{AB}} = 0$, the slip and coriolis acceleration terms are zero. Stacking Eqns. (A.15) and (A.16) and collecting terms yields:

$$\dot{\boldsymbol{\vartheta}}_{B} \;=\; \begin{bmatrix} \dot{\mathbf{V}}_{\mathrm{B}} \\ {}^{N}\dot{\boldsymbol{\omega}}^{B} \end{bmatrix} \;=\; \begin{bmatrix} I_3 & -\mathbf{P}_{\mathrm{AB}}\times \\ 0 & I_3 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}_{\mathrm{A}} \\ {}^{N}\dot{\boldsymbol{\omega}}^{A} \end{bmatrix} \;+\; \begin{bmatrix} 0 \\ {}^{A}\dot{\boldsymbol{\omega}}^{B} \end{bmatrix} + \begin{bmatrix} {}^{N}\boldsymbol{\omega}^{A}\times({}^{N}\boldsymbol{\omega}^{A}\times\mathbf{P}_{\mathrm{AB}}) \\ {}^{N}\boldsymbol{\omega}^{A}\times\; {}^{A}\boldsymbol{\omega}^{B} \end{bmatrix}$$

$$\dot{\boldsymbol{\vartheta}}_{B} \;=\; \phi_{B}\dot{\boldsymbol{\vartheta}}_{A} \;+\; s_{B}\ddot{\mathbf{q}}_{B} \;+\; \mathbf{d}_{B} \quad \text{(A.17)}$$

A.2.3 Body C

The spatial velocity of the distal body C is constructed from the linear velocity of point C and the angular velocity of body C:

$$\mathbf{V}_{\mathrm{C}} \;=\; \mathbf{V}_{\mathrm{B}} \;+\; {}^{\mathrm{B}}\mathbf{V}_{\mathrm{BC}} \;+\; {}^{N}\boldsymbol{\omega}^{B}\times\mathbf{P}_{\mathrm{BC}} \quad \text{(A.18)}$$

$$ {}^{N}\boldsymbol{\omega}^{C} \;=\; {}^{N}\boldsymbol{\omega}^{B} \;+\; {}^{B}\boldsymbol{\omega}^{C} \quad \text{(A.19)}$$

Stacking Eqns. (A.18) and (A.19) and collecting terms yields:

$$\boldsymbol{\vartheta}_{C} \;=\; \begin{bmatrix} \mathbf{V}_{\mathrm{C}} \\ {}^{N}\boldsymbol{\omega}^{C} \end{bmatrix} \;=\; \begin{bmatrix} I_3 & -\mathbf{P}_{\mathrm{BC}}\times \\ 0 & I_3 \end{bmatrix} \begin{bmatrix} \mathbf{V}_{\mathrm{B}} \\ {}^{N}\boldsymbol{\omega}^{B} \end{bmatrix} \;+\; \begin{bmatrix} 0 \\ {}^{B}\boldsymbol{\omega}^{C} \end{bmatrix} \quad \text{(A.20)}$$

$$\boldsymbol{\vartheta}_{C} \;=\; \phi_{C}\boldsymbol{\vartheta}_{B} \;+\; s_{C}\dot{\mathbf{q}}_{C}$$

The spatial acceleration of body C is constructed from the linear acceleration of point C and the angular acceleration of body C:

$$\dot{\mathbf{V}}_{\mathrm{C}} \;=\; \dot{\mathbf{V}}_{\mathrm{B}} + {}^{\mathrm{B}}\dot{\mathbf{V}}_{\mathrm{BC}} + {}^{N}\dot{\boldsymbol{\omega}}^{B}\times\mathbf{P}_{\mathrm{BC}} + \;2\;{}^{N}\boldsymbol{\omega}^{B}\times{}^{\mathrm{B}}\mathbf{V}_{\mathrm{BC}} + {}^{N}\boldsymbol{\omega}^{B}\times({}^{N}\boldsymbol{\omega}^{B}\times\mathbf{P}_{\mathrm{BC}}) \quad \text{(A.21)}$$

$$ {}^{N}\dot{\boldsymbol{\omega}}^{C} \;=\; {}^{N}\dot{\boldsymbol{\omega}}^{B} \;+\; {}^{B}\dot{\boldsymbol{\omega}}^{C} \;+\; {}^{N}\boldsymbol{\omega}^{B} \times {}^{B}\boldsymbol{\omega}^{C} \tag{A.22} $$

Assuming the the slip and coriolis terms are zero, stacking Eqns. (A.21) and (A.22) and collecting terms yields:

$$ \dot{\boldsymbol{\vartheta}}_C \;=\; \begin{bmatrix} \dot{\mathbf{V}}_{\mathrm{C}} \\ {}^{N}\dot{\boldsymbol{\omega}}^{C} \end{bmatrix} \;=\; \begin{bmatrix} I_3 & -\mathbf{P}_{\mathrm{BC}}\times \\ 0 & I_3 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}_{\mathrm{B}} \\ {}^{N}\dot{\boldsymbol{\omega}}^{B} \end{bmatrix} \;+\; \begin{bmatrix} 0 \\ {}^{B}\dot{\boldsymbol{\omega}}^{C} \end{bmatrix} + \begin{bmatrix} {}^{N}\boldsymbol{\omega}^{B} \times ({}^{N}\boldsymbol{\omega}^{B} \times \mathbf{P}_{\mathrm{BC}}) \\ {}^{N}\boldsymbol{\omega}^{B} \times {}^{B}\boldsymbol{\omega}^{C} \end{bmatrix} $$

$$ \dot{\boldsymbol{\vartheta}}_C \;=\; \phi_C \dot{\boldsymbol{\vartheta}}_B \;+\; s_C \ddot{\mathbf{q}}_C \;+\; \mathbf{d}_C \tag{A.23} $$

A.3 Newton-Euler Equations of Motion

This section defines the spatial equations of motion. These equations are constructed by stacking the Newton-Euler equations. Included in these equations are the unknown reaction forces which will be eliminated through back substitution. For the floating base body A, the sum of moments is taken about the center of mass. For all other bodies, the sum of moments is taken about the input joint. By defining the equations about this point rather than center of mass, the moment created by the reaction forces at that joint is eliminated, simplifying the calculations. A vector of known quantities $\mathbf{b}_i$ must be defined for each body. This vector is defined in the i frame.

A.3.1 Body A

The Newton-Euler equations for body A are:

$$ \left(\sum \mathbf{F}\right)_{\mathrm{A}} \;=\; \frac{d}{dt}\,\mathbf{K}_{\mathrm{A}} \;=\; \frac{d}{dt}\,(m_A \mathbf{V}_{\mathrm{A}}) $$

$$ \mathbf{F}_{\mathrm{A}*} - \mathbf{F}_{\mathrm{B}} \;=\; \dot{m}_A \mathbf{V}_{\mathrm{A}} \;+\; m_A \dot{\mathbf{V}}_{\mathrm{A}} \tag{A.24} $$

$$(\sum \mathbf{M})_{\mathrm{A}} \;=\; \frac{d}{dt}\,\mathbf{H}_{\mathrm{A}} \;=\; \frac{d}{dt}\,(I_{AA}{}^{N}\boldsymbol{\omega}^{A})$$

$$\mathbf{M}_{\mathrm{A}*} \;-\; \mathbf{M}_{\mathrm{B}} \;-\; \mathbf{P}_{\mathrm{AB}} \times \mathbf{F}_{\mathrm{B}} \;=\; \dot{I}_{AA}{}^{N}\boldsymbol{\omega}^{A} \;+\; I_{AA}{}^{N}\dot{\boldsymbol{\omega}}^{A} \;+\; {}^{N}\boldsymbol{\omega}^{A} \times I_{AA}{}^{N}\boldsymbol{\omega}^{A} \tag{A.25}$$

where $\mathbf{H}$ and $\mathbf{K} \in \mathbb{R}^{3\times 1}$ denote angular and linear momenta, respectively, $I_{AA} \in \mathbb{R}^{3\times 3}$ denotes the inertia tensor of body A taken about point A, m_A is the scalar mass of body A, $\mathbf{M}_{\mathrm{A}*}$ and $\mathbf{F}_{\mathrm{A}*} \in \mathbb{R}^{3\times 1}$ denote the external moments and forces acting upon body A, and $\mathbf{M}_{\mathrm{B}}$ and $\mathbf{F}_{\mathrm{B}} \in \mathbb{R}^{3\times 1}$ denote the reaction moments and forces from the A-B joint acting on body A. Note that the external moment and force terms include actuation and control inputs. Also note that it is assumed in this derivation that these bodies are rigid, so the quantities $\dot{I}_{AA}$ and $\dot{m}_A$ are both zero. Stacking Eqns. (A.24) and (A.25) and collecting terms yields:

$$\begin{bmatrix} \mathbf{F}_{\mathrm{A}*} \\ \mathbf{M}_{\mathrm{A}*} \end{bmatrix} - \begin{bmatrix} I_3 & 0 \\ \mathbf{P}_{\mathrm{AB}}\times & I_3 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{\mathrm{B}} \\ \mathbf{M}_{\mathrm{B}} \end{bmatrix} = \begin{bmatrix} m_A I_3 & 0 \\ 0 & I_{AA} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}_{\mathrm{A}} \\ {}^{N}\dot{\boldsymbol{\omega}}^{A} \end{bmatrix} + \begin{bmatrix} 0 \\ {}^{N}\boldsymbol{\omega}^{A} \times I_{AA}{}^{N}\boldsymbol{\omega}^{A} \end{bmatrix}$$

$$\mathbf{g}_A \;-\; \psi_B \mathbf{r}_B \;=\; M_A \dot{\boldsymbol{\vartheta}}_A \;+\; \mathbf{b}_A \tag{A.26}$$

It is of note that the matrix used to modify the reaction forces ψ_i is in fact the transpose of ϕ_i.

A.3.2 Body B

For bodies B and C, Euler's second law may be taken about the input joint, but Newton's second law must be taken with respect to the center of mass. Recall that for body B, the center of mass is point D. The Newton-Euler equations for body B are therefore:

$$(\sum \mathbf{F})_{\mathrm{B}} \;=\; \frac{d}{dt}\,\mathbf{K}_{\mathrm{B}} \;=\; \frac{d}{dt}\,(m_B \mathbf{V}_{\mathrm{D}})$$
$$\mathbf{F}_{\mathrm{B}*} \;+\; \mathbf{F}_{\mathrm{B}} \;-\; \mathbf{F}_{\mathrm{C}} = \; m_B\big[\dot{\mathbf{V}}_{\mathrm{B}} \;+\; {}^{N}\dot{\boldsymbol{\omega}}^{B} \times \mathbf{P}_{\mathrm{BD}} \;+\; {}^{N}\boldsymbol{\omega}^{B} \times ({}^{N}\boldsymbol{\omega}^{B} \times \mathbf{P}_{\mathrm{BD}})\big] \tag{A.27}$$

$$(\sum \mathbf{M})_{\mathrm{B}} \;=\; \frac{d}{dt}\,\mathbf{H}_{\mathrm{B}} \;=\; \frac{d}{dt}\,(I_{BB}{}^{N}\boldsymbol{\omega}^{B})$$

$$\mathbf{M}_{\mathrm{B}*} + \mathbf{M}_{\mathrm{B}} - \mathbf{M}_{\mathrm{C}} - \mathbf{P}_{\mathrm{BC}} \times \mathbf{F}_{\mathrm{C}} \;=\; m_B \mathbf{P}_{\mathrm{BD}} \times \dot{\mathbf{V}}_{\mathrm{B}} + I_{BB}{}^{N}\dot{\boldsymbol{\omega}}^{B} + {}^{N}\boldsymbol{\omega}^{B} \times I_{BB}{}^{N}\boldsymbol{\omega}^{B} \tag{A.28}$$

Note the form of Eqn. (A.28) arises from taking the sum of moments about the input joint at point B - which is not the center of mass. For further explanation of this form, a thorough derivation can be seen in [160]. Also note that I_{BB} is the inertia of body B about point B. This quantity must be constructed using the parallel axis theorem as appropriate. Stacking Eqns. (A.27) and (A.28) and collecting terms yields:

$$\begin{bmatrix} \mathbf{F}_{\mathrm{B}*} \\ \mathbf{M}_{\mathrm{B}*} \end{bmatrix} + \begin{bmatrix} \mathbf{F}_{\mathrm{B}} \\ \mathbf{M}_{\mathrm{B}} \end{bmatrix} - \begin{bmatrix} I_3 & 0 \\ \mathbf{P}_{\mathrm{BC}}\times & I_3 \end{bmatrix} \begin{bmatrix} \mathbf{F}_{\mathrm{C}} \\ \mathbf{M}_{\mathrm{C}} \end{bmatrix}$$

$$= \begin{bmatrix} m_B I_3 & -m_B \mathbf{P}_{\mathrm{BD}}\times \\ m_B \mathbf{P}_{\mathrm{BD}}\times & I_{BB} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}_{\mathrm{B}} \\ {}^{N}\dot{\boldsymbol{\omega}}^{B} \end{bmatrix} + \begin{bmatrix} m_B {}^{N}\boldsymbol{\omega}^{B} \times ({}^{N}\boldsymbol{\omega}^{B} \times \mathbf{P}_{\mathrm{BD}}) \\ {}^{N}\boldsymbol{\omega}^{B} \times I_{BB}{}^{N}\boldsymbol{\omega}^{B} \end{bmatrix} \tag{A.29}$$

$$\mathbf{g}_B \;+\; \mathbf{r}_B \;-\; \psi_C \mathbf{r}_C \;=\; M_B \dot{\boldsymbol{\vartheta}}_B \;+\; \mathbf{b}_B$$

A.3.3 Body C

The Newton-Euler equations for body C are:

$$(\sum \mathbf{F})_{\mathrm{C}} \;=\; \frac{d}{dt}\,\mathbf{K}_{\mathrm{C}} \;=\; \frac{d}{dt}\,(m_C \mathbf{V}_{\mathrm{E})} \tag{A.30}$$

$$\mathbf{F}_{\mathrm{C}*} \;+\; \mathbf{F}_{\mathrm{C}} \;=\; m_C\big[\dot{\mathbf{V}}_{\mathrm{C}} \;+\; {}^{N}\dot{\boldsymbol{\omega}}^{C} \times \mathbf{P}_{\mathrm{CE}} \;+\; {}^{N}\boldsymbol{\omega}^{C} \times ({}^{N}\boldsymbol{\omega}^{C} \times \mathbf{P}_{\mathrm{CE}})\big]$$

$$(\sum \mathbf{M})_{\mathrm{C}} \;=\; \frac{d}{dt}\,\mathbf{H}_{\mathrm{C}} \;=\; \frac{d}{dt}\,(I_{CC}{}^{N}\boldsymbol{\omega}^{C}) \tag{A.31}$$

$$\mathbf{M}_{\mathrm{C}*} \;+\; \mathbf{M}_{\mathrm{C}} \;=\; m_C \mathbf{P}_{\mathrm{CE}} \times \dot{\mathbf{V}}_{\mathrm{C}} \;+\; I_{CC}{}^{N}\dot{\boldsymbol{\omega}}^{C} \;+\; {}^{N}\boldsymbol{\omega}^{C} \times I_{CC}{}^{N}\boldsymbol{\omega}^{C}$$

Stacking Eqns. (A.30) and (A.31) and collecting terms yields:

$$\begin{bmatrix} \mathbf{F}_{C*} \\ \mathbf{M}_{C*} \end{bmatrix} + \begin{bmatrix} \mathbf{F}_{C} \\ \mathbf{M}_{C} \end{bmatrix} = \begin{bmatrix} m_C I_3 & -m_C \mathbf{P}_{CE}\times \\ m_C \mathbf{P}_{CE}\times & I_{CC} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}_{C} \\ {}^N\dot{\boldsymbol{\omega}}^C \end{bmatrix} + \begin{bmatrix} m_C {}^N\boldsymbol{\omega}^C \times ({}^N\boldsymbol{\omega}^C \times \mathbf{P}_{CE}) \\ {}^N\boldsymbol{\omega}^C \times I_{CC} {}^N\boldsymbol{\omega}^C \end{bmatrix}$$

$$\mathbf{g}_C \; + \; \mathbf{r}_C \; = \; M_C \dot{\boldsymbol{\vartheta}}_C \; + \; \mathbf{b}_C \tag{A.32}$$

A.4 Back Substitution

This section shows how the Newton-Euler equations can be combined with the definitions of spatial acceleration to eliminate the reaction forces. This process begins with the distal body and moves along the kinematic chain back to the parent body.

A.4.1 Body C

Consider the spatial acceleration and Newton-Euler equation of body C, given by Eqns. (A.23) and (A.32):

$$\dot{\boldsymbol{\vartheta}}_C \; = \; \phi_C \dot{\boldsymbol{\vartheta}}_B \; + \; s_C \ddot{\mathbf{q}}_C \; + \; \mathbf{d}_C$$

$$\mathbf{g}_C \; + \; \mathbf{r}_C \; = \; M_C \dot{\boldsymbol{\vartheta}}_C \; + \; \mathbf{b}_C$$

Inserting Eqn. (A.23) into (A.32) and collecting terms yields:

$$\mathbf{g}_C \; + \; \mathbf{r}_C \; = \; M_C \big[\phi_C \dot{\boldsymbol{\vartheta}}_B \; + \; s_C \ddot{\mathbf{q}}_C \; + \; \mathbf{d}_C \big] \; + \; \mathbf{b}_C$$

$$\mathbf{r}_C \; = \; M_C \big[\phi_C \dot{\boldsymbol{\vartheta}}_B \; + \; s_C \ddot{\mathbf{q}}_C \big] \; + \; \big[M_C \mathbf{d}_c \; + \; \mathbf{b}_C \; - \; \mathbf{g}_C \big] \tag{A.33}$$

$$\mathbf{r}_C \; = \; M_C \big[\phi_C \dot{\boldsymbol{\vartheta}}_B \; + \; s_C \ddot{\mathbf{q}}_C \big] \; + \; \mathbf{h}_C$$

It it known that there will not be a reaction force or moment in the joint space of the degree of freedom:

$$s_i^T \mathbf{r}_i \; = \; \mathbf{0} \tag{A.34}$$

This identity can be used to solve for the generalized acceleration in terms of the parent body spatial acceleration while eliminating the remaining unknown joint reaction forces:

$$\mathbf{0} \;=\; s_c^T \mathbf{r}_C \;=\; s_c^T \Big[M_C \big[\phi_C \dot{\boldsymbol{\vartheta}}_B \;+\; s_C \ddot{\mathbf{q}}_C \big] \;+\; \mathbf{h}_C \Big]$$

$$s_C^T M_C s_C \; \ddot{\mathbf{q}}_C \;=\; -s_c^T \big[M_C \phi_C \dot{\boldsymbol{\vartheta}}_B \;+\; \mathbf{h}_C \big] \tag{A.35}$$

$$\ddot{\mathbf{q}}_C \;=\; -u_C^{-1} s_c^T \big[M_C \phi_C \dot{\boldsymbol{\vartheta}}_B \;+\; \mathbf{h}_C \big]$$

where

$$u_C \;=\; s_C^T M_C s_C \tag{A.36}$$

For a single-degree-of-freedom joint, the quantity u_i is a scalar value. Substituting Eqn. (A.35) into (A.33) and collecting terms yields:

$$\mathbf{r}_C \;=\; M_C \Big[\phi_C \dot{\boldsymbol{\vartheta}}_B \;-\; s_C u_C^{-1} s_C^T \big[M_C \phi_C \dot{\boldsymbol{\vartheta}}_B \;+\; \mathbf{h}_C \big] \Big] \;+\; \mathbf{h}_C$$

$$\mathbf{r}_C = \big[M_C \phi_C - M_C s_C u_C^{-1} s_C^T M_C \phi_C \big] \dot{\boldsymbol{\vartheta}}_B \;+\; \big[I_6 - M_C s_C u_C^{-1} s_C^T \big] \mathbf{h}_C \tag{A.37}$$

$$\mathbf{r}_C \;=\; P_C M_C \phi_C \dot{\boldsymbol{\vartheta}}_B \;+\; P_C \mathbf{h}_C$$

where

$$P_C \;=\; \big[I_6 - M_C s_C u_C^{-1} s_C^T \big] \tag{A.38}$$

The reaction force at point C has been solved for in terms of the parent body spatial acceleration and vectors of known values only. It can be seen from Eqn. (A.35) that the generalized acceleration can be calculated once the spatial acceleration of the parent body is defined. These forces may now be propagated to the parent body B.

A.4.2 Body B

Consider the reaction force at point C and Newton-Euler equation of body B, given by Eqns. (A.37) and (A.29):

$$\mathbf{r}_C \;=\; P_C M_C \phi_C \dot{\boldsymbol{\vartheta}}_B \;+\; P_C \mathbf{h}_C$$

$$\mathbf{g}_B \;+\; \mathbf{r}_B \;-\; \psi_C \mathbf{r}_C \;=\; M_B \dot{\boldsymbol{\vartheta}}_B \;+\; \mathbf{b}_B$$

Inserting Eqn. (A.37) into (A.29) and collecting terms yields:

$$\mathbf{g}_B \;+\; \mathbf{r}_B \;-\; \psi_C \big[P_C M_C \phi_C \dot{\boldsymbol{\vartheta}}_B + P_C \mathbf{h}_C \big] \;=\; M_B \dot{\boldsymbol{\vartheta}}_B \;+\; \mathbf{b}_B$$

$$\mathbf{r}_B \;=\; \big[M_B + \psi_C P_C M_C \phi_C \big] \dot{\boldsymbol{\vartheta}}_B \;+\; \big[\mathbf{b}_B - \mathbf{g}_B + \psi_C P_C \mathbf{h}_C \big] \tag{A.39}$$

$$\mathbf{r}_B \;=\; M_B^* \dot{\boldsymbol{\vartheta}}_B \;+\; \big[\mathbf{b}_B - \mathbf{g}_B + \psi_C P_C \mathbf{h}_C \big]$$

Recall the spatial acceleration of body B given by Eqn. (A.17):

$$\dot{\boldsymbol{\vartheta}}_B \;=\; \phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; s_B \ddot{\mathbf{q}}_B \;+\; \mathbf{d}_B$$

Inserting Eqn. (A.17) into (A.39) and collecting terms yields:

$$\mathbf{r}_B \;=\; M_B^* \big[\phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; s_B \ddot{\mathbf{q}}_B \;+\; \mathbf{d}_B \big] \;+\; \big[\mathbf{b}_B - \mathbf{g}_B + \psi_C P_C \mathbf{h}_C \big]$$

$$\mathbf{r}_B = M_B^* \big[\phi_B \dot{\boldsymbol{\vartheta}}_A + s_B \ddot{\mathbf{q}}_B \big] + \big[M_B^* \mathbf{d}_B + \mathbf{b}_B - \mathbf{g}_B + \psi_C P_C \mathbf{h}_C \big] \tag{A.40}$$

$$\mathbf{r}_B \;=\; M_B^* \big[\phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; s_B \ddot{\mathbf{q}}_B \big] \;+\; \big[\mathbf{h}_B \;+\; \psi_C P_C \mathbf{h}_C \big]$$

$$\mathbf{r}_B \;=\; M_B^* \big[\phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; s_B \ddot{\mathbf{q}}_B \big] \;+\; \mathbf{h}_B^*$$

Using the identity in Eqn. (A.34) to eliminate the reaction forces and solve for the generalized acceleration:

$$\mathbf{0} \;=\; s_B^T \mathbf{r}_B \;=\; s_B^T \Big[M_B^* \big[\phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; s_B \ddot{\mathbf{q}}_B \big] \;+\; \mathbf{h}_B^* \Big]$$

$$s_B^T M_B^* s_B \; \ddot{\mathbf{q}}_B \;=\; -s_B^T \big[M_B^* \phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; \mathbf{h}_B^* \big] \tag{A.41}$$

$$\ddot{\mathbf{q}}_B \;=\; -u_B^{-1} s_B^T \big[M_B^* \phi_B \dot{\boldsymbol{\vartheta}}_A + \mathbf{h}_B^* \big]$$

where

$$u_B \;=\; s_B^T M_B^* s_B \tag{A.42}$$

Inserting Eqn. (A.41) into (A.40) and collecting terms yields:

$$\mathbf{r}_B \;=\; M_B^* \Big[\phi_B \dot{\boldsymbol{\vartheta}}_A \;- s_B u_B^{-1} s_B^T \big[M_B^* \phi_B \dot{\boldsymbol{\vartheta}}_A + \mathbf{h}_B^* \big] \Big] \;+\; \mathbf{h}_B^*$$

$$\mathbf{r}_B = \big[M_B^* \phi_B - M_B^* s_B u_B^{-1} s_B^T M_B^* \phi_B \big] \dot{\boldsymbol{\vartheta}}_A \;+\; \big[I_6 - M_B^* s_B u_B^{-1} s_B^T \big] \mathbf{h}_B^* \tag{A.43}$$

$$\mathbf{r}_B \;=\; P_B M_B^* \phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; P_B \mathbf{h}_B^*$$

where

$$P_B \;=\; \big[I_6 \;-\; M_B^* s_B u_B^{-1} s_B^T \big] \tag{A.44}$$

The reaction forces at point B have been solved for in terms of the parent body spatial acceleration and vectors of known values only. These forces may now be propagated to the parent body A.

A.4.3 Body A

Consider the reaction force at point B and Newton-Euler equation of body A, given by Eqns. (A.43) and (A.26):

$$\mathbf{r}_B \;=\; P_B M_B^* \phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; P_B \mathbf{h}_B^*$$

$$\mathbf{g}_A \;-\; \psi_B \mathbf{r}_B \;=\; M_A \dot{\boldsymbol{\vartheta}}_A \;+\; \mathbf{b}_A$$

Inserting Eqn. (A.43) into (A.26) and collecting terms yields:

$$\mathbf{g}_A \;-\; \psi_B \big[P_B M_B^* \phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; P_B \mathbf{h}_B^* \big] \;=\; M_A \dot{\boldsymbol{\vartheta}}_A \;+\; \mathbf{b}_A$$

$$\begin{aligned} \mathbf{0} \;&=\; \big[M_A + \psi_B P_B M_B^* \phi_B \big] \dot{\boldsymbol{\vartheta}}_A \;+\; \big[\mathbf{b}_A - \mathbf{g}_A + \psi_B P_B \mathbf{h}_B^* \big] \\ \mathbf{0} \;&=\; M_A^* \dot{\boldsymbol{\vartheta}}_A \;+\; \big[\mathbf{h}_A + \psi_B P_B \mathbf{h}_B^* \big] \\ \mathbf{0} \;&=\; M_A^* \dot{\boldsymbol{\vartheta}}_A \;+\; \mathbf{h}_A^* \end{aligned} \tag{A.45}$$

The spatial acceleration of the floating base parent body can now be found by inverting the augmented mass matrix of body A:

$$\dot{\boldsymbol{\vartheta}}_A \;=\; -\big(M_A^* \big)^{-1} \big(\mathbf{h}_A^* \big) \tag{A.46}$$

Consider what this algorithm has accomplished: by eliminating the unknown reaction forces at joints, the generalized accelerations of the parent body can be solved for independently by inversion of, at most, a 6×6 matrix. These calculated accelerations can then be propagated forward to solve for the next set of accelerations, and so on. Using this algorithm, the symbolic equations of motion can be avoided entirely. As mentioned before, methods that invert the symbolic mass matrix have a computational cost of $\mathrm{O}(N^3)$ that scales with the degrees of freedom cubed. The recursive algorithm has a computational cost of $\mathrm{O}(N)$ when performed serially and $\mathrm{O}(logN)$ when performed in parallel. For small systems, the symbolic approach performs admirably, but is quickly outpaced for larger systems.

A.5 Further Comments

In the case of spherical joints, the calculated joint accelerations $\dot{\omega}$ cannot be directly integrated to find useable joint velocities ω. These joints are often described

using Euler angles or Euler parameters. Euler angles, which consist of a series of three simple rotations, are simple to implement but susceptible to singularity when the middle rotation is such that the first and third axes are aligned. While there are several combinations of Euler angles that can be selected, each of these choices suffers from the same singularity problem. A more difficult but more robust solution is the use of Euler parameters which defines the rotation using four coordinates. This rotation definition is singularity free, even at the specific orientation that causes Euler angles to fail. However, this definition requires that the equation of motion be converted into the Euler parameter vector space and that a normality constraint be implemented. The work described in Ch. 5 used the recursive algorithm detailed above with Euler parameters at each joint. How the recursive algorithm was adapted for Euler parameters is described in Appendices B and C.

Also, it should be noted this derivation makes no mention of the frames of the quantities. As such, the user must take care to consider what frames each quantity is defined in. When two quantities are added or subtracted, it must be ensured that these quantities are in the same frame. The user may find it convenient to define transformation matrices of the form:

$$\chi = \begin{bmatrix} R & 0_{3\times3} \\ 0_{3\times3} & R \end{bmatrix} \tag{A.47}$$

where R denotes a (3×3) rotation matrix.

APPENDIX B

EULER PARAMETERS

B.1 Euler Parameters Introduction

This appendix presents the Euler parameters formulation used to avoid the singularity problem associated with general rotations between coordinate frames. Body attached coordinate frames are convenient for defining the changes in orientation experienced by moving bodies. This change in orientation is referred to as a rotation. For the simplest case, this rotation occurs about a single axis shared by both frames. These single axis rotations are often referred to as simple rotations. Consider the simple rotation shown in Fig. B.1 where the frame $\widehat{\mathbf{A}}$ rotates with respect to an intuitively known and non-moving frame $\widehat{\mathbf{N}}$. From this diagram, a series of claims can be made defining the axes of the $\widehat{\mathbf{A}}$ frame with respect to the $\widehat{\mathbf{N}}$ frame:

$$\begin{aligned} \widehat{\mathbf{A}}_1 &= cos(\theta)\widehat{\mathbf{N}}_1 + sin(\theta)\widehat{\mathbf{N}}_2 \\ \widehat{\mathbf{A}}_2 &= -sin(\theta)\widehat{\mathbf{N}}_1 + cos(\theta)\widehat{\mathbf{N}}_2 \\ \widehat{\mathbf{A}}_3 &= \widehat{\mathbf{N}}_3 \end{aligned} \tag{B.1}$$

These claims can be written in vector-matrix form:

$$\begin{bmatrix} \widehat{\mathbf{A}}_1 \\ \widehat{\mathbf{A}}_2 \\ \widehat{\mathbf{A}}_3 \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{N}}_1 \\ \widehat{\mathbf{N}}_2 \\ \widehat{\mathbf{N}}_3 \end{bmatrix} \tag{B.2}$$

$$\widehat{\mathbf{A}} = {}^{\mathrm{N}}_{\mathrm{A}}R\, \widehat{\mathbf{N}}$$

where ${}^{\mathrm{N}}_{\mathrm{A}}R$ defines the rotation of frame $\widehat{\mathbf{A}}$ with respect to frame $\widehat{\mathbf{N}}$. Recognize that this matrix transforms vectors defined in the $\widehat{\mathbf{N}}$ frame into the $\widehat{\mathbf{A}}$ frame. For transforming vectors defined in the $\widehat{\mathbf{A}}$ frame into the $\widehat{\mathbf{N}}$ frame, the transpose of ${}^{\mathrm{N}}_{\mathrm{A}}R$ may be used:

$$\begin{aligned} \widehat{\mathbf{N}} &= {}^{\mathrm{A}}_{\mathrm{N}}R\, \widehat{\mathbf{A}} \\ {}^{\mathrm{A}}_{\mathrm{N}}R &= {}^{\mathrm{N}}_{\mathrm{A}}R^T \end{aligned} \tag{B.3}$$

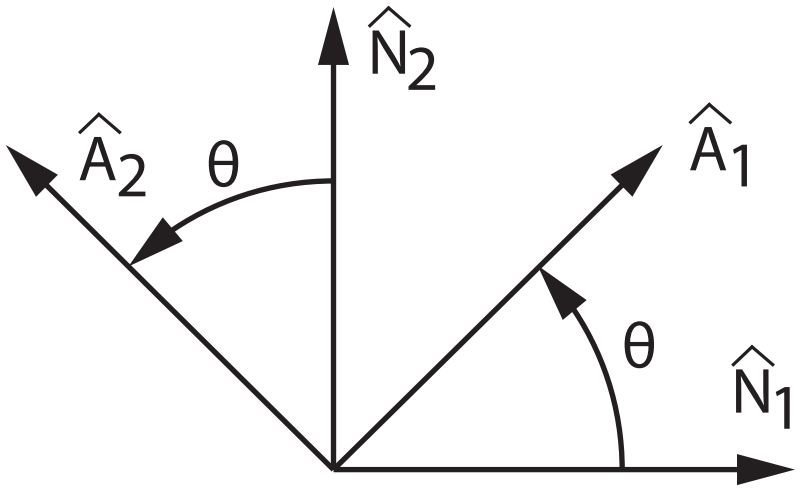

Figure B.1. Simple rotation about a shared $\widehat{\mathbf{3}}$ axis.

These simple rotations can also be staged in series across multiple frames:

$$\begin{aligned} \widehat{\mathbf{B}} &= {}^{\mathrm{A}}_{\mathrm{B}}R\ \widehat{\mathbf{A}} \\ \widehat{\mathbf{A}} &= {}^{\mathrm{N}}_{\mathrm{A}}R\ \widehat{\mathbf{N}} \\ \longrightarrow\ \widehat{\mathbf{B}} &= {}^{\mathrm{A}}_{\mathrm{B}}R\ {}^{\mathrm{N}}_{\mathrm{A}}R\ \widehat{\mathbf{N}} = {}^{\mathrm{N}}_{\mathrm{B}}R\ \widehat{\mathbf{N}} \end{aligned} \tag{B.4}$$

Defining the general rotation of one frame with respect to another frame is most intuitively done using Euler angles, a sequence of three simple rotations about any set of non-consecutive orthogonal vectors:

$${}^{\mathrm{N}}_{\mathrm{A}}R = R_1(\theta_1)\ R_2(\theta_2)\ R_3(\theta_3) \tag{B.5}$$

where each rotation angle θ_i is the Euler angle. This definition shows that for an arbitrary rotation, there are three degrees of freedom. In the dynamics of aircraft, these Euler angles are often referred to as roll, pitch, and yaw.

However, regardless of the sequence of vectors chosen - be they defined in inertial or body frame - the resulting rotation matrix is susceptible to singularities at certain orientations. These singularities occur when the middle angle is such that the first and third rotation axes are aligned. A singularity here refers to a mathematical event

in which a matrix loses rank or a quantity becomes ambiguous. In the case of Euler angles, the individual angles become unrecoverable and ambiguous. This singularity can lead to failures in simulations and control mechanisms.

Many alternatives to Euler angles exist for defining general rotations, such as Principle Axis-Angles and Rodrigues Parameters. For further discussion on these two formulations, see [160]. Another alternative to Euler angles is Euler parameters, a set of four homogenous coordinates:

$$\mathbf{e} \;=\; [\; e_0 \; e_1 \; e_2 \; e_3 \;]^T \tag{B.6}$$

where $\mathbf{e}$ denotes the four-tuple of Euler parameters. These four Euler parameters can be related to the Principle Axis-Angles rotation definition. Considering the general rotation shown in Fig. B.2, the Euler parameters are defined by:

$$\begin{aligned} e_0 &= cos\Big(\frac{\phi}{2}\Big) \\ e_1 &= cos(\theta_1)\; sin\Big(\frac{\phi}{2}\Big) \\ e_2 &= cos(\theta_2)\; sin\Big(\frac{\phi}{2}\Big) \\ e_3 &= cos(\theta_3)\; sin\Big(\frac{\phi}{2}\Big) \end{aligned} \tag{B.7}$$

where

$$cos(\theta_1) \;=\; \widehat{\mathbf{n}} \cdot \widehat{\mathbf{N}}_1 \qquad cos(\theta_2) \;=\; \widehat{\mathbf{n}} \cdot \widehat{\mathbf{N}}_2 \qquad cos(\theta_3) \;=\; \widehat{\mathbf{n}} \cdot \widehat{\mathbf{N}}_3 \tag{B.8}$$

and $\widehat{\mathbf{n}}$ denotes the principle axis of rotation and ϕ is a rotation angle about that principle axis.

Recall that the definition of Euler angles shows that there are three degrees of freedom associated with an arbitrary rotation, so some constraint must exist amongst

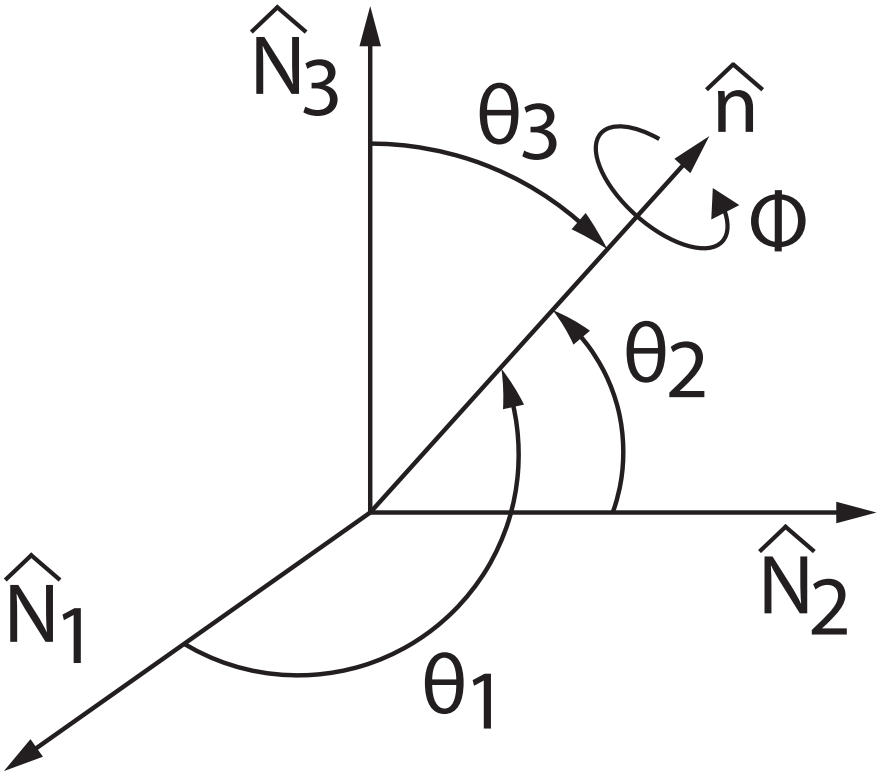

Figure B.2. Rotation about an arbitrary axis $\widehat{\mathbf{n}}$.

the Euler parameters to address the fourth coordinate. A normality constraint is often implemented of the form:

$$\begin{aligned} e_0^2 \ + \ e_1^2 \ + \ e_2^2 \ + \ e_3^2 \ &= \ 1 \\ \mathbf{e}^T\mathbf{e} \ &= \ 1 \end{aligned} \tag{B.9}$$

where the derivative of Eqn. (B.9) provides the useful speed constraint:

$$\begin{aligned} e_0\dot{e}_0 \ + \ e_1\dot{e}_1 \ + \ e_2\dot{e}_2 \ + \ e_3\dot{e}_3 \ \ &= \ 0 \\ \dot{\mathbf{e}}^T\mathbf{e} \ + \ \mathbf{e}^T\dot{\mathbf{e}} \ &= \ 0 \\ \rightarrow \ \dot{\mathbf{e}}^T\mathbf{e} \ &= \ 0 \end{aligned} \tag{B.10}$$

Given a set of Euler parameters $\mathbf{e}$, the rotation matrix associated with transforming vectors from the $\widehat{\mathbf{A}}$ frame to $\widehat{\mathbf{N}}$ frame is given by:

$$
{}^{\mathrm{A}}_{\mathrm{N}}R = 2\begin{bmatrix} \frac{1}{2} - e_2^2 - e_3^2 & e_1e_2 - e_3e_0 & e_1e_3 + e_2e_0 \\ e_1e_2 + e_3e_0 & \frac{1}{2} - e_1^2 - e_3^2 & e_2e_3 - e_1e_0 \\ e_1e_3 - e_2e_0 & e_2e_3 + e_1e_0 & \frac{1}{2} - e_1^2 - e_2^2 \end{bmatrix} \tag{B.11}
$$

The same rotation matrix may also be generated from:

$$
\begin{aligned} {}^{\mathrm{A}}_{\mathrm{N}}R &= I_3\left[1 - 2\ \epsilon^T\epsilon\right] + 2\ \epsilon\ \epsilon^T + 2\ e_0\epsilon\times \\ \epsilon &= \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T \end{aligned} \tag{B.12}
$$

where I_3 denotes a 3×3 identity matrix.

B.2 Angular Velocity

The angular velocity vector $\boldsymbol{\omega}$ can be related to the time derivatives of the Euler parameters:

$$
\dot{\mathbf{e}} = \frac{1}{2}\begin{bmatrix} e_0 & -e_3 & e_2 & e_1 \\ e_3 & e_0 & -e_1 & e_2 \\ -e_2 & e_1 & e_0 & e_3 \\ -e_1 & -e_2 & -e_3 & e_0 \end{bmatrix}\bar{\boldsymbol{\omega}} \tag{B.13}
$$

$$
\dot{\mathbf{e}} = \frac{1}{2}E\bar{\boldsymbol{\omega}}
$$

where $\bar{\boldsymbol{\omega}}$ denotes $\boldsymbol{\omega}$ mapped to (4×1) space:

$$
\bar{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \tag{B.14}
$$

Equation (B.13) may also be written as:

$$
\dot{\mathbf{e}} = \frac{1}{2}E\eta^T\boldsymbol{\omega} \tag{B.15}
$$

where η^T is a 4×3 matrix used to map $\omega$to a 4×1 vector space:

$$\eta^T \;=\; \begin{bmatrix} I_3 \\ 0_{1\times 3} \end{bmatrix} \tag{B.16}$$

Note that the angular velocity $\boldsymbol{\omega}$ is defined strictly in the body frame. Equation (B.15) may be rearranged to solve for $\boldsymbol{\omega}$ in terms of $\dot{e}$:

$$\boldsymbol{\omega} \;=\; 2\eta E^T \dot{\mathbf{e}} \tag{B.17}$$

It can be seen from Eqns. (B.15) and (B.17) that η might be unncessary if E were defined as a 4×3 matrix. However, it is necessary to define E as a 4×4 matrix to ensure that the inverse of E is also the transpose:

$$E^{-1} \;=\; E^T \tag{B.18}$$

such that

$$E^T E \;=\; E E^T \;=\; I_4 \tag{B.19}$$

Taking the derivative of Eqn. (B.19) leads to the useful identity:

$$\dot{E}^T E + E^T \dot{E} \;=\; \dot{E} E^T + E \dot{E}^T \;=\; 0_{4\times 4} \tag{B.20}$$

B.3 Angular Acceleration

The angular acceleration can be found by taking the derivative of Eqn. (B.17)

$$\dot{\boldsymbol{\omega}} \;=\; 2\eta\big[\; E^T \ddot{\mathbf{e}} \;+\; \dot{E}^T \dot{\mathbf{e}} \;\big] \tag{B.21}$$

where $\dot{E}$ is defined by taking the derivative of E:

$$\dot{E} \;= \frac{d}{dt} E \tag{B.22}$$

Equation (B.21) may also be found by taking the derivative of Eqn. (B.15):

$$\ddot{\mathbf{e}} \;=\; \frac{1}{2}[\; E\eta^T \dot{\boldsymbol{\omega}} \;+\; \dot{E}\eta^T \boldsymbol{\omega} \;] \tag{B.23}$$

and by taking advantage Eqn. (B.19) and the useful identity:

$$\eta\ \eta^T\ = I_3 \tag{B.24}$$

the quantity $\dot{\boldsymbol{\omega}}$ may be isolated:

$$\frac{1}{2}E\eta^T\dot{\boldsymbol{\omega}}\ =\ \ddot{\mathbf{e}}\ -\ \frac{1}{2}\dot{E}\eta^T\boldsymbol{\omega}$$

$$\eta^T\dot{\boldsymbol{\omega}}\ =\ 2E^T\ddot{\mathbf{e}}\ -\ E^T\dot{E}\eta^T\boldsymbol{\omega} \tag{B.25}$$

$$\dot{\boldsymbol{\omega}}\ =\ 2\eta E^T\ddot{\mathbf{e}}\ -\ \eta E^T\dot{E}\eta^T\boldsymbol{\omega}$$

Inserting Eqn. (B.15) into Eqn. (B.25) yields:

$$\dot{\boldsymbol{\omega}}\ =\ 2\eta E^T\ddot{\mathbf{e}}\ -\ 2\eta E^T\dot{E}\eta^T\eta E^T\dot{\mathbf{e}} \tag{B.26}$$

Consider the second terms on the left hand sides of Eqn. (B.21) and (B.26). For those two equations to be equal, the following claim must be shown to be true:

$$2\eta\dot{E}^T\ =\ -2\eta E^T\dot{E}\eta^T\eta E^T \tag{B.27}$$

This equivalence can be shown by:

$$\begin{aligned}
\dot{E}^T\ &=\ -E^T\dot{E}\eta^T\eta E^T \\
E\dot{E}^T\ &=\ -\dot{E}\eta^T\eta E^T \\
E^T\dot{E}^TE\ &=\ -\dot{E}\eta^T\eta \\
E^T\dot{E}^TE\eta^T\ &=\ -\dot{E}\eta^T \\
E^T\dot{E}^TE\ &=\ -\dot{E} \\
\dot{E}^TE\ &=\ -E^T\dot{E} \\
\dot{E}^TE\ +\ E^T\dot{E}\ &=\ 0
\end{aligned} \tag{B.28}$$

which is the identity given in Eqn. (B.20). Equation (B.26) may therefore be expressed as

$$\dot{\boldsymbol{\omega}} \;=\; 2\eta E^T \ddot{\mathbf{e}} \;-\; 2\eta \dot{E}^T \dot{\mathbf{e}} \tag{B.29}$$

which is Eqn. (B.21).

B.4 Change in Angular Momentum

Consider Euler's second law:

$$(\textstyle\sum \mathbf{M}) \;=\; \dfrac{d}{dt}\mathbf{H} \;=\; I\dot{\boldsymbol{\omega}} \;+\; \boldsymbol{\omega} \times I\boldsymbol{\omega} \tag{B.30}$$

Inserting Eqns. (B.17) and (B.21) into Eqn. (B.30) yields:

$$\dot{\mathbf{H}} \;=\; I(2\eta E^T \ddot{\mathbf{e}} - 2\eta \dot{E}^T \dot{\mathbf{e}}) \;+\; (2\eta E^T \dot{\mathbf{e}}) \times I(2\eta E^T \dot{\mathbf{e}}) \tag{B.31}$$

For convenience, an intermediate variable L and its time derivative $\dot{L}$ can be defined:

$$\begin{aligned} L \;&=\; 2\eta E^T \\ \dot{L} \;&=\; 2\eta \dot{E}^T \end{aligned} \tag{B.32}$$

Equation (B.31) may now be expressed as:

$$\dot{\mathbf{H}} \;=\; I(L\ddot{\mathbf{e}} \;+\; \dot{L}\dot{\mathbf{e}}) \;+\; (L\dot{\mathbf{e}}) \times I(L\dot{\mathbf{e}}) \tag{B.33}$$

For convenience, an intermediate variable K can be defined:

$$K \;=\; I\dot{L}\dot{\mathbf{e}} \;+\; (L\dot{\mathbf{e}}) \times I(L\dot{\mathbf{e}}) \tag{B.34}$$

Equation (B.33) may now be expressed as:

$$\dot{\mathbf{H}} \;=\; IL\ddot{\mathbf{e}} \;+\; K \tag{B.35}$$

It can be seen that L and K are both known quantities and are size 3×4 and 3×1, respectively.

B.5 Equations of Motion

Of the multiple methods for generating the equations of motion, consider the method formulated by Thomas Kane [150]. Kane's method is an iterative process exceptionally useful for generating the equations of motion when using Euler parameters. The scalar equation of motion for the i^{th} degree of freedom is given by:

$$F_i - F_i^* = 0 \tag{B.36}$$

$$F_i \;=\; \sum_{j}^{bodies} \left[\; (\sum \mathbf{F})_{\mathrm{j}} \cdot \frac{\partial \mathbf{V}_{\mathrm{j}}}{\partial \dot{q}_i} \;+\; (\sum \mathbf{M})_{\mathrm{j}} \cdot \frac{\partial \boldsymbol{\omega}_j}{\partial \dot{q}_i} \; \right] \tag{B.37}$$

$$F_i^* \;=\; \sum_{j}^{bodies} \left[\; m_j \dot{\mathbf{V}}_{\mathrm{j}} \cdot \frac{\partial \mathbf{V}_{\mathrm{j}}}{\partial \dot{q}_i} \;+\; \dot{\mathbf{H}}_j \cdot \frac{\partial \boldsymbol{\omega}_j}{\partial \dot{q}_i} \; \right] \tag{B.38}$$

where F_i and F_i^* are called the Generalized Active Forces and Generalized Inertia Forces, respectively. It can be seen that there are several quantities that must be defined:

1. Linear and angular velocity ($\mathbf{V}$ and $\boldsymbol{\omega}$)
2. Change in linear and angular momentum ($m\dot{\mathbf{V}}$ and $\dot{\mathbf{H}}$)
3. Sum of forces and moments ($(\sum \mathbf{F})$ and $(\sum \mathbf{M})$)

Consider an ungrounded body A with body attached frame $\widehat{\mathbf{A}}$. Body A has six degrees of freedom associated with translation and rotation. This body has a known mass m and inertia matrix I about its center of mass defined in body frame. Euler parameters are used to avoid singularity problems when defining the rotation matrix for the floating base. There are then seven generalized coordinates consisting of the three translations and four Euler parameters:

$$\mathbf{r} = [\; q_1 \; q_2 \; q_3 \;]^T \tag{B.39}$$

$$\mathbf{e} = [\; q_4 \; q_5 \; q_6 \; q_7 \;]^T \tag{B.40}$$

$$\mathbf{q} = [\; \mathbf{r}^T \; \mathbf{e}^T \;]^T \tag{B.41}$$

with the vectors $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ denoting the time derivatives. The position, velocity, and acceleration of the center of mass may be defined as:

$$\mathbf{P}_{\mathrm{NA}} \;=\; q_1\,\widehat{\mathbf{N}}_1 \;+\; q_2\,\widehat{\mathbf{N}}_2 \;+ q_3\,\widehat{\mathbf{N}}_3 \tag{B.42}$$

$$\mathbf{V}_{\mathrm{A}} \;=\; \dot{q}_1\,\widehat{\mathbf{N}}_1 \;+\; \dot{q}_2\,\widehat{\mathbf{N}}_2 \;+\; \dot{q}_3\,\widehat{\mathbf{N}}_3 \tag{B.43}$$

$$\dot{\mathbf{V}}_{\mathrm{A}} \;=\; \ddot{q}_1\,\widehat{\mathbf{N}}_1 \;+\; \ddot{q}_2\,\widehat{\mathbf{N}}_2 \;+\; \ddot{q}_3\,\widehat{\mathbf{N}}_3 \tag{B.44}$$

The rotation matrix R, transformation matrix E, and its time derivative $\dot{E}$ are defined using $\mathbf{e}$ and $\dot{\mathbf{e}}$ according to Eqns. (B.11), (B.13), and (B.22). The angular velocity and angular accelerations are defined according to Eqns. (B.17) and (B.21) and are repeated here:

$$\boldsymbol{\omega} \;=\; 2\eta E^T\dot{\mathbf{e}}$$

$$\dot{\boldsymbol{\omega}} \;=\; 2\eta\big[\; E^T\ddot{\mathbf{e}} \;+\; \dot{E}^T\dot{\mathbf{e}} \;\big]$$

The change in angular momentum $\dot{\mathbf{H}}$ is given by Eqn. (B.35) and is repeated here:

$$\dot{\mathbf{H}} \;=\; IL\ddot{\mathbf{e}} \;+\; K$$

A known set of external forces and moments are defined as:

$$\mathbf{F} \;=\; \Big[\; F_1 \;F_2 \;F_3\Big]^T \tag{B.45}$$

$$\mathbf{M} \;=\; \Big[\; M_1 \;M_2 \;M_3\Big]^T \tag{B.46}$$

All the quantities required for generating the equations of motion using Kane's method are now defined. For the three degrees of freedom associated with translation, the partial derivatives are:

$$\frac{\partial\boldsymbol{\omega}}{\partial\dot{\mathbf{q}}_{1-3}} \;=\; \mathbf{0} \tag{B.47}$$

$$\frac{\partial\mathbf{V}_{\mathrm{A}}}{\partial\dot{\mathbf{q}}_{1-3}} \;=\; I_3 \tag{B.48}$$

The Generalized Active Forces and Generalized Inertia Forces are then:

$$\mathbf{F}_{1-3} = \mathbf{F} \cdot I_3 = \mathbf{F} \tag{B.49}$$

$$\mathbf{F}^*_{1-3} = m\dot{\mathbf{V}}_{\mathrm{A}} \cdot I_3 = m\dot{\mathbf{V}}_{\mathrm{A}} \tag{B.50}$$

The equations of motion for the three translational degrees of freedom may be succinctly expressed as:

$$\mathbf{F}_{1-3} - \mathbf{F}^*_{1-3} = \mathbf{F} - m\dot{\mathbf{V}}_{\mathrm{A}} = \mathbf{0} \tag{B.51}$$

The equation of motion for the translation of the body in the $\widehat{\mathbf{N}}_1$ direction is then:

$$m\ddot{q}_1 = F_1 \tag{B.52}$$

For the three degrees of freedom associated with rotation, the partial derivatives are:

$$\frac{\partial \boldsymbol{\omega}}{\partial \dot{\mathbf{q}}_{4-7}} = \frac{\partial \boldsymbol{\omega}}{\partial \dot{\mathbf{e}}} = 2\eta E^T \tag{B.53}$$

$$\frac{\partial \mathbf{V}_{\mathrm{A}}}{\partial \dot{\mathbf{q}}_{4-7}} = \frac{\partial \mathbf{V}_{\mathrm{A}}}{\partial \dot{\mathbf{e}}} = \mathbf{0} \tag{B.54}$$

The Generalized Active Forces and Generalized Inertia Forces are then:

$$\mathbf{F}_{4-7} = \mathbf{M} \cdot (2\eta E^T) = 2(\eta E^T)^T \mathbf{M} = 2E\eta^T \mathbf{M} \tag{B.55}$$

$$\mathbf{F}^*_{4-7} = \dot{\mathbf{H}} \cdot (2\eta E^T) = 2(\eta E^T)^T (IL\ddot{\mathbf{e}} + K) = 2E\eta^T \big(IL\ddot{\mathbf{e}} + K\big) \tag{B.56}$$

The equations of motion for the four Euler parameters may be expressed as:

$$\mathbf{F}_{4-7} - \mathbf{F}^*_{4-7} = 2E\eta^T \mathbf{M} - 2E\eta^T (IL\ddot{\mathbf{e}} + K) = \mathbf{0} \tag{B.57}$$

Rearranging Eqn. B.57 yields:

$$\begin{aligned} 2E\eta^T IL\, \ddot{\mathbf{e}} + 2E\eta^T K &= 2E\eta^T \mathbf{M} \\ L^T IL\, \ddot{\mathbf{e}} + L^T K &= L^T \mathbf{M} \end{aligned} \tag{B.58}$$

which may be further rearranged to collect all known quantities:

$$L^T I L\, \ddot{\mathbf{e}} \;=\; L^T(\mathbf{M} - K) \tag{B.59}$$

Equation (B.58) can be represented by the general form:

$$M(\mathbf{e})\, \ddot{\mathbf{e}} \;+\; C(\mathbf{e}, \dot{\mathbf{e}}) \;=\; \mathbf{\Gamma}(\mathbf{e}) \tag{B.60}$$

These rotational equations of motion can be combined with the equations for translation shown in Eqn. (B.51). These combined equations of motion can then be represented by the general form:

$$M(\mathbf{q})\, \ddot{\mathbf{q}} \;+\; C(\mathbf{q}, \dot{\mathbf{q}}) \;=\; \mathbf{\Gamma}(\mathbf{q}) \tag{B.61}$$

where $M(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{\Gamma}(\mathbf{q})$ denote the mass matrix, nonlinear velocity product terms, and the Generalized Active Forces, respectively, mapped to the 7-dimensional space corresponding to translation and the Euler parameters. Note that in the case of a spherical joint, the dimensions of Eqn. (B.60) are 4×1.

Consider the equation of motion for an ungrounded body A solved using the recursive formulation presented in Appendix A:

$$M_A^* \dot{\boldsymbol{\vartheta}}_A \;=\; -\mathbf{h}_A^* \tag{B.62}$$

The M_A^* term denotes a modified mass matrix and has dimensions 6×6. To solve for $\ddot{\mathbf{e}}_A$ in place of ${}^N\dot{\boldsymbol{\omega}}^A$, Eqn. (B.62) must be transformed to the Euler parameter vector space similar to Eqn. (B.59). This may be accomplished by:

$$Y_A^T M_A^* Y_A \dot{\boldsymbol{\vartheta}}_A^{\dagger} \;=\; -Y_A^T \mathbf{h}_A^* \tag{B.63}$$

$$Y_A \;=\; \begin{bmatrix} I_3 & 0_{3\times 4} \\ 0_{3\times 3} & L_A \end{bmatrix} \tag{B.64}$$

$$\dot{\boldsymbol{\vartheta}}_A^{\dagger} \;=\; \begin{bmatrix} \dot{\mathbf{V}}_{\mathrm{A}} \\ \ddot{\mathbf{e}}_A \end{bmatrix} \tag{B.65}$$

Also, consider the equation of motion for the kinematic joint B shown by Eqn. (A.41), which is repeated here:

$$s_B^T M_B^* s_B \; \ddot{\mathbf{q}}_B \;=\; -s_B^T \big[M_B^* \phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; \mathbf{h}_B^* \big] \tag{B.66}$$

If joint B is spherical, the dimensions of s_B are 6×3. The $s_B^T M_B^* s_B$ term denotes a modified mass matrix and has dimensions 3×3. Since the spatial acceleration of body A is calculated first it is known with respect to solving Eqn. (B.66). This equation then can be converted to a form similar to Eqn. (B.59) to solve for the Euler accelerations at that joint:

$$L_B^T s_B^T M_B^* s_B L_B \; \ddot{\mathbf{e}}_B \;=\; -L_B^T s_B^T \big[M_B^* \phi_B \dot{\boldsymbol{\vartheta}}_A \;+\; \mathbf{h}_B^* \big] \tag{B.67}$$

where the new mass matrix is given by $L_B^T s_B^T M_B^* s_B L_B$ and has the dimensions 4×4. Because the mass matrices in Eqns. (B.58), (B.63), and (B.67) have been mapped to the Euler parameter vector space, they are singular and non-invertible. However, there is the normality constraint associated with the Euler parameters, given by Eqn. (B.9), that can be applied to eliminate a dependent degree of freedom. Application of this constraint will reduce the size of the mass matrix by one, allowing that mass matrix to be inverted to solve for the independent accelerations. Those independent accelerations can then be used to solve for the dependent acceleration. These accelerations can be integrated to update states for the next integration step. Appendix C discusses constraints and presents the online constraint embedding procedure used to accommodate Euler parameters for the work shown in Chapter 5.

APPENDIX C

CONSTRAINT EMBEDDING

C.1 Constraint Embedding Introduction

This appendix presents the online constraint embedding approach used to enforce the Euler parameter normality constraint in the work presented in Chapter 5. This approach can be used to apply holonomic constraints, such as the Euler parameter constraint, as well as non-holonomic constraints, such as those seen in contact-impact problems [161, 162]. Since mass matrices mapped to Euler parameter vector spaces are non-invertible, either this or some similar method must be used to reduce the mass matrix such that it is invertible. The application of this method to the work in Chapter 5 is similar to the implementation seen in [163].

Numerous methods have been formulated to address the Euler parameter constraint problem. Some common approaches use Lagrange multipliers [164, 165, 166, 167, 168, 169], constraint projection [170, 171], and symbolic constraint embedding [165, 150, 167, 172]. The methods using Lagrange multipliers and constraint projection solve the equations of motion and differential constraint equations simultaneously. These approaches do however lead to errors in the integrated coordinates requiring continuous renormalization to prevent drift [173, 174, 175].

The online approach detailed here is based on the coordinate partitioning method introduced in [165] and reduces the equations of motion to a minimal form such that the constraints are satisfied implicitly during numerical integration [165, 167, 172]. This approach requires identification of a dependent coordinate which may be resolved by singular value decomposition [176, 174, 175], QR decomposition [174, 175, 177], LU factorization [176], or projection [167, 178]. The first three of these methods lead to well conditioned equations but are computationally expensive [167]. While the projection method has a lower computational cost then the others, it requires the inverse of the original matrix which cannot be used with Euler parameters. The online approach has a lower computational cost than the decompositions and does not

require an inversion of the mass matrix for performing the reduction. This method partitions the mass matrix into an invertible subspace using a constraint Jacobian matrix relating the dependent and independent speeds. This matrix subspace corresponds to the reduced set of independent speeds and is invertible. The equations of motion can then be solved for the independent accelerations.

C.2 Constraint Embedding Procedure

Consider again the Euler parameter constraints, given by Eqns. (B.9) and (B.10), which are repeated here:

$$\begin{aligned} e_0^2 \ + \ e_1^2 \ + \ e_2^2 \ + \ e_3^2 \ &= \ 1 \\ e_0\dot{e}_0 \ + \ e_1\dot{e}_1 \ + \ e_2\dot{e}_2 \ + \ e_3\dot{e}_3 \ &= \ 0 \end{aligned} \tag{C.1}$$

Equation (C.1) can be rearranged to solve for one of the speeds in terms of the others. This is useful for defining a specified dependent variable using a constraint Jacobian matrix. Consider the case where e_0 is selected as the dependent variable:

$$\begin{aligned} \dot{e}_0 \ &= \ -\frac{1}{e_0}\left(e_1\dot{e}_1 \ + \ e_2\dot{e}_2 \ + \ e_3\dot{e}_3\right) \\ \dot{e}_0 \ &= \ \begin{bmatrix} -\dfrac{e_1}{e_0} & -\dfrac{e_2}{e_0} & -\dfrac{e_3}{e_0} \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} \end{aligned} \tag{C.2}$$

For a general set of generalized coordinates, Eqn. (C.2) may be represented by:

$$\dot{\mathbf{q}}_p \ = \ A_{pm}\dot{\mathbf{q}}_m \tag{C.3}$$

where $\dot{\mathbf{q}}_p \in \mathbb{R}^{np\times 1}$ and $\dot{\mathbf{q}}_m \in \mathbb{R}^{nm\times 1}$ denotes the dependent and independent generalized coordinates, respectively, where np and nm are the number of dependent and independent coordinates, respectively. Note that $np + nm$ must equal the total number of generalized coordinates. The matrix $A_{pm} \in \mathbb{R}^{np\times nm}$ is the constraint

Jacobian matrix relating the dependent and independent speeds. Note that as the dependent coordinate approaches zero, the matrix A becomes singular. It is therefore critical that the dependent coordinate change throughout a simulation to prevent this singularity. However, each time the dependent coordinate changes, some error is generated. A thorough investigation of the error generated by switching the dependent coordinate at various points in the simulation process is presented in [163].

The vector of all generalized speeds can be easily rearranged to isolate the dependent and independent terms using a permutation matrix P:

$$\begin{bmatrix} \dot{\mathbf{q}}_m \\ \dot{\mathbf{q}}_p \end{bmatrix} \;=\; P^T \dot{\mathbf{q}} \tag{C.4}$$

In the case of a single set of Euler parameters in which e_0 is identified as the dependent variable, the matrix P would be:

$$P \;=\; \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{C.5}$$

such that P^T isolates the dependent and independent speeds and P returns the isolated speeds to their original vector arrangement. Equation (C.3) can also be differentiated to yield the useful expression:

$$\ddot{\mathbf{q}}_p \;=\; A_{pm}\ddot{\mathbf{q}}_m \;+\; \dot{A}_{pm}\dot{\mathbf{q}}_m \tag{C.6}$$

Consider again the equation of motion for a rigid multibody model first shown in Eqn. (2.1) and repeated here:

$$M(\mathbf{q})\;\ddot{\mathbf{q}} \;+\; C(\mathbf{q},\dot{\mathbf{q}}) \;=\; \mathbf{\Gamma}(\mathbf{q},\dot{\mathbf{q}}) \tag{C.7}$$

The permutation matrix P can be used to decompose this equation into the dependent and independent parts:

$$(P^T M P)(P^T \ddot{\mathbf{q}}) \; + \; P^T C \; = \; P^T \boldsymbol{\Gamma}$$

$$\begin{bmatrix} M_{mm} & M_{mp} \\ M_{pm} & M_{pp} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_m \\ \ddot{\mathbf{q}}_p \end{bmatrix} \; + \; \begin{bmatrix} C_m \\ C_p \end{bmatrix} \; = \; \begin{bmatrix} \boldsymbol{\Gamma}_m \\ \boldsymbol{\Gamma}_p \end{bmatrix} \tag{C.8}$$

Recall that linear algebraic expressions are used to represent a set of scalar equations which can always be recovered through decomposition. Equation (C.8) can be shown in a form consistent with Kane's method [150] by decomposing and rearranging:

$$\begin{aligned} \mathbf{F}_m - \mathbf{F}_m^* \; &= \; \boldsymbol{\Gamma}_m - M_{mm}\ddot{\mathbf{q}}_m - M_{mp}\ddot{\mathbf{q}}_p - C_m \; = \; \mathbf{0} \\ \mathbf{F}_p - \mathbf{F}_p^* \; &= \; \boldsymbol{\Gamma}_p - M_{pm}\ddot{\mathbf{q}}_m - M_{pp}\ddot{\mathbf{q}}_p - C_p \; = \; \mathbf{0} \end{aligned} \tag{C.9}$$

The matrix A_{pm} can be used reduce the equations of motion to their minimal form [150]:

$$\widetilde{\mathbf{F}}_m - \widetilde{\mathbf{F}}_m^* \; = \; \mathbf{F}_m - \mathbf{F}_m^* + A_{pm}^T(\mathbf{F}_p - \mathbf{F}_p^*) \; = \; \mathbf{0} \tag{C.10}$$

Inserting Eqn. (C.9) into (C.10) yields:

$$\begin{aligned} \widetilde{\mathbf{F}}_m - \widetilde{\mathbf{F}}_m^* \; &= \; \big[\boldsymbol{\Gamma}_m - M_{mm}\ddot{\mathbf{q}}_m - M_{mp}\ddot{\mathbf{q}}_p - C_m\big] \; + \\ & A_{pm}^T\big[\boldsymbol{\Gamma}_p - M_{pm}\ddot{\mathbf{q}}_m - M_{pp}\ddot{\mathbf{q}}_p - C_p\big] \; = \; \mathbf{0} \end{aligned} \tag{C.11}$$

$$\begin{aligned} \big[\boldsymbol{\Gamma}_m + A_{pm}^T\boldsymbol{\Gamma}_p\big] \; &= \; \big[M_{mm} + A_{pm}^T M_{pm}\big]\ddot{\mathbf{q}}_m \; + \\ & \big[M_{mp} + A_{pm}^T M_{pp}\big]\ddot{\mathbf{q}}_p \; + \; \big[C_m + A_{pm}^T C_p\big] \end{aligned} \tag{C.12}$$

Inserting Eqn. (C.6) into (C.12) yields:

$$\begin{aligned} \big[\boldsymbol{\Gamma}_m + A_{pm}^T\boldsymbol{\Gamma}_p\big] \; &= \; \big[M_{mm} + A_{pm}^T M_{pm}\big]\ddot{\mathbf{q}}_m \; + \\ \big[M_{mp} + A_{pm}^T M_{pp}\big]&\big[A_{pm}\ddot{\mathbf{q}}_m \; + \; \dot{A}_{pm}\dot{\mathbf{q}}_m\big] \; + \; \big[C_m + A_{pm}^T C_p\big] \end{aligned} \tag{C.13}$$

$$\begin{aligned} \big[\boldsymbol{\Gamma}_m + A_{pm}^T\boldsymbol{\Gamma}_p\big] \; &= \; \big[M_{mm} + A_{pm}^T M_{pm} + M_{mp}A_{pm} + A_{pm}^T M_{pp}A_{pm}\big]\ddot{\mathbf{q}}_m \; + \\ & \big[C_m + A_{pm}^T C_p + (M_{mp}\dot{A}_{pm} + A_{pm}^T M_{pp}\dot{A}_{pm})\dot{\mathbf{q}}_m\big] \end{aligned} \tag{C.14}$$

Equation (C.14) can be simplified to:

$$\widetilde{M}_m \ddot{\mathbf{q}}_m \ + \ \widetilde{C}_m \ = \ \widetilde{\mathbf{\Gamma}}_m \tag{C.15}$$

where

$$\begin{aligned} \widetilde{M}_m \ &= \ \left[M_{mm} + A_{pm}^T M_{pm} + M_{mp} A_{pm} + A_{pm}^T M_{pp} A_{pm}\right] \\ \widetilde{C}_m \ &= \ \left[C_m + A_{pm}^T C_p + (M_{mp}\dot{A}_{pm} + A_{pm}^T M_{pp}\dot{A}_{pm})\dot{\mathbf{q}}_m\right] \\ \widetilde{\mathbf{\Gamma}}_m \ &= \ \left[\mathbf{\Gamma}_m + A_{pm}^T \mathbf{\Gamma}_p\right] \end{aligned} \tag{C.16}$$

Equation (C.15) shows the reduced equations of motion. The independent generalized accelerations can then be calculated by inverting the reduced mass matrix $\widetilde{M}_m$. Equation (C.6) can then be used to solve for the dependent generalized accelerations. A vector of these isolated accelerations can then be converted back to the original vector of accelerations for integration using the permutation matrix P:

$$\ddot{\mathbf{q}} \ = \ P \begin{bmatrix} \ddot{\mathbf{q}}_m \\ \ddot{\mathbf{q}}_p \end{bmatrix} \tag{C.17}$$

C.3 Further Comments

Considering the content presented in Appendices A and B and the online constraint embedding approach presented above, the full process of simulating large kinematic systems using recursive dynamics and Euler parameters is now revealed. The recursive algorithm presented in Appendix A is used to numerically calculate vectors and matrices of known quantities for each body in a forward kinematics initial pass along the kinematic chain. These terms are then modified through the back substitution routines that eliminate the unknown reaction forces. This process is repeated until the unknown accelerations for the parent body can be solved for in terms of only known quantities. The generalized accelerations of each body can then be solved for in a final pass along the kinematic chain. If the generalized accelerations for a par-

ticular body include Euler accelerations, the numerically generated mass matrix and vector of known quantities are then converted into the Euler parameter vector space using the approach presented in Appendix B. This conversion extends the dimension of the equation of motion by one and yields a non-invertible mass matrix. The online constraint embedding approach presented in this appendix can then be used to reduce the order of the newly extended equation of motion to a minimal form. Note that this extension-reduction procedure yields an invertible mass matrix that is distinct from the original produced by the recursive algorithm. This reduced equation of motion can then be used to solve for the independent generalized accelerations by inverting the reduced mass matrix.

This forward dynamics process is applied across the full length of the kinematic system to fully solve the equations of motion in a computationally efficient manner yielding small errors in the integration of the Euler parameters. This process is compatible with systems of any scale and any appropriate force and drag model. The recursive algorithm can also be reformulated for prismatic joints. Further computational savings can be achieved using the methods discussed in Chapter 1 such as parallel processing.

APPENDIX D

SIMULATION PARAMETERS

D.1 Simulation Parameters Introduction

This appendix presents the simulation parameters used in the works presented in Chapters 2 - 5. Many of these parameters, such as the Coulomb and Lennard-Jones potential constants for ions are well defined and easily found in research literature and textbooks.

For the EBOV receptor protein polymer modeled in Chapter 2, the primary sequence was taken from an online aggregate database [179, 180]. This primary sequence was used to calculate the total mass of the receptor. The rigid components of the residue side chains are modeled as rigid bodies. The atoms comprising each of these side chain components were modeled in a body attached coordinate frame to identify the location of the center of mass and calculate the rigid body inertia using parallel axis theorem. The primary sequence then established how these residues and side chains were kinematically connected. Initial positions were selected to match structural studies [84]. The theoretical drug particle is modeled as a sphere. Mass and inertia properties are estimated by assuming the drug particle has the same density as the EBOV receptor. The parameters used in Chapter 2 are summarized in Table D.1.

For the molten nitrate salt solution modeled in Chapters 3 and 4, the mass and charge properties of the individual ions are widely available. The trigonal planar nitrate anions are modeled as spheres to eliminate the degrees of freedom associated with rotation to facilitate faster computations. It was assumed this assumption would not affect the testing of the modified thermostat and atomistic multiscale formulations. The mass and inertia properties of the silicon dioxide particle are estimated by assuming a particle density consistent with fine grain commercially available silicon dioxide powder. The charge of this particle is estimated by assuming the individual silicon dioxide molecules become negatively charged when exposed to water molecules

in air during the initial mixture into solution. This charge was then scaled to account for how a cation attracted to a silicon dioxide molecule within the particle would also interact with the adjoining silicon dioxide molecules. The parameters used in Chapters 3 and 4 are summarized in Table D.2.

For the conjugated nanoparticle modeled in Chapter 5, the primary sequence of the GP1b protein was taken from an online aggregate database [149]. This sequence was coarse grained into four sub-bodies. The primary sequence of each coarse grained body was used to calculate the total mass of that body. The inertia properties were found by approximating the bodies as spheres with radii estimated from structural studies [181]. Conformational spring constants were estimated from similar works on coarse grained biopolymers [79]. Potential constants were estimated to approximate the sum of potential interactions between all the atoms within a coarse grained sub-body and a VWF receptor. Nanoparticle size, mass and inertia properties, and conjugated ligand density were estimated from [23]. The parameters used in Chapter 5 are summarized in Tables D.3 and D.4.

Table D.1. Chapter 2 Simulation Parameters

Quantity	Definition	Units	Value Used
	Nanoparticle		
m_n	Mass	zg	0.4
L_n	Characteristic length (radius)	nm	1.5
	Bulk protein		
m_{bp}	-	-	4.6
L_{bp}	-	-	2.5
	Carbon		
m_C	-	-	$1.994 * 10^{-2}$
ϵ_C	Lennard-Jones constant	zg nm^2 ns^{-2}	294
σ_C	Lennard-Jones constant	nm	0.355
	Nitrogen		
m_N	-	-	$2.326 * 10^{-2}$
ϵ_N	-	-	425
σ_N	-	-	0.326
	Hydrogen		
m_H	-	-	$1.674 * 10^{-3}$
ϵ_H	-	-	126
σ_H	-	-	0.257
$AbsErr$	Absolute Error	unitless	10^{-8}
$RelErr$	Relative Error	unitless	10^{-7}

Table D.2. Chapters 3 and 4 Simulation Parameters

Quantity	Definition	Units	Value Used
	Sodium		
m_{Na}	mass	zg	0.03817
c_{Na}	electric charge	C	$+1\ e$
ϵ_{Na}	Lennard-Jones constant	$zg\ nm^2ns^{-2}$	597.794
σ_{Na}	Lennard-Jones constant	nm	0.273
	Potassium		
m_K	-	-	0.06493
c_K	-	-	$+1\ e$
ϵ_K	-	-	694.105
σ_K	-	-	0.305
	Nitrate		
m_{NO_3}	-	-	0.10296
c_{NO_3}	-	-	$-1\ e$
ϵ_{NO_3}	-	-	1500†
σ_{NO_3}	-	-	0.450†
$AbsErr$	Absolute Error	unitless	10^{-7}
$RelErr$	Relative Error	unitless	10^{-6}

† Estimated value

Table D.3. Chapter 5 Simulation Parameters

Quantity	Definition	Units	Value Used
	Nanoparticle		
m_O	Mass	fg	10.72
I_O	Inertia about center of mass	fg μm^2	0.29
$\beta_{v,O}$	Translational damping constant	fg μs^{-1}	$3.36 * 10^3$
$\beta_{\omega,O}$	Rotational damping constant	fg μm^2 μs^{-1}	178.95
L_O	Characteristic length (radius)	μm	0.2
	Body A		
m_A	-	-	$3.91 * 10^{-5}$
I_A	-	-	$6.25 * 10^{-11}$
$\beta_{v,A}$	-	-	33.55
$\beta_{\omega,A}$	-	-	$1.79 * 10^{-4}$
L_A	-	-	$2 * 10^{-3}$
A^*_A	Lennard-Jones constant	fg μm^{14} μs^{-2}	$4.23 * 10^{-27}$
B^*_A	Lennard-Jones constant	fg μm^{8} μs^{-2}	$3.15 * 10^{-11}$
C^*_A	Coulomb constant	fg μm^{3} μs^{-2}	0
k_A	Spring constant	fg μm^2 μs^{-2}	$1 * 10^{-3}$
	Body B		
m_B	-	-	$9.31 * 10^{-6}$
I_B	-	-	$8.38 * 10^{-12}$
$\beta_{v,B}$	-	-	25.16
$\beta_{\omega,B}$	-	-	$7.55 * 10^{-5}$
L_B	-	-	$1.5 * 10^{-3}$
A^*_B	-	-	$2.98 * 10^{-29}$
B^*_B	-	-	$1.31 * 10^{-12}$
C^*_B	-	-	$2.66 * 10^{-6}$
k_B	-		$2 * 10^{-3}$
	Body C		
m_C	-	-	$1.66 * 10^{-5}$
I_C	-	-	$1.04 * 10^{-11}$
$\beta_{v,C}$	-	-	20.97
$\beta_{\omega,C}$	-	-	$4.37 * 10^{-5}$
L_C	-	-	$1.3 * 10^{-3}$
A^*_C	-	-	$6.12 * 10^{-30}$
B^*_C	-	-	$8.02 * 10^{-13}$
C^*_C	-	-	$2.66 * 10^{-6}$
k_C	-		$2 * 10^{-3}$

Table D.4. Chapter 5 Simulation Parameters Continued

Quantity	Definition	Units	Value Used
	Body D		
m_D	Mass	fg	$1.63 * 10^{-5}$
I_D	Inertia about center of mass	fg μm^2	$1.02 * 10^{-11}$
$\beta_{v,D}$	Translational damping constant	fg μs^{-1}	20.97
$\beta_{\omega,D}$	Rotational damping constant	fg μm^2 μs^{-1}	$4.37 * 10^{-5}$
L_D	Characteristic length (radius)	μm	$1.3 * 10^{-3}$
A_D^*	Lennard-Jones constant	fg μm^{14} μs^{-2}	$5.94 * 10^{-30}$
B_D^*	Lennard-Jones constant	fg μm^8 μs^{-2}	$7.79 * 10^{-13}$
C_D^*	Coulomb constant	fg μm^3 μs^{-2}	$2.66 * 10^{-6}$
k_D	Spring constant	fg μm^2 μs^{-2}	$5 * 10^{-3}$
$AbsErr$	Absolute Error	unitless	10^{-6}
$RelErr$	Relative Error	unitless	10^{-5}

REFERENCES

[1] J. Kreuter, "Drug delivery to the central nervous system by polymeric nanoparticles: What do we know?" *Adv. Drug Deliver. Rev.*, vol. 71, pp. 2–14, May 2014.

[2] X. Mou, Z. Ali, S. Li, and N. He, "Applications of magnetic nanoparticles in targeted drug delivery system," *J. Nanosci. Nanotechno.*, vol. 15, no. 1, pp. 54–62, January 2015.

[3] A. Sirelkhatim, S. Mahmud, A. Seeni, N. H. M. Kaus, L. C. Ann, S. K. M. Bakhori, H. Hasan, and D. Mohamad, "Review on zinc oxide nanoparticles: antibacterial activity and toxicity mechanism," *Nano-micro Lett.*, vol. 7, no. 3, pp. 219–242, July 2015.

[4] D. Bobo, K. J. Robinson, J. Islam, K. J. Thurecht, and S. R. Corrie, "Nanoparticle-based medicines: a review of fda-approved materials and clinical trials to date," *Pharm. Res.*, vol. 33, no. 10, pp. 2373–2387, October 2016.

[5] W. Zhou, J. Zhou, Y. Zhou, J. Lu, K. Zhou, L. Yang, Z. Tang, L. Li, and S. Chen, "N-coped carbon-wrapped cobalt nanoparticles on n-doped graphene nanosheets for high-efficiency hydrogen production," *Chem. Mater.*, vol. 27, no. 6, pp. 2026–2032, 2015.

[6] D. Wang, Y. Yu, H. He, J. Wang, W. Zhou, and H. D. Abruna, "Template-free synthesis of hollow-structured co3o4 nanoparticles as high-performance anodes for lithium-ion batteries," *ACS Nano*, vol. 9, no. 2, pp. 1775–1781, 2015.

[7] T. Korten, A. Mansson, and S. Diez, "Towards the application of cytoskeletal motor proteins in molecular detection and diagnostic devices," *Curr. Opin. Biotech.*, vol. 21, no. 4, pp. 477–488, August 2010.

[8] G. Korotcenkov, V. Brinzari, and B. K. Cho, "Conductometric gas sensors based on metal oxides modified with gold nanoparticles: a review," *Microchim. Acta*, vol. 183, no. 3, pp. 1033–1054, March 2016.

[9] H. Wang and M. Pumera, "Fabrication of micro/nanoscale motors," *Chem. Rev.*, vol. 115, no. 16, pp. 8704–8735, August 2015.

[10] X. Ma, A. C. Hortelao, T. Patino, and S. Sanchez, "Enzyme catalysis to power micro/nanomachines," *ACS Nano*, vol. 10, no. 10, pp. 9111–9122, September 2016.

[11] I. Santiago, "Nanoscale active matter matters: Challenges and opportunities for self-propelled nanomotors," *Nano Today*, vol. In Press, 2018.

[12] G. Mittal, V. Dhand, K. Y. Rhee, S.-H. Park, and W. R. Lee, "A review on carbon nanotubes and graphene as fillers in reinforced polymer nanocomposites," *J. Ind. Eng. Chem.*, vol. 21, pp. 11–25, 2015.

[13] F. Masood, "Polymeric nanoparticles for targeted drug delivery system for cancer therapy," *Mater. Sci. Eng. C*, vol. 60, pp. 569–578, March 2016.

[14] S.-E. Park, W. kyu Ko, J. H. Park, M. Bayome, J. Park, D. N. heo, S. H. Lee, J.-H. Moon, K. I. Kwon, and Y.-A. Kook, "Antibacterial effect of silver and gold nanoparticle coated modified c-palatal plate," *J. Nanosci. Nanotechno.*, vol. 16, no. 8, pp. 8809–8813, 2016.

[15] R. Mansell, T. Vemulkar, D. C. M. C. Petit, Y. Cheng, J. Murphy, M. S. Lesniak, and R. P. Cowburn, "Magnetic particles with perpendicular anisotropy for mechanical cancer cell destruction," *Sci Rep*, vol. 7, 2017.

[16] C. Corbo, R. Molinaro, M. Tabatabaei, and O. C. F. M. Mahmoudi, "Personalized protein corona on nanoparticles and its clinical implications," *Biomater. Sci.-UK*, vol. 5, pp. 378–387, 2017.

[17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.

[18] E. Fermi, J. Pasta, and S. Ulam, "Studies of nonlinear problems," Los Alamos Scientific Laboratory, Tech. Rep., 1955.

[19] B. J. Alder and T. E. Wainwright, "Phase transition for a hard sphere system," *J. Chem. Phys.*, vol. 27, 1957.

[20] A. Rahman, "Correlations in the motion of atoms in liquid argon," *Phys Review*, vol. 136, pp. 405–411, October 1964.

[21] K. A. Henzler-Wildman, M. Lei, V. Thai, S. J. Kerns, M. Karplus, and D. Kern, "A hierarchy of timescales in protein dynamics is linked to enzyme catalysis," *Nature*, vol. 450, pp. 913–916, December 2007.

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

[23] L.-C. Su, H. Xu, R. T. Tran, Y.-T. Tsai, L. Tang, S. Banerjee, J. Yang, and K. T. Nguyen, "In situ re-endothelialization via multifunctional nanoscaffolds," *ACS Nano*, vol. 8, no. 10, pp. 10 826–10 836, 2014.

[24] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda," *ACM Q*, vol. 6, no. 2, pp. 40–53, March/April 2008.

[25] J. E. Stone, D. Gohara, and G. Shi, "Opencl: A parallel programming standard for heterogenous computing systems," *IEEE Des. Test*, vol. 12, no. 3, pp. 66–73, May 2010.

[26] R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel o(log(n)) calculation of rigid-body-dynamics. part 1: Basic algorithm," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 867–875, 1999.

[27] ——, "A divide-and-conquer articulated-body algorithm for parallel o(log(n)) calculation of rigid-body-dynamics. part 2: Trees, loops, and accuracy," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 876–892, 1999.

[28] W. Liu, B. Schmidt, G. Voss, and W. Muller-Wittig, "Accerlating molecular dynamics simulations using graphics processing units with cuda," *Comput. Phys. Commun.*, vol. 179, no. 9, pp. 634–641, November 2008.

[29] J. A. Anderson, C. D. Lorenz, and A. Travesset, "General purpose molecular dynamics simulations fully implemented on graphics processing units," *J Comput Phys*, vol. 227, no. 10, pp. 5342–5359, May 2008.

[30] D. Meagher, "Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer," Rensselaer Polytechnic Institute, Tech. Rep. IPL-TR-80-111, 1980.

[31] A. K. Mazur and R. A. Abagyan, "New metholdology for computer-aided modeling of biomolecular structure and dynamics 1. non-cyclic structures," *J. Biomol. Struct. Dyn.*, vol. 6, no. 4, pp. 815–832, 1989.

[32] A. Jain, N. Vaidehi, and G. A. Rodriguez, "A fast recursive algorithm for molecular dynamics simulation," *J. Comput. Phys.*, vol. 106, no. 2, pp. 258–268, 1993.

[33] H. M. Chun, C. E. Padilla, D. N. Chin, M. Watanabe, V. I. Karlov, H. E. Alper, K. Soosaar, K. B. Blair, O. M. Becker, L. S. D. Caves, R. Nagle, D. N. Haney, and B. L. Farmer, "Mbo(n)d: A multibody method for long-time molecular dynamics simulations," *J. Comp. Chem.*, vol. 21, no. 3, pp. 159–184, 2000.

[34] S. Redon and M. C. Lin, "An efficient, error-bounded approximation algorithm for simulating quasi-statics of complex linkages," *Comput. Aided Design*, vol. 38, no. 4, pp. 300–314, 2006.

[35] M. Poursina, K. D. Bhalerao, S. C. Flores, K. S. Anderson, and A. Laederach, "Strategies for articulated multibody-based adaptive coarse grain simulation of rna," *Method Enzymol.*, vol. 487, pp. 73–98, 2011.

[36] G. S. Balaraman, I.-H. Park, A. Jain, and N. Vaidehi, "Folding of small proteins using constrained molecular dynamics," *J. Phys. Chem.*, vol. 115, no. 23, pp. 7588–7596, 2011.

[37] V. K. Gangupomu, J. R. Wagner, I.-H. Park, A. Jain, and N. Vaidehi, "Mapping conformational dynamics of proteins using torsional dynamics simulations," *Biophys. J.*, vol. 104, no. 9, pp. 1999–2008, 2013.

[38] P. Malczyk and J. Fraczek, "Molecular dynamics simulation of simple polymer chain formation using divide and conquer algorithm based on the augmented lagrangian method," *Proc. Inst. Mech. Eng. K*, vol. 229, no. 2, pp. 116–131, 2014.

[39] H. M. Chun, C. E. Padilla, D. N. Chin, M. Watanabe, V. I. Karlov, H. E. Alper, K. Soosaar, K. B. Blair, O. M. Becker, L. S. D. Caves, R. Nagle, D. N. Haney, and B. L. Farmer, "Mbo(n)d: A multibody method for long-time molecular dynamics simulations," *J. Comp. Chem.*, vol. 21, no. 3, pp. 160–184, 2000.

[40] J. Eastwood and R. Hockney, "Shaping the force law in two-dimensional particle-mesh models," *J. Comput. Phys.*, vol. 16, pp. 342–359, 1974.

[41] R. Hockney and J. Eastwood, *Computer Simulation Using Particles*. McGraw Hill, 1981.

[42] M. Deserno and C. Holm, "How to mesh up ewald sums. i. a theoretical and numerical comparison of various particle mesh routines," *J. Chem. Phys.*, vol. 109, no. 18, pp. 7678–7693, 1998.

[43] ——, "How to mesh up ewald sums. ii. an accurate error estimate for the particle-particle-particle-mesh algorithm," *J. Chem. Phys.*, vol. 109, no. 18, pp. 7694–7701, 1998.

[44] M. Poursina and K. S. Anderson, "Long-range force and moment calculations in multiresolution simulations of molecular systems," *J. Comput. Phys.*, vol. 231, pp. 7237–7254, 2012.

[45] ——, "An improved fast multipole method for elctrostatic potential calculations in a class of coarse-grained molecular simulations," *J. Comput. Phys.*, vol. 270, pp. 613–633, 2014.

[46] J. Laflin and K. S. Anderson, "A multibody approach for computing long-range forces between rigid-bodies using multipole expansions," *J. Mech. Sci. Technol.*, vol. 29, no. 7, pp. 2671–2676, 2015.

[47] R. J. Loncharich and B. R. Brooks, "The effects of truncating long-range forces on protein dynamics," *Proteins*, vol. 6, no. 1, pp. 32–45, 1989.

[48] P. J. Steinbach and B. R. Brooks, "New spherical-cutoff methods for long-range forces in macromolecular simulation," *J. Comput. Chem.*, vol. 15, no. 7, pp. 667–683, 1994.

[49] M. Schlosshauer and D. Baker, "Realistic protein–protein association rates from a simple diffusional model neglecting longrange interactions, free energy barriers, and landscape ruggedness," *Protein Sci*, vol. 13, no. 6, pp. 1660–1669, 2004.

[50] A. D. Schuyler and G. S. Chirikjian, "Efficient determinatino of low-frequency normal modes of large protein structures by cluster-nma," *J. Mol. Graph. Model.*, vol. 24, no. 1, pp. 46–58, September 2005.

[51] A. H. Nayfeh, *Perturbation Methods*. Wiley, 1973.

[52] J. D. Coe, B. G. Levine, and T. J. Martinez, "Ab initio molecular dynamics of excited-state intramolecular proton transfer using multireference perturbation theory," *J. Phys. Chem. A*, vol. 111, no. 44, pp. 11 302–11 310, June 2007.

[53] D. Shivakumar, J. Williams, Y. Wu, W. Damm, J. Shelley, and W. Sherman, "Prediction of absolute solvation free energies using molecular dynamics free energy perturbation and the opls force field," *J. Chem. Theory Comput.*, vol. 6, no. 5, pp. 1509–1519, April 2010.

[54] S. Bouzat and F. Falo, "The influence of direct motor–motor interaction in models for cargo transport by a single team of motors," *Phys. Biol.*, vol. 7, no. 4, 2010.

[55] I. Ben-Ari, K. Boushaba, A. Matzavinos, and A. Roitershtein, "Stochastic analysis of the motion of dna nanomechanical bipeds," *B. Math. Biol.*, vol. 73, no. 8, pp. 1932–1951, August 2011.

[56] D.-N. Kim, C.-T. Nguyen, and M. Bathe, "Conformational dynamics of supramolecular protein assemblies," *J. Struct. Biol.*, vol. 173, no. 2, pp. 261–270, February 2011.

[57] S. O. Nielson, C. F. Lopex, G. Srinivas, and M. L. Klein, "Coarse grain models and the computer simulation of soft materials," *J. Phys. Condens. Matter*, vol. 16, no. 15, p. R481, 2004.

[58] A. Y. Shih, P. L. Freddolino, A. Arkhipov, and K. Schulten, "Assembly of lipoprotein particles revealed by coarse-grained molecular dynamics simulations," *J. Struct. Biol.*, vol. 157, no. 3, pp. 579–592, March 2007.

[59] T. Martinetz and K. Schulten, "Topology representing networks," *Neural Netw.*, vol. 7, no. 3, pp. 507–522, 1994.

[60] R. M. Mukherjee and K. S. Anderson, "Efficient methodology for multibody simulations with discontinuous changes in system definition," *Multibody Syst. Dyn.*, vol. 18, pp. 145–168, 2007.

[61] M. Poursina and K. S. Anderson, "Optimization problem and efficient partitioning algorithm for transitions to finer-scale models in adaptive resolution simulation of articulated biopolymers," *Multibody Syst. Dyn.*, vol. 42, no. 1, pp. 97–117, 2018.

[62] M. Fyta, S. Melchionna, S. Succi, and E. Kaxiras, "Hydrodynamic correlations in the translocation of a biopolymer through a nanopore: theory and multiscale simulations," *Phys. Rev. E*, vol. 78, no. 3, 2008.

[63] D. Y. Cheung, B. Duan, and J. T. Butcher, "Current progress in tissue engineering of heart valves: multiscale problems, multiscale solutions," *Expert Opin. Biol. Th.*, vol. 15, no. 8, pp. 1155–1172, 2015.

[64] G. S. Jung and M. J. Buehler, "Multiscale modeling of muscular-skeletal systems," *Annu. Rev. Biomed. Eng.*, vol. 19, no. 1, pp. 435–457, 2017.

[65] K. Kremer and F. Muller-Plathe, "Multiscale simulation in polymer science," *Mol. Simulat.*, vol. 28, no. 8-9, pp. 729–750, 2002.

[66] Q. H. Zeng, A. B. Yu, and G. Q. Lu, "Multiscale modeling and simulation of polymer nanocomposites," *Prog. Polym. Sci.*, vol. 33, no. 2, pp. 191–269, February 2008.

[67] J. Q. Broughton, F. F. Abraham, N. Bernstein, and E. Kaxiras, "Concurrent coupling of length scales: Methodology and application," *Phys. Rev. B*, vol. 60, no. 4, pp. 2391–2403, July 1999.

[68] G. Lu, E. B. Tadmor, and E. Kaxiras, "From electrons to finite elements: A concurrent multiscale approach for metals," *Phys. Rev. B*, vol. 73, no. 2, p. 024108, January 2006.

[69] P. R. Budarapu, R. Gracie, S.-W. Yang, X. Zhuang, and T. Rabczuk, "Efficient coarse graining in multiscale modeling of fracture," *Theoretical and Applied Fracture Mechanics*, vol. 69, pp. 126–143, February 2014.

[70] H. Talebi, M. Silani, and T. Rabczuk, "Concurrent multiscale modeling of three dimensional crack and dislocation propagation," *Advances in Engineering Software*, vol. 80, pp. 82–92, February 2015.

[71] A. Bowling and A. F. Palmer, "The small mass assumption applied to the multibody dynamics of motor proteins," *Journal of Biomechanics*, vol. 42, no. 9, pp. 1218–1223, Jun. 2009. [Online]. Available: http://www.jbiomech.com/issues

[72] M. Haghshenas-Jaryani and A. Bowling, "Multiscale dynamic modeling of processive motor proteins," in *Proceedings of the IEEE International Conference Robotics and Biomimetics (ROBIO)*, December 7-11 2011, pp. 1403–1408.

[73] ——, "Multiscale dynamic modeling of flexibility in myosin v using a planar mechanical model," in *Proceedings of the IEEE International Conference Robotics and Biomimetics (ROBIO)*, December 11-14 2012, pp. 366–371.

[74] M. Haghshenas-Jaryani, J. Drake, N. Tran, A. Bowling, and S. Mohanty, "Multiscale modeling and simulation of a microbead in an optical trapping process," in *Proceedings of the ASME 2nd Global Congress on Nanoengineering for Medicine and Biology (NEMB2013)*, February 4-6 2013, pp. V001T05A003, 2 pages.

[75] M. Haghshenas-Jaryani and A. Bowling, "Multiscale dynamic modeling flexibility in myosin v," in *Proceedings of the ASME International Design Engineering*

Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE), August 4-6 2013, pp. V07AT10A008, 9 pages.

[76] M. Haghshenas-Jaryani, B. Black, S. Ghaffari, J. Drake, A. Bowling, and S. Mohanty, "Dynamics of microscopic objects in optical tweezers: Experimental determination of underdamped regime and numerical simulation using multiscale analysis," *Nonlinear Dynam.*, vol. 76, no. 2, April 2014.

[77] A. Palanki and A. Bowling, "Dynamic model of estrogen docking using multiscale analysis," *Nonlinear Dynam.*, vol. 79, no. 2, January 2015.

[78] A. Bowling and M. Haghshenas-Jaryani, "A multiscale modeling approach for biomolecular systems," *Multibody System Dynamics*, vol. 33, pp. 333–365, April 2015.

[79] M. Haghshenas-Jaryani and A. Bowling, "Modeling flexibility in myosin v using a multiscale articulated multi-rigid body approach," *J. Comput. Nonlin. Dyn.*, vol. 10, no. 1, January 2015.

[80] A. Guy and A. Bowling, "Multiscale modeling of ebola virus glycoprotein," in *ASME NEMB*, Houston, TX USA, February 2016.

[81] (2018, April). [Online]. Available: https://www.ncbi.nlm.nih.gov/Structure/index.shtml

[82] Z. Yang, H. J. Duckers, N. J. Sullivan, A. Sanchez, E. G. Nabel, and G. J. Nabel, "Identification of the ebola virus glycoprotein as the main viral determinant of vascular cell cytotoxicity and injury," *Nat. Med.*, vol. 6, pp. 886–889, 2000.

[83] K. Chandran, N. J. Sullivan, U. Felbor, S. P. Whelan, and J. M. Cunningham, "Endosomal proteolysis of the ebola virus glycoprotein is necessary for infection," *Science*, vol. 308, no. 5728, pp. 1643–1645, June 2005.

[84] J. E. Lee and E. O. Saphire, "Ebolavirus glycoprotein structure and mechanism of entry," *Future Virol.*, vol. 4, no. 6, 2009.

[85] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. M. Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *J. Am. Chem.*, vol. 117, pp. 5179–5197, 1995.

[86] A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus, "All-atom empirical potential for molecular modeling and dynamics studies of proteins," *J. Phys. Chem.*, vol. 102, pp. 3586–3616, 1998.

[87] G. Guennebaud, B. Jacob *et al.*, "Eigen v3.3," http://eigen.tuxfamily.org, November 2016.

[88] L. Fox, *Numerical Solutions of Ordinary and Partial Differential Equations*. Palo Alto: Addison-Wesley, 1962.

[89] J. E. Lee, M. L. Fusco, A. J. Hessell, W. B. Oswald, D. R. Burton, and E. O. Saphire, "Structure of the ebola virus glycoprotein bound to an antibody from a human survivor," *Nature*, vol. 454, pp. 177–182, July 2008.

[90] J. Misasi, M. S. A. Gilman, M. Kanekiyo, M. Gui, A. Cagigi, S. Mulangu, D. Corti, J. E. Ledgerwood, A. Lanzavecchia, J. Cunningham, J. J. Muyembe-Tamfun, U. Baxa, B. S. Graham, Y. Xiang, N. J. Sullivan, and J. S. McLellan, "Structural and molecular basis for ebola virus neutralization by protective human antibodies," *Science*, vol. 351, no. 6279, pp. 1343–1346, 2016.

[91] R. K. Pathria and P. D. Beale, *Statistical Mechanics*. Elsevier, 2011.

[92] L. V. Woodcock, "Isothermal molecular dynamics calculations for liquid salts," *Chem. Phys. Letters*, vol. 10, pp. 257–261, 1971.

[93] H. C. Andersen, "Molecular dynamics simulations at constant pressure and/or temperature," *J. Chem. Phys.*, vol. 72, pp. 2384–2393, 1980.

[94] D. J. Evans, W. G. Hoover, B. H. Failor, B. Moran, and A. J. C. Ladd, "Nonequilibrium molecular dynamics via gauss's principle of least constraint," *Physical Review A*, vol. 28, no. 2, pp. 1016–1021, 1978.

[95] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, "Molecular dynamics with coupling to an external bath," *J. Chem. Phys.*, vol. 81, p. 3684, 1984.

[96] S. Nóse, "A molecular dynamics method for simulations in the canonical ensemble," *Mol. Phys.*, vol. 52, pp. 255–268, 1984.

[97] W. G. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," *Phys. Rev. A*, vol. 31, pp. 1695–1697, 1985.

[98] J. Delhommelle and D. J. Evans, "Configurational temperature thermostat for fluids undergoing shear flow: application to liquid chlorine," *Molecular Physics*, vol. 99, no. 21, pp. 1825–1829, 2001.

[99] C. Braga and K. P. Travis, "A configurational temperature nóse-hoover thermostat," *J. Chem. Phys.*, vol. 123, no. 13, p. 134101, 2005.

[100] G. Bussi, D. Donadio, and M. Parrinello, "Canonical sampling through velocity scaling," *J. Chem. Phys.*, vol. 126, p. 014101, 2007.

[101] K. P. Travis and C. Braga, "Configurational temperature control for atomic and molecular systems," *J. Chem. Phys.*, vol. 128, p. 014111, 2008.

[102] P. K. Patra and B. Bhattacharya, "A deterministic thermostat for controlling temperature using all degrees of freedom," *J. Chem. Phys.*, vol. 140, p. 064106, 2014.

[103] G. U. Gamboa, J. M. Vasquez-Perez, P. Calaminici, and A. M. Koster, "Influence of thermostats on the calculations of heat capacities from born-

oppenheimer molecular dynamics simulations," *Int. J. Quantum Chem.*, vol. 110, pp. 2172–2178, 2010.

[104] H. A. Posch, W. G. Hoover, and F. J. Vesely, "Canonical dynamics of the nóse oscillator: Stability, order, and chaos," *Phys. Rev. A*, vol. 33, no. 6, pp. 4253–4265, 1986.

[105] D. Kusnezov and A. Bulgac, "Canonical ensembles from chaos ii: Constrained dynamical systems," *Annals of Physics*, vol. 214, no. 1, pp. 180–218, 1992.

[106] D. J. Tobias, G. J. Martyna, and M. L. Klein, "Molecular dynamics simulations of a protein in the canonical ensemble," *J. Phys. Chem.*, vol. 97, no. 49, pp. 12 959–12 966, 1993.

[107] A. Cheng and K. M. M. Jr., "Application of the nóse-hoover chain algorithm to the study of protein dynamics," *J. Phys. Chem.*, vol. 100, no. 5, pp. 1927–1937, 1996.

[108] N. Vaidehi, A. Jain, and W. A. Goddard, "Constant temperature constrained molecular dynamics: The newton-euler inverse mass operator method," *Journal of Physical Chemistry*, vol. 100, pp. 10 508–10 517, 1996.

[109] H. A. Posch and W. G. Hoover, "Time-reversible dissipative atractors in three and four phase-space dimensions," *Phys. Rev. E*, vol. 55, no. 6, pp. 6803–6810, 1997.

[110] J. Schnack, "Molecular dynamics investigations on a quantum system in a thermostat," *Physica A*, vol. 259, pp. 49–58, 1998.

[111] D. Ringe and G. A. Petsko, "The 'glass transition' in protein dynamics: what it is, why it occurs, and how to exploit it," *Biophysical Chemistry*, vol. 105, no. 2-3, pp. 667–680, 2003.

[112] W. G. Hoover, C. G. Hoover, H. A. Posch, and J. A. Codelli, "the second law of thermodynamics and multifractal distribution functions: Bin counting, pair

correlations, and the kaplan-yorke conjecture," *Comm. Nonlin. Sci. Num. Sim.*, vol. 12, no. 2, pp. 214–231, 2007.

[113] L. Rui, H. Yuan-Zhong, W. Hui, and Z. Yu-Jun, "Numerical distortion and effects of thermostat in molecular dynamics simulations of single-walled carbon nanotubes," *Chinese Physics B*, vol. 17, no. 11, pp. 4253–4259, 2008.

[114] H. Bosetti, H. A. Posch, and C. Dellago, "Time-reversal symmetry and covariant lyapunov vectors for simple particle models in and out of thermal equilibrium," *Phys. Rev. E*, vol. 82, 2010.

[115] R. A. Shelly, K. Toprak, and Y. Bayazitoglu, "Nóse-hoover thermostat length effect on thermal conductivity of single wall carbon nanotubes," *Acta Mechanica*, vol. 225, pp. 1223–1233, 2014.

[116] M. A. Olson, S. Chaudhury, and M. S. Lee, "Comparison between self-guided langevin dynamics and molecular dynamics simulations for structure refinement of protein loop conformations," *J. Comp. Chem.*, vol. 32, no. 14, pp. 3014–3022, 2011.

[117] T. V. S. Krishnan, J. S. Babu, and S. P. Sathian, "A molecular dynamics study on the effect of thermostat selection on the physical behavior of water molecules inside single walled carbon nanotubes," *J. of Molecular Liquids*, vol. 188, pp. 42–48, 2013.

[118] M. Poursina and K. S. Anderson, "Canonical ensemble simulation of biopolymers using a coarse-grained articulated generalized divide-and-conquer scheme," *Comput. Phys. Commun.*, vol. 184, no. 3, pp. 652–660, 2013.

[119] J. C. Sprott, W. G. Hoover, and C. G. Hoover, "Heat conduction, and the lack thereof, in time-reversible dynamical systems: Generalized nóse-hoover oscillators with a temperature gradient," *Phys. Rev. E*, vol. 89, no. 4, 2014.

[120] J. Li and J. D. Lee, "Reformulation of the nóse-hoover thermostat for heat conduction simulation at nanoscale," *Acta Mechanica*, vol. 225, no. 4-5, pp. 1223–1233, 2014.

[121] A. C. Bránka and K. W. Wojciechowski, "Generalization of nóse and nóse-hoover isothermal dynamics," *Phys. Rev. E*, vol. 62, no. 3, pp. 3281–3292, 2000.

[122] A. C. Bránka, M. Kowalik, and K. W. Wojciechowski, "Generalization of the nóse-hoover approach," *Annals of Physics*, vol. 214, no. 1, pp. 180–218, 1992.

[123] H. Watanabe and H. Kobayashi, "Ergodicity of a thermostat family of the nóse-hoover type," *Phys. Rev. E*, vol. 75, no. 4, 2007.

[124] G. J. Martyna, M. L. Klein, and M. Tuckerman, "Nóse-hoover chains: The canonical ensemble via continuous dynamics," *J. Chem. Phys.*, vol. 97, no. 4, p. 2635, 1992.

[125] W. G. Hoover, K. Aoki, C. G. Hoover, and S. V. D. Groot, "Time-reversible deterministic thermostats," *Physica D: Nonlinear Phenomena*, vol. 187, no. 1-4, pp. 253–267, 2004.

[126] M. Chieruzzi, G. F. Cerritelli, A. Miliozzi, and J. M. Kenny, "Effect of nanoparticles on heat capacity of nanofluids based on molten salts as pcm for thermal energy storage," *Nanoscale Res. Lett.*, vol. 8, no. 1, p. 448, 2013.

[127] J. Seo and D. Shin, "Size effect of nanoparticle on specific heat in a ternary nitrate (lino3–nano3–kno3) salt eutectic for thermal energy storage," *Appl. Therm. Eng.*, vol. 102, pp. 144–148, 2016.

[128] A. Bowling, A. F. Palmer, and L. Wilhelm, "Contact and impact in the multibody dynamics of motor protein locomotion," *Langmuir*, vol. 25, no. 22, pp. 12 974–12 981, Nov. 2009. [Online]. Available: http://pubs.acs.org/toc/langd5/0/0

[129] M. Haghshenas-Jaryani, B. Black, S. Ghaffari, J. Drake, A. Bowling, and S. Mohanty, "Dynamics of microscopic objects in optical tweezers: Experimental determination of underdamped regime and numerical simulation using multiscale analysis," *Nonlinear Dynamics*, vol. 76, no. 2, pp. 1013–1030, April 2014.

[130] E. Faou, "Nóse-hoover dynamics in a shaker," *J. Chem. Phys.*, vol. 124, 2006.

[131] T. Morishita, "From nóse-hoover chain to nóse-hoover network: Design of non-hamiltonian equations of motion for molecular-dynamics with multiple thermostats," *Molecular Physics*, vol. 108, no. 10, pp. 1337–1347, 2010.

[132] A. Sergi and G. S. Ezra, "Bulgac-kusnezov-nóse-hoover thermostats," *Phys. Rev. E*, vol. 81, no. 3, 2010.

[133] M. Esposito, "Nonequilibrium thermodynamics and nóse-hoover dynamics," *J. Phys. Chem. B*, vol. 115, no. 18, pp. 5144–5147, 2011.

[134] T. Megyes, S. Balint, E. Peter, T. Grosz, I. Bako, H. Krienke, and M.-C. Bellissent-Funel, "Solution structure of $nano_3$ in water: Diffraction and molecular dynamics simulation study," *Journal of Physical Chemistry B*, vol. 113, pp. 4054–4064, 2009.

[135] Y. Nagaraj, "Modeling and simulation of nanofluid for heat storage," Master's thesis, University of Texas at Arlington, December 2014.

[136] A. Guy and A. Bowling, "Modification of nóse-hoover thermostat to improve temperature response in molecular simulations," *J. Comput. Nonlin. Dyn.*, vol. 12, no. 3, 2017.

[137] L. Verlet, "Computer "experiments" on classical fluids. i. thermodynamic properties of lennard-jones molecules," *Phys. Rev.*, vol. 159, no. 1, pp. 98–103, 1967.

[138] R. Valencia-Torres, G.-H. Sun, and S.-H. Dong, "Quantum information entropy for a hyperbolical potential function," *Phys. Scripta*, vol. 90, no. 3, 2015.

[139] R. Yakimova, M. Syvajarvi, C. Lockowandt, and M. K. Linnarsson, "Silicon carbide grown by liquid phase epitaxy in microgravity," *J. Mater. Res.*, vol. 13, no. 7, pp. 1812–1815, July 1998.

[140] Y. F. Zhou, J. Y. Xu, Y. Liu, L. D. Chen, Y. Y. Huang, and W. X. Huang, "Influence of microgravity on ce-doped bi12sio20 crystal defect," *B. Mater. Sci.*, vol. 30, no. 3, pp. 211–214, June 2007.

[141] T. Okutani, Y. Kabeya, and H. Nagai, "Thermoelectric n-type silicon germanium synthesize by unidirectional solidification in microgravity," *J. Alloy Compd.*, vol. 551, pp. 607–615, 2013.

[142] R. R. Hantgan, G. Hindriks, R. G. Taylor, J. J. Sixma, and P. G. deGroot, "Glycoprotein ib, von willebrand factor, and glycoprotein iib:iiia are all involved in platelet adhesion to fibrin in flowing whole blood," *Blood*, vol. 76, no. 2, pp. 345–353, 2016.

[143] M. H. Kroll, T. S. Harris, J. L. Moake, R. I. Handin, and A. I. Schafer, "von willebrand factor binding to platelet gpib initiates signals for platelet activation," *J. Clin. Invest.*, vol. 88, no. 5, pp. 1568–1573, 1991.

[144] M. F. Byrne, S. W. Kerrigan, P. A. Corcoran, J. C. Atherton, F. E. Murray, D. J. Fitzgerald, and D. M. Cox, "Helicobacter pylori binds von willebrand factor and interacts with gpib to induce platelet aggregation," *Gastroenterology*, vol. 124, no. 7, pp. 1846–1854, 2003.

[145] S. Kanaji, J. N. Orje, Y. Kamikubo, T. Kanaji, J. Mattson, A. Zarpellon, S. A. Fahs, R. Sood, S. L. Haberichter, Z. M. Ruggeri, and R. R. Montgomery, "Humanized von willebrand factor-glycoprotein ib interaction in mouse models of hemostasis and thrombosis," *Blood*, vol. 128, no. 22, p. 558, 2016.

[146] J. Patzke and E. J. Favaloro, *Laboratory Testing for von Willebrand Factor Activity by Glycoprotein Ib Binding Assays (VWF:GPIb).* Springer, 2017.

[147] K. Tam, S. Cheyyatraviendran, J. Venugopal, A. Biswas, M. Choolani, S. Ramakrishna, A. Bongso, and C.-Y. Fong, "Erratum: A nanoscaffold impregnated with human wharton's jelly stem cells or its secretions improves healing of wounds," *J. Cell. Biochem.*, vol. 118, no. 9, p. 3017, 2017.

[148] L. Lu, M. B. Parmar, M. Kulka, P. Kwan, and L. D. Unsworth, "Self-assembling peptide nanoscaffold that activates human mast cells," *ACS Appl. Mater. Interfaces*, vol. 10, no. 7, pp. 6107–6117, 2018.

[149] (2017). [Online]. Available: https://www.uniprot.org/uniprot/P07359

[150] T. R. Kane and D. A. Levinson, *Dynamics: Theory and Applications*, 1st ed. New York: McGraw Hill, 1985.

[151] M. Haghshenas-Jaryani and A. Bowling, "A new switching strategy for addressing euler parameters in dynamic modeling and simulation of rigid multiboy systems." *Multibody Syst. Dyn.*, vol. 30, no. 2, pp. 185–197, 2012.

[152] A. Guy and A. Bowling, "A multiscale formulation for reducing computation time in atomistic simulations," *J. Comput. Nonlin. Dyn.*, vol. 13, no. 5, 2018.

[153] A. Chatterjee, A. Rodriguez, and A. Bowling, "Analytic solution for planar indeterminate impact problems using an energy constraint," *Multibody Syst. Dyn.*, vol. 42, no. 3, pp. 347–379, March 2018.

[154] W. W. Armstrong, "Recursive solution to the equations of motion for an n-link manipulator," in *Proc. 5th World Cong. on Th. Mach. and Mech. (ASME)*, vol. 2, Montreal, Canada, 1979, pp. 1343–1346.

[155] G. Rodriguez, "Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics," *IEEE Trans. on R&A*, vol. RA-3, no. 6, pp. 624–639, 1987.

[156] W. Schielen, "Computational aspects in multibody system dynamics," *Comput. Method Appl. M.*, vol. 90, no. 1-3, pp. 569–582, 1991.

[157] G. Rodriguez and K. Kreutz-Delgado, "Spatial operator factorization and inversion of the manipulator mass matrix," *IEEE Trans. on R&A*, vol. 8, no. 1, pp. 65–76, 1992.

[158] S. K. Saha, "A decomposition of the manipualator inertia matrix," *IEEE Trans. on R&A*, vol. 2, no. 301-304, April 13.

[159] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2007.

[160] A. P. Bowling, *Vector Mechanics: A Systematic Approach*, 3rd ed. Aqualan Press, LLC, 2017.

[161] S. Goyal, E. N. Pinson, and F. W. Sinden, "Simulation of dynamics of interacting rigid bodies including friction i: General problem and contact model," *Eng. Comput.*, vol. 10, no. 3, pp. 162–174, 1994.

[162] A. A. Barhorst and L. J. Everett, "Contact/impact in hybrid parameter multiple body mechanical systems," *J. Dyn. Syst.-T. ASME*, vol. 117, no. 4, 1995.

[163] M. Haghshenas-Jaryani and A. Bowling, "A new switching strategy for addressing euler parameters in dynamic modeling and simulation of rigid multibody systems," *Multibody Syst. Dyn.*, vol. 30, no. 2, pp. 185–197, 2013.

[164] P. E. Nikravesh and I. S. Chung, "Application of euler parameters to the dynamic analysis of three dinensional constrained mechanical systems," *J. Mech. Des.*, vol. 104, pp. 785–791, 1982.

[165] R. A. Wehage and E. J. Haug, "Generalized coordinate partitioning for dinension reduction in analysis of constrained dynamic systems," *J. Mech. Des.*, vol. 104, pp. 247–255, 1982.

[166] H. S. Morton, "Hamiltonian and lagrangian formulations of rigid body rotational dynamics based on euler parameters," *J. Astronaut. Sci.*, vol. 41, pp. 561–591, 1993.

[167] W. Blajer, W. Schiehlen, and W. Schirm, "A projective criterion to the coordinate partitioning method for multibody dynamics," *Arch. Appl Mech.*, vol. 64, pp. 86–98, 1994.

[168] O. A. Bauchau, "A self-stabilized algorithm for enforcing constraints in multibody systems," *Int. J. Solids Struct.*, vol. 40, no. 13-14, pp. 3253–3271, 2003.

[169] J. Zhang, D. Liu, and Y. Liu, "A constraint violation suppressing formulation for spatial multibody dynamics with singular mass matrix," *Multibody Syst. Dyn.*, vol. 36, no. 1, pp. 87–110, 2016.

[170] R. P. Singh and P. W. Likins, "Singular value decomposition for constrained dynamical systems," *J. Appl. Mech.*, vol. 52, no. 4, pp. 943–948, 1985.

[171] M. Wang and D. P. Bertsekas, "Incremental constraint projection methods for variational inequalities," *Math Program.*, vol. 150, no. 2, pp. 321–363, 2014.

[172] R. B. Gillespie, V. Patoglu, I. I. Hussein, and E. R. Westervelt, "On-line symbolic constraint embedding for simulation of hybrid dynamical systems," *Multibody Syst. Dyn.*, vol. 14, no. 3-4, pp. 387–417, 2005.

[173] M. Kleppmann, "Simulation of colliding constrained rigid bodies," University of Cambridge, Tech. Rep. 683, 2007.

[174] O. A. Bauchau and A. Laulusa, "Review of contemporary approaches for constraint enforcement in multibody systems," *J. Comput. Nonlin. Dyn.*, vol. 3, no. 1, 2008.

[175] A. Laulusa and O. A. Bauchau, "Review of classical approaches for constraint enforcement in multibody systems," *J. Comput. Nonlin. Dyn.*, vol. 3, no. 1, 2008.

[176] C. G. Liang and G. M. Lance, "A differentiable null space method for constrained dynamic analysis," *J. Mech. Transm. Autom. Des.*, vol. 109, pp. 405–411, 1987.

[177] F. A. Potra and J. Yen, "Implicit numerical integration for euler-lagrange equations via tangent space parameterization," *Mech. Struct. Mach.*, vol. 19, no. 1, pp. 77–98, 1991.

[178] W. Blajer, "Elimination of constraint violation and accuracy aspects in numerical simulation of multibody systems," *Multibody Syst. Dyn.*, vol. 7, pp. 265–284, 2002.

[179] (2016). [Online]. Available: http://www.uniprot.org/uniprot/Q05320

[180] (2016). [Online]. Available: http://www.uniprot.org/uniprot/A9QPL9

[181] W. Bergmeier, K. Rackebrandt, W. Schroder, H. Zirngibl, and B. Nieswandt, "Structural and functional characterization of the mouse von willebrand factor receptor gp1b-ix with novel monoclonal antibodies," *Blood*, vol. 95, no. 3, pp. 886–893, 2000.

BIOGRAPHICAL STATEMENT

Ashley Guy was born in Bossier City, Louisiana in 1985. He received a B.S. in Biology from Louisiana Tech University in 2007 and a B.S. and Ph.D. in Mechanical Engineering from The University of Texas at Arlington in 2013 and 2018, respectively. His areas of interest include computational dynamics, robotic systems, and synthetic biology, among others. He enjoys reading, cooking, and slaying dragons.