Doctoral Thesis

Mesoscale Modelling of Cytoplasmic Dynein using Fluctuating Finite Element Analysis

Author:
Benjamin S. Hanson

Supervisors:
Dr. Sarah A. Harris
Dr Daniel J. Read
Dr Oliver G. Harlen

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

in the

University of Leeds

Department of Theoretical Physics
School of Physics and Astronomy

January 2018

Declaration of Authorship

I, Benjamin Hanson, confirm that the work submitted is my own, except where work which has formed part of jointly authored publications has been included. My own contribution and that of the other authors to this work has been explicitly indicated below. I confirm that appropriate credit has been given within the thesis where reference has been made to the work of others.

- The inertialess formulation of FFEA presented in Chapter 3 and the kinetic model developed in Chapter 4 were contributed solely by myself. The software tests mentioned in this thesis, specifically regarding mechanical energies and the steric interpolation function in Chapter 3, were developed and performed by myself, but integrated into the code base as a joint effort of all authors of the following publication:

  A. Solernou, B. Hanson, R. A. Richardson, Rob. Welch, D. J. Read, O. G. Harlen, and S. A. Harris. Fluctuating finite element analysis (FFEA): A continuum mechanics software tool for mesoscale simulation of biomolecules. Submitted following revisions, September 2017

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Signed: _____

Date: _____

"While the strong sharpen their claws, the weak sharpen their minds!"
- Sora, No Game No Life

Acknowledgements

Ah, where do you even start with something like this? Perhaps alphabetically? No, just as people pop into my brain I think.

From the bottom of my heart, I'd like to thank everyone who has ever shared an office with me over the course of my PhD. That currently includes (in clockwise order from me) Steve, Guanghui, Pete, Shaj, Ethan, Sophie, Edgar, Devesh and Helen (and Nikita now too!). Also non / previous office people, Dan, Winke and probably Robin, but I doubt he'll ever read this. There's definitely a 'senpai notice me' joke there! And thanks to all the people in all the other groups, even Astro! And also thanks to those new people who started about a week ago and whose names I keep forgetting. You were the best ones...

I also really need to thank Albert Solernou, who arrived at the perfect time to help make our software usable! Thanks for letting me bounce my ideas off of you each week as well, that must've been more annoying than useful for you. If I ever become a post-doc and get an annoying PhD student, I'll email you and tell you about it!

My supervisory team need many thanks. Sarah Harris, Daniel Read and Oliver Harlen, your combined wealth of knowledge, experience and expertise in all relevant fields for this project were equally, if not more important than your encouragement, support and compliment sandwiches! Let's publish some actual papers now this thesis is done!

Finally, I need to thank my family for their support. Thanks to all of them, but especially to the following four. Thanks to Nikki Kidura, whose continuous encouragement and faith in my ability is what kept me going all these years. Thanks to Gill Hanson, who only said "Have you got a real job yet?" once, and always showed an active interest in the crazy maths stuff. Thanks to Craig Goddard, who's sarcastic view on my work and life choices always made the whole lot much less stressful! And to Eric Kilburn, who was the one who got me properly interested in maths and science in the first place when he taught me about trigonometry all those years ago.

You're all amazing. Oh, and I'm really glad that you aren't Tories as well ;)

UNIVERSITY OF LEEDS

Abstract

Theoretical Physics
School of Physics and Astronomy

Doctor of Philosophy

Mesoscale Modelling of Cytoplasmic Dynein using Fluctuating Finite Element Analysis

by Benjamin S. Hanson

At the forefront of biological experimentation and simulation technology is the attempt to understand the biological mesoscale, the regime in which thermal fluctuations are still vital for function but atomic resolution may no longer be required. There is a wealth of low-resolution biomolecular structural data of macromolecules available for study, and experimental developments are allowing these biomolecules to be visualised to near-atomic resolution without the need for crystallisation. It is clear that a new form of simulation is required to take advantage of this structural data in order to better understand the dynamics of proteins at the biological mesoscale, and their relationship to dynamics at both the microscale and the macroscale.

The work presented in this thesis concerns the development of Fluctuating Finite Element Analysis (FFEA), a mesoscale simulation technique that treats globular macromolecules as visco-elastic continuum objects subject to an additional thermal stress, satisfying our definition of the mesoscale. I have further developed the constitutive continuum model to better represent biological macromolecules, and designed a new solution procedure in order to both increase the computational efficiency of the algorithm and to remove superfluous dynamical information. I also introduce a completely new kinetic framework that couples to the underlying simulation protocol, enabling us to simulate discrete biological events, such as conformational changes, within a continuous dynamical simulation.

I apply FFEA to the molecular motor cytoplasmic dynein, a mesoscopic system exhibiting dynamical features that are beyond the scope of standard molecular dynamics simulations, but well within the mesoscopic regime FFEA was designed for. I determine the physical parameters that an FFEA model of dynein requires for consistency

with both experimental and high-resolution molecular dynamics simulations. Finally, I consider the diffusional properties of dynein with respect to its microtubule track, with the aim of understanding the potential mechanisms that enable the motor to be processive.

Contents

List of Figures

List of Tables

Abbreviations

FFEA	**F**luctuating **F**inite **E**lement **A**nalysis
MD	**M**olecular **D**ynamics
RBC	**R**ed **B**lood **C**ell
DFT	**D**ensity **F**unctional **T**heory
CG	**C**oarse-**G**rain
VMD	**V**isual **M**olecular **D**ynamics
VdW	**V**an **d**er **W**aals
GPU	**G**raphical **P**rocessor **U**nits
QM/MM	**Q**uantum **M**echanics / **M**olecular **M**echanics
GNM	**G**aussian **N**etwork **M**odel
ANM	**A**nisotropic **N**etwork **M**odel
DPD	**D**issipative **P**article **D**ynamics
CFD	**C**omputational **F**luid **D**ynamics
LBM	**L**attice-**B**oltzmann **M**ethod
PDE	**P**artial **D**ifferential **E**quation
ODE	**O**rdinary **D**ifferential **E**quation
FEA	**F**inite **E**lement **A**nalysis
ADP	**A**denosine **D**i**P**hosphate
ATP	**A**denosine **T**ri**P**hosphate
IFT	**I**ntra**F**lagellar **T**ransport
MTBD	**M**icro**T**ubule **B**inding **D**omain
Cryo-EM	**C**ryogenic-**E**lectron **M**icroscopy

Physical Constants

Avogadro's Constant	N_A	=	$6.022\ 140\ 857 \times 10^{23} \mathrm{mol}^{-1}$
Boltzmann's Constant	k_B	=	$1.380\ 648\ 52 \times 10^{-23} \mathrm{J/K}$

Dedicated to Nikki Kidura, who is everything humanity should aspire to be. No pressure...

Chapter 1

Introduction

Biomolecular simulation allows scientists to study static equilibrium conformations and subsequent time evolution of biomolecules as a function of their structure. With much of the small length-scale and fast time-scale dynamical behaviour of biomolecules being out of scope for the experimental methods available today, simulation technology is a necessary counterpart to experimentation, enabling us to understand the higher resolution mechanisms that lead to the lower resolution experimental observations.

The continuous development of computing technology has lead to Moore's Law, the doubling of computational power every 2 years [10], being rigidly followed ever since its conception. Simulation technology has followed a similar trend, and has been under constant development ever since the molecular dynamics (MD) method was first introduced, validated and optimised for use on general purpose computers [11]. In 1967, Levitt was the first to develop generalised force-fields for atomic simulations [12], with immediate applications including calculation of thermodynamic properties and vibrational spectra of n-alkane molecules [13] and structural refinement of macromolecular crystal structures using energy minimisation [14]. These initial insights and applications paved the way for MD to be generalised to any protein with a known structure. As computers have continued to increase in speed, algorithms have been optimised and force-fields have been made more accurate in parallel with similar advances in experimental techniques. However, the methodological approach to biomolecular simulation has remained largely unchanged, with MD still being the 'go to' method for simulations even for the largest proteins with structures available in the Protein Data Bank (PDB) [15].

At the upper limit of biophysics, at the scale of entire organisms, one would not consider the specific motions of each and every atom and electron in a system when describing

its properties. Our experiences with everyday length and time-scales show us that it simply isn't necessary to do so, and Avogadro's constant, $N_A = 6.022 \times 10^{23}\text{mol}^{-1}$, and the associated mole unit of quantity clearly shows it to be an infeasible task. So the question arises: at what length-scale are we able to stop considering the discrete atomic and electronic nature of biomolecules with regards to their function?

1.1 The Spatio-Temporal Regimes of Molecular Biology

We can begin to address the physical relevance of different levels of structural resolution by looking at the length-scale regimes into which well-known biomolecules and biological structures are traditionally categorised, and the physical models used to describe their dynamics.

1.1.1 The Macroscale

The macroscale, as it is traditionally thought of, comprises the length-scales at which whole biological organisms, such as humans, exist. In this regime interactions with the physical world are deterministic, with processes such as jumping, throwing a ball, or even relatively complex activities like fluid flow having completely predictable outcomes given a set of initial conditions[1].

In addition to predictability, the length-scales at the macroscale are so large in comparison to the fundamental interparticle distances that we are able to model physical properties as being spatially continuous throughout the system, as well as continuously varying in time. For this reason, biological systems at the macroscale are often described using continuum mechanics. This allows us to model biomechanical processes such as cardiovascular flow [16], objects such as bones [17] or even hearts [18] and study how these systems respond due to physical interactions. For example, one might consider how the viscosity of blood affects its flow as it approaches an arterial junction [19] or the elastic limit of bone, and subsequent plastic and brittle behaviour i.e. how bones break [17].

[1] Chaotic systems are an exception to this. Although these types of systems have deterministic dynamics that are almost impossible to predict in practice, this is due to a strong dependence on the initial state of the system. We will be dealing with a different kind of uncertainty.

1.1.2 The Nanoscale

The nanoscale refers to the regime in which the discrete nature of atomic structures becomes apparent in the dynamics of systems. At these length-scales, we cannot neglect the stochastic motion caused by thermal fluctuations as it is of the same order of magnitude as the size of the structures of interest. It follows that both spatial and temporal fluctuations in measurable properties at the nanoscale are of equal importance to biological function as their mean value. It is for this reason that we mainly treat biomolecules at the nanoscale using discrete sets of particles, to keep track of the individual particle fluctuations required for biological function.

In contrast with the macroscale, we mainly use discrete classical mechanics to describe the motion and interactions of individual atoms, and statistical mechanics to study the dynamics and probability distributions resulting from their associated ensembles. Using a well-known molecule as an example, consider a starch hydrocarbon polymer and how its length might affect its hydrolysis to glucose [20]. The relative size of thermal fluctuations with respect to the size of the polymer means that over time it will occupy a wide range of physical configurations [21]. The molecule will spend much of its time away from its lowest energy structure and therefore each chemical interaction site may not always be in a conformation that can accommodate the hydrolysis enzyme. In order to study how these conformational variations affect the interaction between the two molecules, we must consider at least an atomic representation of the molecule to keep track of these nanoscale fluctuations. To study the reaction kinetics themselves, we would also need to include the electronic structure variations to determine how the bond energies changes as a function of structural conformation[2].

1.1.3 The Mesoscale

Our definitions of the nanoscale and macroscale are at the extremes of each of the biological length scales of interest, and it is, to some extent, intuitively clear how we should physically view the systems. At the macroscale, thermal fluctuations are negligible and we can use a continuum approximation, and at the nanoscale thermal fluctuations are of the utmost importance in both time and space and require a discrete atomic description. However, the function of many biological systems occur at intermediate length and time-scales, the so-called *mesoscale*, where it is unclear which, if any, of these categories is appropriate.

[2]Starches also have internal interactions with themselves forming granular tertiary structure, which makes this problem much more complex. Our aim here is simply to show the relevance of thermal fluctuations.

We will use blood as an example to illustrate what happens as we transition from the nanoscale to the macroscale. The different possible representations of blood as we increase our length-scale from the nanoscale are shown in Figure 1.1.

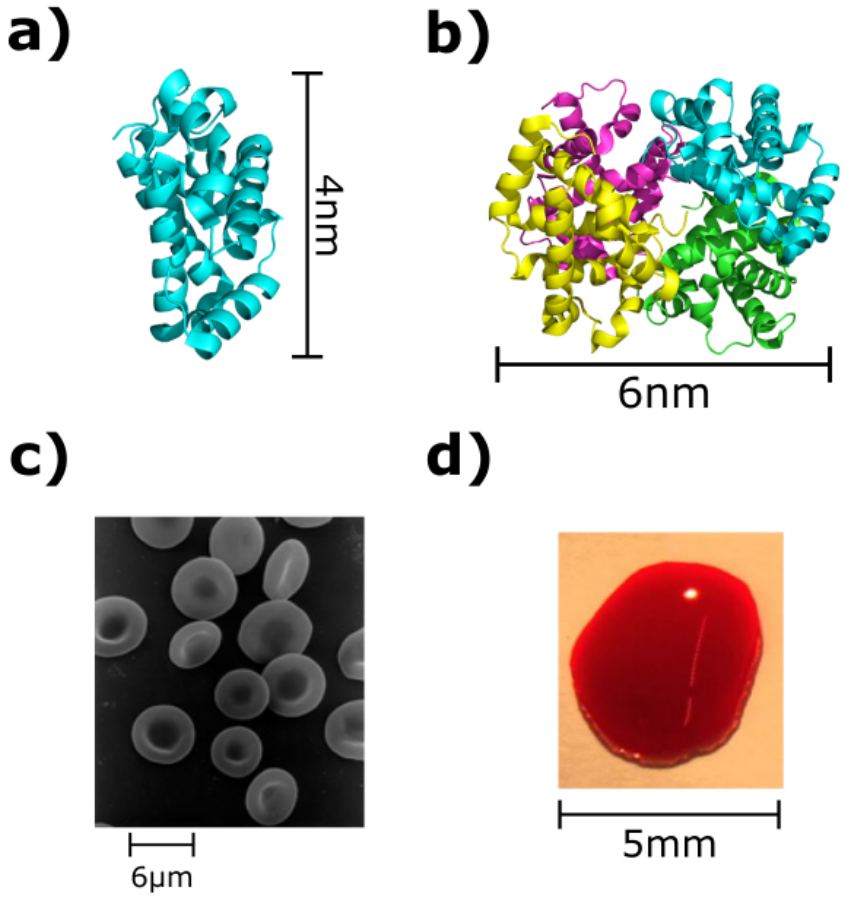


FIGURE 1.1: The structure of blood at various length scales. **a)** A hemoglobin subunit visualised in PyMOL (PDB-ID:5NI1). **b)** A hemoglobin molecule visualised in PyMOL [1]. **c)** An electron micrograph of human red blood cells [2]. **d)** A single droplet of human blood [3].

Figure 1.1a shows a single hemoglobin subunit visualised by its secondary structure. It has been shown that the specific fold responsible for formation of the heme group in hemoglobin (the functional group which binds iron, and subsequently oxygen) is evolutionarily conserved [22]. Therefore, the stochastic thermal motion and varying arrangements of the individual helical structures (known as α-helices) affect how the subunits bind together and so, to understand the assembly of hemoglobin from these subunits, it would be necessary to model this as a discrete atomic system to capture the specific atomic interactions conserved through this fold.

We can then begin to think about the successfully assembled hemoglobin molecule as a whole (Figure 1.1b). Although a similar length-scale to the subunit, the specific function of this quaternary structured protein changes how we might physically understand it. For the oxygen we breath to distribute around the body for respiration, hemoglobin and oxygen molecules must first diffuse together in the alveoli of the lungs.

The diffusion is caused by the thermal fluctuations of the individual molecules of oxygen and those that form hemoglobin. These latter fluctuations ultimately result in the diffusion of the entire of hemoglobin molecule. The characteristic length-scale associated with the diffusion is of the same order of magnitude as the object itself, $\sim 6nm$, which is much greater than inter-atomic distances. Hence, if we are considering the relative abundances of hemoglobin and oxygen, or the mean free time between interactions between the two species, it would be more intuitive to consider a hemoglobin molecule as a single body diffusing through a viscous external medium, removing many 'unnecessary' degrees of freedom from the diffusion problem and hence the removal of atomic resolution for this process. Yet, during the diffusion process, we must at some point consider the atomic nature of the system once again when looking at how an oxygen molecule enters the hemoglobin complex, and even the quantum mechanical effects when considering how that oxygen is then able to bind to the central iron ion. We see that both atomic and mesoscopic length-scales are in play here, depending on which process we wish to model.

Human red blood cells (RBCs) exist at much larger length-scales, functioning as containers and transporters of hemoglobin molecules en masse. They have a well-defined and continuous concave structure [23] (Figure 1.1c) and in addition, lack all but the most essential organelles to maximise space for hemoglobin [24]. Due to the relative lack of discrete internal structure, they are often modelled as continuum objects with associated elasticities and internal viscosities [25]. This enables the deformation of RBCs as they pass through constricted vessels to be modelled as a continuous, elastic process [26]. However, when considering specific interactions at the cell membrane, the dynamics are often atomistic in nature. For RBC transmembrane proteins, such as receptor proteins and ion channels, the dynamics of single ions or small proteins are required to study specific interactions and therefore thermal fluctuations and spatial discretisation are again important. Yet the local curvature around channels and the overall elastic, continuum nature of the structure are also known to affect the function of these proteins. This curvature is generated from a combination of local scaffolding caused by hydrophilic 'coat' proteins and macroscale scaffolding formed by the microtubule and actin networks which span the entire cell [27]. We see that RBCs exhibit both nanoscale and macroscale properties, and even some coupling between the two regimes.

Finally, as the flow generated by the heart overcomes the stochastic thermal effects of diffusion, and the vessel is much larger than the RBC, the collection of RBCs, plasma and other components form whole blood (Figure 1.1d) which on large length scales acts as a deterministic fluid [28]. Although the physical properties of the fluid change

through processes such as coagulation [29], the structure clearly remains continuous and deterministic from simple everyday observations.

It is clear that blood as a general substance does not rigidly fit into the classical nanoscale and macroscale regimes. The different structural components of blood associated with different length-scales each have different associated physics [30]. At the extremes ends, such as nanoscale hemoglobin subunits and macroscale whole blood, it is clear which physical model is appropriate. However, for the intermediate structures, hemoglobin and the whole RBC, the appropriate physical model depends upon the specific process of interest. This spatio-temporal range, where biological systems begin to transition from the nanoscale to the macroscale, is known as the mesoscale, and it is the least understood regime in molecular biology.

To understand how we might model dynamics at the mesoscale, we will first consider a selection of currently available techniques for each of the biological length-scale regimes.

1.2 The Simulation of Biological Systems

At each of the different length-scales of biology, different physical processes dominate the dynamics of the system. With appropriate equations of motion, computational power allows us to simulate the dynamics of biological systems at each of these length-scales. Here we provide an overview of simulation techniques appropriate to each length-scale, and look at what is missing from currently available biological simulation tools.

1.2.1 Quantum Mechanical Methods (Resolution: Sub-nanoscale)

The forces between atoms arise from their associated electron distributions, governed by the Schrödinger equation [31],

$$-\frac{\hbar^2}{2m}\boldsymbol{\nabla}^2\Psi + V(\vec{x},t)\Psi = E\Psi, \tag{1.1}$$

where $V(\vec{x},t)$ is the total potential energy the particles experience, E the total energy and $\Psi = \Psi(\vec{x},t)$ is the wavefunction of the system. The indistinguishability of electons within an atomic network makes solving Equation 1.1 exceedingly difficult in the general case [31], so approximations must be made to elucidate any information.

Density Functional Theory (DFT) is a quantum mechanical method that enables the calculation of physical observables without the explicit evaluation of Equation 1.1

through implementation of the Hohenburg-Kohn (H-K) theorems [32]. These theorems show that the ground-state wavefunction, Ψ_0, can be fully and uniquely defined by the electron density distribution, $n(\vec{x})$, in the ground-state such that $\Psi_0 = \Psi\left[n_0(\vec{x})\right]$ [33], and,

$$n(\vec{x}) = N \int d^3x_1 ... \int d^3x_N \Psi^*(\vec{x}_1, ...\vec{x}_N)\Psi(\vec{x}_1, ...\vec{x}_N), \tag{1.2}$$

where N is the total number of electrons. By treating n as the single variable of interest, we can define each of our physical observables as functionals of this electron density, specifically the energy,

$$\begin{aligned} E &= \langle \Psi\left[n(\vec{x})\right] \left| \hat{T} + \hat{V} + \hat{U} \right| \Psi\left[n(\vec{x})\right] \rangle, \\ \Rightarrow E(n) &= T(n) + V(n) + U(n), \end{aligned} \tag{1.3}$$

where $T(n)$, $V(n)$ and $U(n)$ are the set of specific energy functionals. Numerical minimisation of the total energy functional with respect to n yields n_0, the ground-state density, which uniquely determines every physical observable of the ground-state that can be calculated from E_0 and n_0. Becke provides an in-depth review of the forms of $T(n), V(n)$ and $U(n)$ and the specific operations involved in his perspective [34].

DFT has been used to study a variety of different chemical processes, from adsorption of ligands to chain folding [35]. However, due to the relatively poor scaling of DFT with system size, it is currently restricted to systems with a relatively small number of electrons, $N_e \sim 10^2$, and therefore an even smaller number of atoms. DFT, as with many quantum mechanical calculations, effectively involves eigen-decomposition of a Hamiltonian energy operator which is formally an $O(N_e^3)$ operation. Even with algorithmic optimisation from available software packages, the coupling between electrons can never be fully 'eliminated' from the computation so that at best the cost will scale as $O(N_e^2)$, with the true scaling being somewhere in between. This is clearly infeasible for proteins comprised of many thousands of atoms, and so as we move up in length-scale we will need alternative models.

1.2.2 Molecular Dynamics (Resolution: Nanoscale)

A few simple but significant approximations allow us to reduce the electronic interactions from Section 1.2.1 to much simpler forms. The limit of the Born-Oppenheimer approximation allows us to assume that the reaction of the electron cloud to the motion of the nucleus is not only decoupled, but infinitely fast. We can now assume the nuclei, hereafter referred to simply as atoms, to be point-like particles connected in a covalent network by their electron clouds. The connectivity of this network remains

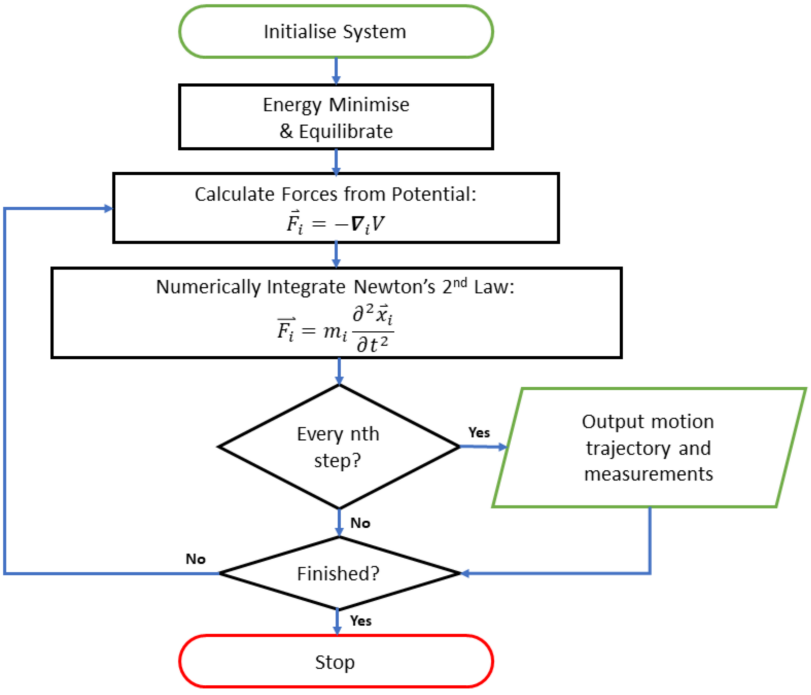


FIGURE 1.2: The overall method of MD. Although developed for atomic systems, this throughput can be applied to any discrete set of particles (see Section 1.2.3).

constant in this framework, and so electronic motion, and therefore chemistry, is not included in the model. The second approximation is the assumption that the interactions of the overall electronic system with any atom, i, can be written as a classical potential energy, $V(\{x_i\}, t)$. We can therefore define Newtonian forces $\vec{F}_i$ on each atom as the gradient of the total potential, $\vec{F}_i = -\boldsymbol{\nabla}_i V$, thus 'removing' the quantum effects from the equations. Applying Newton's Second Law, and integrating numerically the resulting equation of motion for each atom allows us to evolve the system forward in time. This form of simulation is known as Molecular Dynamics (MD).

We can define a general potential energy V_{MD} which can be applied to any biomolecule. The general functional form of the total potential used in MD [36] is,

$$\begin{aligned} V_{MD} = & \sum_{\text{bonds}} k_b(l - \bar{l}_b)^2 \\ & + \sum_{\text{angles}} k_a(\theta - \bar{\theta}_a)^2 \\ & + \sum_{\text{torsions}} \sum_{n} k_\phi \left(1 + cos(n\phi - \gamma_\phi)\right) \\ & + \sum_{i}^{N} \sum_{j=i+1}^{N} \epsilon_{ij} \left[\left(\frac{r_{ij}^{eq}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{ij}^{eq}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}. \end{aligned} \tag{1.4}$$

The first three terms in Equation 1.4 represent the covalent bond interactions, and the last two represent Van der Waals (VdW) and long-range electrostatics. The action of the external solvent is also a necessary part of protein function [37], and so the inclusion of an appropriate water model within MD is required. Water can either be added to the model as explicitly defined molecules which are subject to the general potential Equation 1.4 [38], or implicitly using an approximate coupling to the Poisson-Boltzmann equation [39].

For an MD simulation to be specified for a particular protein, we must define the full set of system parameters that define the specific force response to motion for that protein. These extensional and angular spring-constants, and torsion angles for each possible bond type (C−H, C−C, C=O etc.) in a number of different possible chemical environments have been calculated from either experimentation or higher resolution simulation (DFT, for example) and compiled into MD *force-fields* which define the set of constants $\{k_b, \bar{l}_b, k_a, \bar{l}_a, k_\phi, \gamma_\phi\}$. Software implementations of the MD method then use these force-fields together with Newton's Second Law, integrating the equations of motion to evolve the system forward in time and simulate protein dynamics in general.

Today, MD can be applied to any protein with a known atomic structure. The MoDEL database contains multi-nanosecond trajectories for 1595 proteins, specifically chosen to include the most diverse set of structures possible [40]. More commonly, a simulation suite is developed for a specific molecule. A very relevant example for this thesis are recent MD simulations of a monomer of the molecular motor dynein, generating 200ns simulation trajectories [41]. Using an approximation to increase the speed of the calculation of the long-range electrostatics, the simulations were able to provide evidence that the presence of an adenosine triphosphate (ATP) molecule in place of an adenosine diphosphate (ADP) molecule in the primary binding site increases the flexibility of the molecule. We will be considering these specific simulations in more detail in Chapter 5.

Notice that even these 200ns simulations required the development of a new computational procedure to increase the speed of the MD calculations [41]. For N_a atoms, the algorithms to calculate long-range electrostatic and VdW interactions have a formally $O(N_a^2)$ time-dependence. Using multi-grid techniques and mean field approximations this calculation can be reduced to $O(N_a \ln(N_a))$ for long-range interactions [42], but the close-range electrostatics are still a huge computational bottleneck. The time-integration procedure of MD is also highly constrained, requiring a simulation timestep smaller than the fastest time-scale in the system for numerical stability, which for biological systems is typically $\sim 10^{-15}$s [43]. As with DFT (see Section 1.2.1), the

relatively poor scaling with system size, and timestep limitation, means that MD simulations require infeasible runtimes in order to simulate the dynamics of many proteins over the timescales of their motion. The group of Shaw have been addressing the scalability issue by developing ANTON [44], a special purpose processor designed specifically for running MD simulations. This hardware optimisation has enabled sufficiently long-time simulations of proteins to observe protein conformational changes [45] and is bringing us closer to being able to address the problem of protein folding.

Motivated by the idea of hardware acceleration, recent advances in MD technology have concentrated on integrating MD algorithms with graphical processor units (GPUs). GPU hardware development is driven primarily by the fast growing commercial interactive entertainment sector, making it a sustainable time investment for MD integration. The parallelisation capabilities of GPUs are currently allowing simulations of up to 3 million atoms, limited only by the memory capacity of the GPUs themselves [46].

However, even with the possibility of almost unlimited parallelisation, the inclusion of every single atom in the system may be an unnecessary use of resources when studying the global dynamics of large biological systems [47]. To compensate for this inefficiency, we require coarse-graining methods when simulating larger systems.

1.2.2.1 Quantum Mechanics / Molecular Mechanics

Warshel and Levitt [48] united quantum mechanical methods and molecular dynamics into a hybrid method known as QM/MM. The model treats specific parts of a protein using quantum mechanical methods, but the remainder is studied using lower resolution molecular dyamics. By using an appropriate coupling scheme between the regions [49], QM/MM is able to capture accurate QM properties for chemically active regions whilst still considering the flexibility and electrostatics of the rest of the protein. However, such a method is still limited by the same constraints of regular MD, and so is still not appropriate for large protein simulations.

1.2.3 Coarse-grained Discrete Methods(Resolution:Nano/Mesoscale)

The scaling of the MD method means it often cannot feasibly be applied to the aforementioned 'large' systems at atomistic resolution due to the runtime required. Coarse-grained (CG) methods have been developed to lessen the computational load by reducing the number of degrees of freedom in the system but conserving the general MD method shown in Figure 1.2.

Within large molecules, various intermediate structures are formed as we move up in length-scale: atoms form residues, residues form secondary structure, secondary to tertiary and so on. Each of these intermediaries can be considered as the fundamental unit in a coarse-grained simulation technique using the general MD method. Provided that effective force-fields that describe the interactions between these units can be defined, the MD algorithm is able to produce dynamical trajectories that generate the dynamics of these intermediate structures at a much lower computational cost than all-atom MD. This discrete coarse-graining approach can group arbitrary numbers of atoms and functional groups together, and so naturally there are multiple different CG methods available to be applied to different length-scales. Izvekov *et al.* developed a method which generalises the creation of a CG force-field from higher-resolution structural data [50], but perhaps a more well-known example is the MARTINI model [51].

The MARTINI model uses a potential similar to that of MD, but where the fundamental units are groups of atoms. On average, every 4 heavy atoms are represented as a single CG particle, with specific charged, polar, non-polar and apolar particle 'types', each with additional hydrogen bonding capabilities. The bonded force parameters are determined from the underlying atomic structure, and non-bonded interactions are still treated using a Lennard-Jones potential for Van der Waals and Coulomb electrostatics for charged particles [52]. Once parametrised the CG simulation progresses in a manner identical to MD, as per Figure 1.2.

By reducing the number of degrees of freedom in this fashion we lose dynamical information at smaller length-scales, which often corresponds to the fastest motions in the system. However, as we saw in Section 1.2.2, our computational limitations are due both to the number of particles and the smallest time-scale in the system, and so CG simulations using the MARTINI model provide a substantial speed increase in comparison to all-atom MD. Spatial resolution, then, forms somewhat of a trade-off with simulation speed.

This acceleration has allowed CG MD simulations of much larger systems than all-atom MD. Applications include, but aren't limited to, self-assembly processes, protein conformational change and lipid membrane formation and interactions [52]. A recent study probed the interaction and permeation of particles through a lipid bilayer membrane [53]. The lengths of these simulations enabled comparison with experimental results, which in turn allowed conclusions to be drawn about the underlying mechanisms responsible for experimental observations. For example, Oroskar *et al.* were able to show that a gold nano-particle functionalised with hydrophobic ligands, designed

to transport drug molecules, weakened the membrane following permeation. The hydrophobicity of the nano-particle displaced lipid molecules from the membrane, with larger nano-particles causing permanent dissociation and potentially irreparable damage. Hydrophilic nano-particles on the other hand were still able to permeate through the membrane, yet caused only minor disruption to the membrane with almost immediate recovery of membrane structure once the nano-particle exited. These simulations give evidence that CG MD, specifically those implementing the MARTINI model in this case, capture a significant proportion of the relevant dynamics one would obtain from all-atom MD, but with a much smaller runtime.

1.2.3.1 Gaussian Network Models

Proteins in general can be viewed as a network of 'backbone' carbon atoms with functional groups attached to them. The backbone carbon network, or Cα atoms, are (mostly) responsible for the overall flexibility of the molecules, so the next level of coarse-graining we may consider is including only those Cα atoms in our calculations, and removing electrostatic interactions altogether. A model in which molecular structures are represented as a series of Cα atoms connected by Hookean springs is known as a Gaussian Network Model (GNM) [54]. The simplest CG potential to describe a GNM containing N Cα atoms is significantly less complex than the MARTINI model,

$$V_{GNM} = \frac{1}{2}\sum_{i=0}^{N}\sum_{j=i}^{N}\gamma_{ij}(R_{ij} - R_{ij}^{0})^2, \tag{1.5}$$

where $R_{ij} = R_i - R_j$ is the distance between each pair of atoms in the structure, R_{ij}^0 being the equilibrium distance, and γ_{ij} is the associated stiffness. Here, there is no 'type' defined for each particle, as was the case for MARTINI. Instead, each particle is simply described by its position in space with interactions modelled as pure elastic connections with neighbouring atoms. Equation 1.5 can be rewritten in matrix form,

$$V_{GNM} = \frac{1}{2}\Delta\vec{R}^T\mathbf{K}\Delta\vec{R}, \tag{1.6}$$

where $\mathbf{K}$ is the stiffness matrix of the system, populated with the values γ_{ij} and applied to the vector of deviations from equilibrium $\vec{R}$. A cut-off distance r^c is also specified to simplify the calculation further, such that for $R_{ij}^0 > r^c$, $K_{ij} = 0$. In other words, elastic connections are neglected in the calculations for atoms that are too far away from one another.

The GNM potential is simple enough that we do not need to run a simulation at all to study the resulting dynamics. It can be shown (see Chapter 3) that all of the possible dynamical information of a GNM is contained within the stiffness matrix, **K**. Diagonalisation of **K** gives a series of elastic normal modes. The eigenvectors describe the relative motions of each atom in the mode, and the eigenvalues describe the relative stiffness of each mode.

Although the stiffness matrix is able to incorporate inhomogeneous spring constants γ_{ij} between each pair of Cα atoms (known specifically as an Anisotropic Network Model (ANM)), it is still based on a linearisation of the force-field between atoms. Thus, the eigendecomposition gives only first-order dynamical information regardless of the spring constants chosen. Hence, any long-range or non-linear effects are not included in a GNM. However, GNMs have been suprisingly successful in their analysis of the motion of large proteins. A GNM of the entire ribosome, a $\sim$ 3MDa assembly of 55 protein subunits and 3 rRNA segments, was able to show strong positive and negative correlations between the motions of different regions within the superstructure [55]. Although they verified previous experimental observations, these correlations alone imply a ratchet-like motion that gives an immediate insight into the mechanism by which a segment of RNA may be translated through the ribosome.

1.2.3.2 Dissipative Particle Dynamics

Coarse-grained MD and network models effectively cover the entire range of possible methodologies for coarse-graining large molecules via structural averaging. However, an additional spatially discrete method considers only the overall shape of the system, rather than the underlying atomistic structure.

Dissipative Particle Dynamics (DPD) takes the overall volumetric structure of a biological object and populates that volume with close-packed spherical particles [56]. These spherical particles have no relation to the underlying, higher resolution structure, they are used purely to fill space within the defined volume. On each of these particles we apply a stochastic force, representing the effect of temperature, and a viscous force, representing the internal and external frictional forces. We can also include any conservative force we wish, giving us a general force on any particle i, $\vec{F}_i$, of the form,

$$\vec{F}_i = \sum_{j \neq i} \vec{F}^c_{ij} + \vec{F}^d_{ij} + \vec{F}^t_{ij}, \tag{1.7}$$

where $\vec{F}^c_{ij}$, $\vec{F}^d_{ij}$ and $\vec{F}^t_{ij}$ are the conservative, dissipative and thermal forces respectively.

The thermal and dissipative forces are mathematically coupled through the fluctuation-dissipation theorem, such that the energy within the system conforms to equipartion at equilibrium [21]. These terms are both given generalised forms such that the frictional interaction range and strength can be varied *a priori* without the need for re-derivation of the governing equations. The fluctuation-dissipation theorem will discussed in more detail in Chapter 2.

In addition to this coupling we have the conservative forces, $\vec{F}^C_{ij}$. The form of this force used in DPD is often arbitrary, with the only 'constraint' other than it being conservative is that it remains as a soft potential that inhibits particle overlap.

DPD has a number of advantages compared to single particle models such as simple Brownian or Langevin dynamical models. Unlike single particle models, the total frictional force within DPD is determined as a superposition of interactions between all *pairs* of connected particles. This ensures that Newton's 3rd law is obeyed, such that $\vec{F}_{ij} = -\vec{F}_{ji} \; \forall \; i, j$, and so momentum is explicitly conserved in addition to preseving equilibrium thermodynamics. As a consequence, if we ensure all additional forces are conservative, a simulation of unbound DPD particles converges to the dynamics predicted by the Navier-Stokes equation. DPD, therefore, is a fluctuating fluid dynamical model [57].

While Brownian dynamics have been used to model large, overdamped systems where the relative local viscosity is so large that momentum is dynamically unimportant [58], if the inertial forces are large and the convergence to hydrodynamical behaviour is important, DPD is the appropriate choice. Recalling our earlier example of the red blood cell at the mesoscale, a 2012 study by Li *et al.* [59] used a conservative DPD force to model molecular chirality, and were thus able to observe the self-assembly of sickle hemoglobin into fibres. Following this, they modelled the RBC itself as a DPD system with appropriate conservative forces matching experimental observations, and inserted the fibres inside the RBC. They saw that extended growth of the fibres within the RBC can cause it to form the abnormal half-moon shape associated with sickle cell disease, showing the emergence of a macroscopic observable as a result of underlying fluctuating fluid dynamics.

1.2.4 Continuum Methods (Resolution: Meso/Macroscale)

The final level of discrete coarse-graining, DPD, took us into a new regime for understanding biological dynamics. The method describes the structure of an object by its volume alone, and the dynamics observed in DPD simulations recover macroscopic

observations of fluid dynamics in the long-time limit. This means that DPD in fact belongs to the field of computational fluid dynamics (CFD), which is a sub-field of the much broader topic of continuum mechanics. We now provide a brief introduction to continuum mechanics before describing a selection of simulation techniques that deal explicitly with continuum mechanics from the beginning.

1.2.4.1 An Introduction to Continuum Mechanics

Continuum mechanics differs from discrete mechanics in how it deals with the concept of spatial position. If we consider an atomic system, each individual atom in that system can be assigned its own properties. Each atom, indexed by i, can be assigned scalar properties such as mass, m_i, charge, q_i, and vector properties such as position, $\vec{x}_i$, and any of its time derivatives, $\vec{v}_i$, $\vec{a}_i$. This implies that the space between these particles is effectively empty[3], and so we can discretise the contents of the space into a finite set of points, each with their own individual properties. But what happens when we start to look at bigger systems, or, when the length-scales we are interested in are much larger than the inter-atomic distances?

We can define a characteristic length, l_c, as the minimum length-scale we wish to consider within a given system. As we 'zoom out' and consider larger systems, l_c increases, while the particle density remains constant. It can be shown that,

$$\frac{d}{l_c} \propto \frac{1}{N^{\frac{1}{3}}}, \tag{1.8}$$

where d is the average inter-particle distance and N is the total number of particles in the system. As $N \to \infty$, we make the continuum approximation in which d tends to zero i.e. the object is fully continuous in space.

Making this change to our model means that we can no longer discuss the individual properties of particles. We now effectively have infinite particles separated from each other by infinitesimal distances, and so our position $\vec{x}_i$ is 'promoted' from a property of a particle to a continuous system co-ordinate, i.e. we now discuss the 'average' properties of all atoms located in the vicinity of any continuous position $\vec{x}$. Hence, all of our other properties, $\vec{v}_i$, m_i, q_i etc are no longer discretely indexed, but parametrised by $\vec{x}$, as well as time. Table 1.1 shows how the quantities change as we transition to a continuum description.

[3] With the exception of force-fields.

	Discrete	Continuous
Spatio	$m_i, \vec{v}_i, q_i, \vec{x}_i$	$\rho_m(\vec{x}), \vec{v}(\vec{x}), \rho_q(\vec{x})$
Spatio-Temporal	$m_i(t), \vec{v}_i(t), q_i(t), \vec{x}_i(t)$	$\rho_m(\vec{x}, t), \vec{v}(\vec{x}, t), \rho_q(\vec{x}, t)$

TABLE 1.1: The difference in variable representation between discrete and continuum systems. In contrast with a discrete representation, where each variable is indexed to tie it to the object is represents, a continuum system uses position itself as an additional continuous variable. This means that continuum systems deal with intensive properties like densities, rather than extensive properties like mass and charge.

It follows from this analysis that any measurable property of a continuum system, whether it be a charge distribution, $\rho_q(\vec{x}, t)$, a temperature variation, $T(\vec{x}, t)$ or even the overall shape of the system, $X(\vec{x}, t)$, is now a field that is a function of space and time. The equations which describe the evolution of these properties must necessarily be partial differential equations (PDEs) involving both position and time.

Just as with DPD, in the continuum formalism we can define biological objects simply by their overall shape, as a continuous volume bounded by a surface. Instead of populating the volume with discrete particles however, we derive PDEs that apply over the volume of that object and define the continuous variation of some physical property of interest. For more detail on continuum mechanics in general, Lai *et al.* provide a highly useful introductory text to the subject [60].

We now look at the methods most commonly used for computationally solving PDEs, using the well known one-dimensional diffusion equation as an example,

$$\frac{\partial f}{\partial t} = D\frac{\partial^2 f}{\partial x^2}. \tag{1.9}$$

1.2.4.2 Finite Difference Methods

The finite difference method is a commonly used method for computationally solving partial differential equations [61], due to its relative simplicity and ease of implementation over simple domains. For any function $f(x, t)$ we can use a Taylor expansion to write an approximation to any derivative at any point x and time t using only points from the original function. Equation 1.9 can be approximated as follows,

$$\begin{aligned} \frac{\partial f}{\partial t} &\approx \frac{f(x, t + h_t) - f(x, t)}{h_t}, \\ \frac{\partial^2 f}{\partial x^2} &\approx \frac{f(x + h_x, t) + f(x - h_x, t) - 2f(x, t)}{h_x^2}, \end{aligned} \tag{1.10}$$

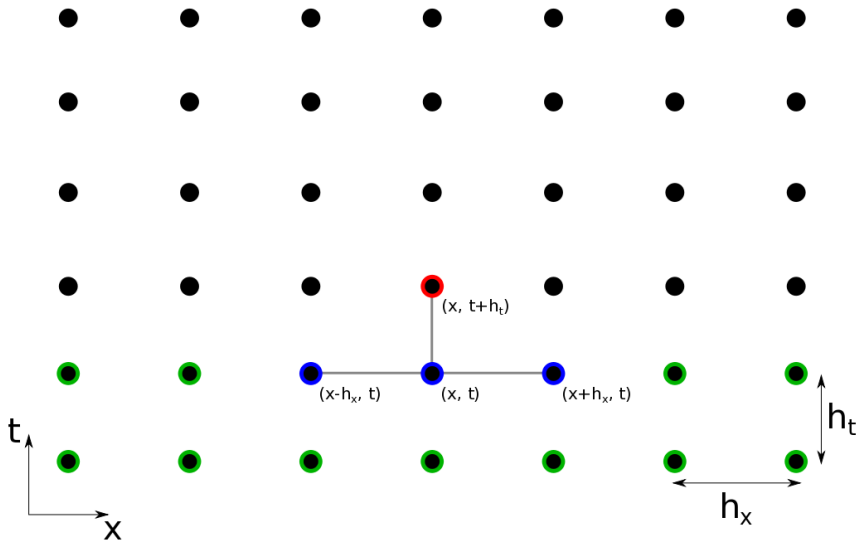


FIGURE 1.3: The grid for a finite difference method with two variables, x and t. Green nodes are those values already known, either through initial conditions or previously calculated. The blue and red nodes form the stencil for the explicit solution of Equation 1.11. Knowledge of the blue values allows the calculation of the red. We can see that if we translate the stencil, a full row of known values would allow the remainder of the time axis of the grid to be calculated.

where h_t and h_x are grid spacings of the variables x and t respectively. Substitution of these types of expansion into any PDE removes the explicit derivative terms, replacing them with multiple evaluations of underlying solution f. For Equation 1.9 this results in,

$$f(x, t + h_t) = f(x, t)\,(1 - 2Dr) + Dr\,(f(x + h_x, t) + f(x - h_x, t))\,, \tag{1.11}$$

where $r = \frac{h_t}{h_x^2}$. This is known as a difference equation, a recursive problem which in this simple example can be solved explicitly and iterated through time using simple computational techniques when given suitable initial and boundary conditions. The two constants h_t and h_x transform the domain of the solution from a continuum into a discrete grid, shown if Figure 1.3.

The finite difference method is a natural approach for the study of physical growth. A recent study by Trivisa *et al.* developed a general model which looked at the growth of tumours by considering the flow of cell density under the action of a transport equation [62]. The model is multi-phasic in that the tumour cells and healthy cells are modelled separately, and the expansion of the tumour is driven primarily by the increased pressure on the healthy cells due to cell division and competition for space. The finite difference approximation to the governing transport and pressure equations

was applied to an initial density distribution with a radial Gaussian profile, and showed that the tumour continued to expand with a circular shape, but quickly became almost homogeneous with respect to density, with a steep decay to zero tumour cell density at the edge.

The weakness of the method however, as we can perhaps tell from Figure 1.3, is that it requires an orthogonal grid. It is therefore not well suited for the complex domain shapes that can exist in higher dimensions.

1.2.4.3 Finite Volume Methods

The finite volume method is an alternative method for the solution of PDEs, and is more suitable for transport equations and conservation equations which involve a flux of some kind, particularly on complex domains [63]. As the name suggests, the finite volume method divides the domain into a set of discrete volume segments, with the solution calculated at the center of each of the segments using the transport though the bounding surfaces of the segments. As a consequence, the finite volume method only requires the total volume to be partitioned into a number of geometrically simple shapes and so can be more easily applied to complex domains.

We begin by discretising the domain of Equation 1.9 into a series of nodes, but this time considering the nodes as the center points of volumetric cells, or line segments in our example (see Figure 1.4). Each cell contains a node, i, and is bounded by partitions defined by the indices $i - \frac{1}{2}$ and $i + \frac{1}{2}$.

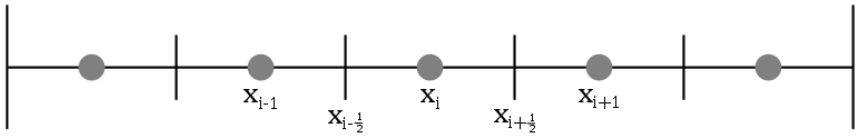


FIGURE 1.4: The discretisation of a 1D line for the finite volume method. Each grey node, i, represents the average solution within the cell bounded by the partitions $i - \frac{1}{2}$ and $i + \frac{1}{2}$. This allows a semi-discrete algebraic equation to be specified at each node. The collection of these equations, plus appropriate boundary conditions, can be solved using standard linear algebra techniques for the solution values at each node, and integrated through time if necessary.

Equation 1.9 is valid over the entire domain, so we can perform a volume integral over the ith cell,

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial f}{\partial t} dx = D \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial^2 f}{\partial x^2} dx. \tag{1.12}$$

We then define the quantity $\bar{f}_i(t)$ as the volume average of $f(x,t)$ in cell i,

$$\bar{f}_i(t) = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f(x,t), \tag{1.13}$$

where $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. This allows us to reverse the order of integration and differentiation on the LHS of Equation 1.12 to obtain,

$$\Delta x_i \frac{\partial \bar{f}_i}{\partial t} = D \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial^2 f}{\partial x^2} dx. \tag{1.14}$$

The RHS of Equation 1.14 can be immediately evaluated in this 1D example,

$$D \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial^2 f}{\partial x^2} dx = \frac{\partial f}{\partial x}\Big|_{i+\frac{1}{2}} - \frac{\partial f}{\partial x}\Big|_{i-\frac{1}{2}}. \tag{1.15}$$

The function $\frac{\partial f}{\partial x} \equiv -J$, where J is the flux associated with the quantity f. Hence,

$$\frac{\partial \bar{f}_i}{\partial t} = \frac{D}{\Delta x_i} \left(J_{i-\frac{1}{2}} - J_{i+\frac{1}{2}} \right). \tag{1.16}$$

The physical interpretation of Equation 1.16 is that the rate of change in the total amount of the quantity f within the volume element i, defined by its volume average $\bar{f}_i$, is equal to the algebraic sum of the amount entering from the left side of the element, $J_{i-\frac{1}{2}}$, and the amount leaving from the right hand side of the element, $J_{1+\frac{1}{2}}$. The use of flux in this manner allows Equation 1.14 to be generalised to any volume element in any spatial dimension through use of the divergence theorem. The values of J, which are spatial gradients of the quantity of interest, can be approximated using differencing schemes similar to those used in the finite difference method. In our 1D example, we can expand the spatial derivatives as linear interpolations between the volume averages defined within each cell. This eventually leads to the final discretised equation for cell i,

$$\frac{\partial \bar{f}_i}{\partial t} = \frac{D}{\Delta x_i \Delta x_{i-\frac{1}{2}}} \bar{f}_{i-1} + \frac{D}{\Delta x_i \Delta x_{i+\frac{1}{2}}} \bar{f}_{i+1} - \frac{D}{\Delta x_i} \left(\frac{1}{\Delta x_{i-\frac{1}{2}}} + \frac{1}{\Delta x_{i+\frac{1}{2}}} \right) \bar{f}_i. \tag{1.17}$$

An equation of the form Equation 1.17 can be written for all values of i. With appropriate boundary conditions for the limiting values of i, this forms a set of algebraic equations for the volume averages $\bar{f}_i$ in each cell, which can be solved using standard linear algebra techniques.

Note that the volumes / widths of each cell, Δx_i, are not necessarily equal. This allows

a greater level of control over the resolution of the method compared to finite difference methods. In a recent publication by Sifounakis *et al.* the finite volume method was applied to the 2D incompressible Navier-Stokes equation to generate a generalised solution regime for any specified initial and boundary conditions [64]. The variability of the cell sizes was used to define 'regions of interest' that were of a much higher resolution than the rest of the grid, thus lowering the overall computational complexity than would be possible with uniform high resolution throughout the system. An initial Gaussian pulse was applied, and velocity and pressure fields calculated and evolved in time until the system equilibrated. The expanding pulse was able to transition into the high resolution region without picking up any erroneous simulation artefacts, with only a small least-squares error compared to the analytical solution for the total grid.

1.2.4.4 Finite Element Methods

The finite element method, or finite element analysis (FEA), is a core topic of this thesis and so we provide a thorough derivation of the method in the following section. FEA is similar to the finite volume method in subdividing the domain into simple geometric shapes, or *elements*, but now uses a polynomial interpolation to define the solution at all points in space.

We begin again by discretising the domain of the equation before considering the equation itself. Within FEA, the subdivision into simple geometric elements can be unstructured, allowing regions of low and high resolution wherever we choose (see Figure 1.5). Algorithms are available to optimise the procedure, efficiently meshing a given domain into a set of tessellating shapes while conforming to user defined constraints [65] i.e. keeping element volumes and shapes within a specified range, explicitly avoiding shapes with too large volumes, minimising the overall loss of volume due to loss of curvature at the boundary, and so on. In three dimensions, tetrahedra are the most common element used as any 3D structure can be subdivided into tetrahedra, with a bounding surface approximated as a set of triangles[4], and due to the large variation in their size and shape, these are broadly referred to as *elements*.

Once we have our discrete computational domain, we can begin to transform the corresponding equation into the appropriate discretised form. Unlike the previous methods we have seen, FEA does not approximate the differential operators. Instead, we seek a solution formed from a restricted set of functions based on the discretisation itself. However, the general complexity of the domain, together with the boundary

[4]Any shape which perfectly tessellates in the dimension of the problem can be used as the FEA mesh building block. Even a combination of different shapes can be used if a tessellation can be formed.

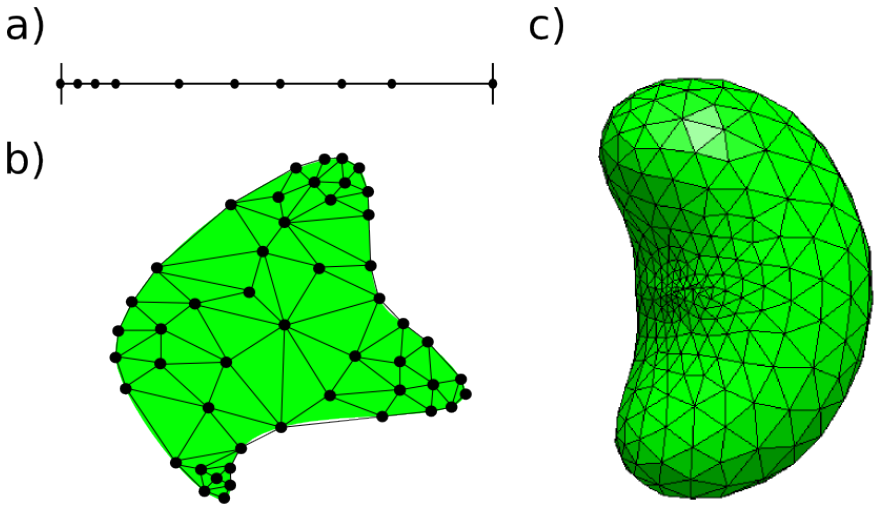


FIGURE 1.5: The discretisation of 1D, 2D and 3D globular structures into finite element meshes. The associated elements are line segments, triangles and tetrahedra respectively. Emphasis is given to the non-uniformity of the element sizes, particularly in the 3D model **c)** where the element density is much higher in the highly curved region to capture fine structural detail. Models **a)** and **b)** were built manually, whereas model **c)** was built using the Blender [4] and Netgen [5] software packages

conditions, may mean that the equation has no exact solution within this restricted set of functions. As such, our overall solution to the specified PDE will necessarily be an approximation of some kind. With this in mind, we use Equation 1.9 as a 1D example and explicitly define the residual, R, of the equation,

$$R = \frac{\partial f}{\partial t} - D\frac{\partial^2 f}{\partial x^2}. \tag{1.18}$$

Given this definition, our aim is to find the function $f(x,t)$ which minimises the absolute value of the residual in some sense. For generality, the finite element method minimises the residual in the *weighted integral* sense, by multiplying Equation 1.18 by a weight function $\omega(x)$ and integrating over the domain V,

$$\int_V \omega R dV = \int_V \omega \left(\frac{\partial f}{\partial t} - D\frac{\partial^2 f}{\partial x^2} \right) dV = 0, \tag{1.19}$$

where Equation 1.19 is known as the weak formulation of Equation 1.9. Note that if we were to choose a functional form for f such that R were identically zero, then the integrand of the RHS of Equation 1.19 must also be zero, and so any solution of Equation 1.9 would also satisfy Equation 1.19. However, the converse is not true.

For our approximation of f, we restrict the space of possible functions to piecewise polynomial interpolations over the elements. Within each element, we approximate the true form of the function $f(x,t)$ as a truncated Taylor expansion, with each element

therefore having its own uniquely associated polynomial which is defined *only* within that element. This set of element equations covers the total domain of Equation 1.9 with no domain overlap, and so f can be defined as the piecewise assembly of the set of element equations. We make the following general approximation for a mesh containing N nodes[5],

$$f(x,t) \approx \sum_{j}^{N} f_j(t)\psi_j(x), \tag{1.20}$$

where $f_j(t)$ are a set of unknown values of $f(x,t)$ evaluated at each of the nodes j, and ψ_j the set of interpolation functions between them, otherwise known as the *shape functions*. Outside of their associated elements, these shape functions are undefined. Within their associated elements, with discrete nodal positions defined as x_j, the shape functions must satisfy the following conditions,

$$\psi_j(x) = \begin{cases} 1 & x = x_j \\ 0 & x = x_i, i \neq j \end{cases}$$

These conditions lead to the required discretisation condition,

$$f(x_j,t) = f_j(t) \tag{1.21}$$

and so, under this approximation, the nodal values in the discretised system are defined to be exactly equal to their counterparts in the original continuous formulation in Equation 1.9.

The values $f_j(t)$ are now a finite set of unknowns in a discretised system, and so to find a uniquely determined set as a solution to Equation 1.19, we require a set of exactly N linearly independent weight functions ω. Each of these can then be substituted into Equation 1.19 to generate an independent algebraic equation, resulting in a set of N algebraic equations in total that form a linear algebra problem with a unique solution.

Provided that the weight functions are at least once differentiable, we can manipulate Equation 1.19 through use of the divergence theorem to transfer a spatial derivative from the RHS onto the weight function,

$$\int_V \omega \frac{\partial f}{\partial t} dV + D \int_V \frac{\partial \omega}{\partial x}\frac{\partial f}{\partial x} dV = D \int_S \omega \frac{\partial f}{\partial x} \hat{n} dS. \tag{1.22}$$

By performing this integration, the solution $f(x,t)$ is now only required to be differentiable once with respect to x, weakening the continuity conditions on the solution through the existence of the weight functions ω (hence the label of 'weak formulation').

[5] Or, in general, N degrees of freedom.

An additional and useful property of this procedure is that, in a similar way to the finite volume method, the use of the divergence theorem on the final term of Equation 1.19 directly evaluates the boundary values as an integration over the bounding surface of the domain, and thus the boundary conditions are explicitly included in the weak form itself. In 1D, we can see from Figure 1.5 that the domain V represents a line segment, and so S is simply the two end-points of the line. For generality, we keep this evaluation in terms of S with $\hat{n}$ as the 1D equivalent of a normal vector, such that $\hat{n} = 1$ at the upper domain limit, and $\hat{n} = -1$ at the lower domain limit.

Substitution of Equation 1.20 into Equation 1.22 gives,

$$\sum_j^N \int_V \omega\psi_j \frac{\partial f_j}{\partial t} dV + D\int_V f_j \frac{\partial \omega}{\partial x}\frac{\partial \psi_j}{\partial x} dV = D\sum_j^N \int_S \omega f_j \frac{\partial \psi_j}{\partial x}\hat{n} dS. \tag{1.23}$$

All that remains is to choose the weight functions. As previously stated, any set of N linearly independent weight functions is sufficient to specify a unique set of $f_j(t)$. Our set has the additional requirement of being at least once differentiable, to satisfy Equation 1.22. The Galerkin formulation of finite element analysis states that, without loss of generality, we can choose the weight functions to be the set of shape functions used to discretise the solution itself [66]. Substitution of $\omega = \psi_i$ into Equation 1.23 gives,

$$\sum_j^N \int_V \psi_i\psi_j \frac{\partial f_j}{\partial t} dV + D\int_V f_j \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} dV = D\sum_j^N \int_S f_j\psi_i \frac{\partial \psi_j}{\partial x}\hat{n} dS, \tag{1.24}$$

which can be rewritten as a linear algebra problem for the set of values f_i at each of the nodes i. In matrix form,

$$\mathbf{A}\frac{\partial \vec{f}}{\partial t} + \mathbf{B}\vec{f} = \vec{F}, \tag{1.25}$$

where,

$$\mathbf{A} = A_{ij} = \int_V \psi_i\psi_j dV,$$
$$\mathbf{B} = B_{ij} = D\int_V \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} dV,$$

and,

$$\vec{F} = F_i = \begin{cases} D\sum_j^N \int_S f_j\psi_i \frac{\partial \psi_j}{\partial x}\hat{n} dS & i \in S \\ 0 & \text{otherwise} \end{cases}$$

So far, we have not specified what specific polynomial form the shape functions have. The most simple form is a linear interpolation between the nodes of each element. However, as long as we add additional nodes within the elements to account for the number of degrees of freedom required for a given order of polynomial, the finite element method can accommodate any set of shape functions over the domain of a single element. In our 1D example, we require $n + 1$ nodes per element for a polynomial of order n; a start and end point (2 nodes) for a linear polynomial (order 1), for example. The strength of the finite element method is that we can increase the accuracy of our approximation of f to as high a level as we wish, either by increasing the element density, which is known a h-refinement, or by increasing the order of the polynomial, called p-refinement [66]. As with any computational increase in accuracy, there is an associated cost in algorithm runtime. In the case of FEA, this is due to the matrices $\mathbf{A}$ and $\mathbf{B}$ increasing in size with both h-refinement and p-refinement. However, unlike many other algorithms, the increase (or decrease) in accuracy is built into the method itself and can be achieved with the same algorithm. We will see the method applied to both simple geometric shapes and the very complex domains of biological molecules in 3D space throughout the remainder of this thesis.

As with the finite difference method, the finite element method has been used to model tumour tissue, specifically heat diffusion throughout the material. A 2009 paper from Lin *et al.* modelled a breast as an ideal hemisphere with 4 radial layers to represent the different tissues [67]. The equation discretised over the mesh was Pennes' bio-heat equation, an empirical biological equivalent of Equation 1.9 for the diffusion of temperature. Each interface between layers constitutes a boundary separating structural discontinuities, the conditions of that can be included in the weak form of the equation. The study was able to show that a tumour embedded in the mesh (effectively forming an extra self-contained layer) generated a measurable thermal fingerprint on the breast surface temperature distribution. From this work, a more recent study used a similar finite element approach to develop a temperature control probe which has the potential to destroy tumour cell through heat death whilst protecting the surrounding healthy cells [68].

Finite element methods have also been used to study mechanical biological structures. Following early models of the mechanical response of (canine) heart tissue to the regular pulsing force [69], more modern work uses varying, but highly controlled, levels of structural resolution to model the repair of the mitral valve within the heart [70]. Computational models of sutures are embedded into the meshes in order to study the effect of different types of suture on the strength of the repair, and whether the mitral valve can sustain the highly localised stress without tearing under the force of a beating

heart. Such modelling is invaluable for surgical techniques as it enables large testing suites of different types of repair without the need for animal testing, vastly lowering the mortality rate of new treatments.

Finite element analysis forms a core part of this thesis for the simulation of biological macromolecules, so for further information on the technique itself and its general applicability, Reddy provides a thorough introductory text [66].

1.3 Thesis Outline

We have introduced a multitude of simulation techniques available to the community that aid us in understanding the function of complex biomolecules. We have seen that many currently available mesoscopic simulation techniques treat systems as networks of spherical beads connected by some form of pair-wise potential. The only real difference between these models is the form of the potential. Yet we have also seen that the mathematics of the macroscale is fundamentally different to that of the microscale, and so that there may be some limit to how far these bead-spring models can help us as we look at larger and larger systems, and the beads no longer represent distinguishable particles within the material. The natural reaction as computational power increases, then, is to rely more on high-resolution simulation techniques and understand mesoscopic dynamical properties as emergent properties. But, a recent review by Gray *et al.* emphasises that "the 'computational microscope' of biomolecular simulation is not infinitely powerful" [47]. They show that the availability of computational resources is insufficient to meet the national demand for high-resolution simulations, and so physical models that are appropriate to the length and time-scales of interest are not only scientifically desirable, but economically essential as well.

CG MD takes us from the 'bottom-up' to the upper limit beyond which bead-spring models are no longer effective. DPD then introduces the concept of overall shape, and populates that shape with loosely connected beads to coincide with macroscale observations. But due to the nature of these beads, the potentials connecting them are somewhat arbitrary. The alternative is to develop a 'top-down' view of the mesoscale, starting from the fluid dynamics and continuum mechanics that already have a fully developed mathematical framework for us to work from. To that end, this thesis presents continued development of a relatively new technique which begins fully in the continuum regime and takes this 'top-down' view of the mesoscale [71].

In this thesis, we present the further development of Fluctuating Finite Element Analysis (FFEA), a novel simulation technique and software package that began development in 2013 with the work of Oliver *et al.* . The technique implements a continuum mechanical approach to model the dynamics of globular biomolecules, but with the inclusion of stochastic thermal effects [71]. Chapter 2 gives an introduction to FFEA and its functionality up to the developmental stage as it was prior to the work presented in this thesis. Chapter 3 then describes the different modifications I have made to the model to allow the simulation of longer time-scales, as well as different forms of simulation. Chapter 4 then looks at an entirely new form of simulation within the FFEA framework which we have called Kinetic FFEA, allowing the modelling and real time simulation of kinetic events in parallel with underlying dynamical models. Finally, Chapter 5 presents an initial study of the molecular motor cytoplasmic dynein, to which we apply Kinetic FFEA in combination with empirical knowledge derived from available experimental evidence in an attempt to determine how, and why, the motor is able to function as a cargo transporter.

Chapter 2

Fluctuating Finite Element Analysis

We saw in Chapter 1 that the current range of mesoscale simulation technologies are mostly based on coarse-grained particle methods that do not fit naturally with the way we describe the macroscale. To make full use of the emerging experimental techniques that allow structure identification at the upper limit of the mesoscale [72], we require a method that maps smoothly onto continuum mechanics as the length scale increases. Those current methods that are available look specifically at viscous fluids. However, Oliver *et al.* (2013) developed a method based on finite element analysis to model proteins as visco-elastic solids [71]. The technique was further developed by Richardson [73] who developed a C++ implementation of the algorithm called Fluctuating Finite Element Analysis (FFEA), a software package designed specifically for the continuum mechanical simulation of large, globular proteins and protein assemblies. These types of protein can have molecular weights ranging from ∼10kDa, approximately that of the hemoglobin molecule we saw in Chapter 1 [74], all the way up to ∼1MDa, approximately the total weight of a fully assembled GroEL chaperonin molecule [75]. System sizes towards the higher end of this range are effectively out of scope for fully atomistic simulations at current computational speeds, at least for readily accesible computational resources [47]. Consequently, many such large proteins have been studied using discrete CG methods. FFEA, though, was not developed as yet another coarse-grained method for simulating atomistic systems. Rather, FFEA originated from the paradigm that the complex properties emerging from large collections of atoms conform more to the mathematical framework of continuum mechanics than discrete Newtonian dynamics. As shown in Chapter 1, these kinds of systems and their interactions with the environment are often *not* spatially discrete at the relevant

length-scales, i.e. they are no longer atomistic in nature, and so we avoid treating them as such.

The mathematics of FFEA is derived from the continuum mechanics formalism, which was introduced in Chapter 1. We progress from there with the idea of our biomolecules of interest being continuum objects.

2.1 A Continuum Equation of Motion

The general formalism of continuum mechanics, further developed from the ideas presented in Section 1.2.4.1, derives a continuum equivalent of Newton's Second Law of motion,

$$\rho \frac{D\vec{u}}{Dt} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \vec{f}, \tag{2.1}$$

where $\vec{u}$ is the velocity field throughout the object, ρ is the object density, $\boldsymbol{\sigma}$ is the total stress, $\vec{f}$ is the external body force and $\frac{D}{Dt}$ is the time derivative in the material frame of reference. Equation 2.1 describes the mechanical evolution of any continuum object with a volume V as its domain, subject to short range forces, which are represented as stresses in a continuum, and long range body forces. Within biomolecular simulation, V represents the volume of a mesoscopic protein or large biomolecule bounded by a molecular surface profile.

Before applying the finite element method, we will consider what form the stress should take to give appropriate behaviour for biological molecules.

2.2 Stresses at the Mesoscale

Experimental studies of proteins in general show that no matter their sequence, proteins often have a preferred equilibrium structure around which they fluctuate [76]. Mathematically, we can describe this using an elastic energy that has a minimum at the equilibrium structure of the object. Within a continuum, the forces that result from this energy are written as an elastic stress. The constitutive model used in FFEA represents a hyperelastic, compressible solid in which the elastic stress is given by,

$$\boldsymbol{\sigma}^e = \frac{G}{|\mathbf{F}|}\mathbf{F}\mathbf{F}^T + \left(K + \frac{G}{3}\right)(|\mathbf{F}| - \alpha)\,\mathbf{I}, \tag{2.2}$$

where G and K are the shear and bulk moduli of the object, $\mathbf{I}$ is the identity matrix and $\mathbf{F}$ is the deformation gradient tensor,

$$\mathbf{F} = \frac{\partial \vec{x}}{\partial \vec{X}}. \tag{2.3}$$

$\mathbf{F}$ is a tensor describing how deformed an object with configuration $\vec{x}$ is from its equilibrium configuration $\vec{X}$. α is an additional constant included to give a stress of zero at $\mathbf{F} = \mathbf{I}$, i.e. in the undeformed state.

The core FFEA constitutive model was chosen to be hyperelastic with experimental evidence in mind. Many previous applications of the standard finite element method to macroscopic biological soft tissues such as ligaments, tendons [77], heart valves [70], and so on, have been successfully modelled using hyperelastic models, with the mitral heart valve model having direct surgical applications. Similar evidence is used to justify our protein compressibility, where explicit water MD simulations have recently been used to show that the volume of the 'typical' protein staphylococcal nuclease has a strong pressure dependence, and some specific temperature dependence [78]. Indeed, topological models have been developed to explain this volumetric change using the underlying atomistic structure and its geometric properties [79].

Also required for any realistic system is a form of viscosity, a term which models the loss of energy to the surroundings. Within FFEA, the viscous properties are modelled as those of a Newtonian fluid, with the viscous stress formed from a linear combination of velocity gradients throughout the system,

$$\boldsymbol{\sigma}^v = \mu \left(\boldsymbol{\nabla}\vec{u} + \boldsymbol{\nabla}\vec{u}^T \right) + \lambda \boldsymbol{\nabla} \cdot \vec{u}\mathbf{I}, \tag{2.4}$$

where μ is the dynamic viscosity and λ is an additional coefficient allowing the definition of a viscosity component specific to bulk deformation, $\mu_{bulk} = \lambda + \frac{2}{3}\mu$. Equation 2.4 means that the viscous stress is proportional to the strain rate within the system, not just the strain itself.

These two stresses are sufficient to model the relaxation properties of a continuum biomolecule. However, the fluctuation-dissipation theorem tells us that with any form of viscous dissipation there must exist an associated stochastic thermal noise, as both phenomena emerge from the same underlying physical process [21]. Hence, for a viscous stress, there must also be an associated thermal stress which is stochastic in nature.

The inclusion of stochastic noise through a thermal stress, $\boldsymbol{\sigma}^t$, is the key component of FFEA and its form was developed by Oliver *et al.* [71] expanding on the work of Landau and Lifshitz [80]. The form of $\boldsymbol{\sigma}^t$ required to give the correct thermodynamic

properties is,

$$\boldsymbol{\sigma}^{\boldsymbol{t}} = \sqrt{\frac{2k_BT}{V\Delta t}}\left(\sqrt{\mu}\mathbf{X} + \sqrt{\lambda}X_0\mathbf{I}\right), \tag{2.5}$$

where k_BT represents the magnitude of the thermal energy within the system via the absolute temperature T, V is the total volume and Δt is the simulation time step. The thermal stress is coupled to the viscous stress through the inclusion of the viscosity parameters μ and λ, and it was shown by Oliver *et al.* that the final FFEA equation of motion obeys the equipartition theorem at equilibrium, with an associated Boltzmann distribution of energy for the system, when the forms of the independent, normalised stochastic variable X_0 and the stochastic tensor $\mathbf{X}$ are as follows,

$$\begin{aligned} \langle X_{ij}\rangle = \langle X_0\rangle &= 0, \\ \langle X_{ij}X_0\rangle &= 0, \\ \langle X_0X_0\rangle &= 1, \\ \langle X_{ij}X_{kl}\rangle &= \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}. \end{aligned} \tag{2.6}$$

The series summation of the components of stress gives us a total stress, $\boldsymbol{\sigma}$,

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^e + \boldsymbol{\sigma}^v + \boldsymbol{\sigma}^t. \tag{2.7}$$

This form of the total stress constitutes a Kelvin-Voigt constitutive model (see Figure 2.1), which enables the system to dissipate energy in such a way that it is always able to return to its equilibrium structure following any deformation, modelling creep behaviour but without retaining any permanent deformation [81]. The elastic stress is a conservative stress which acts to return the continuum to its equilibrium structure. The thermal stress acts to continuously induce deformation which, when coupled to the viscous stress, is consistent with the equipartition theorem. These three stresses together form the mechanical core implemented in the FFEA method.

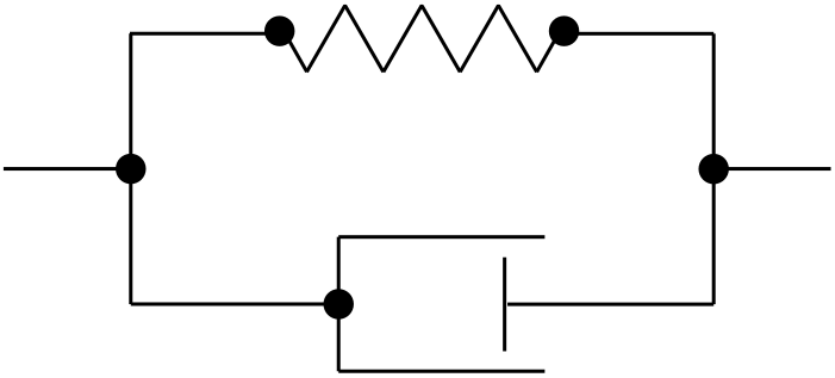


FIGURE 2.1: The standard structural template represented by the Kelvin-Voigt model implemented within FFEA. The component at the top is a spring, representing the elastic component of stress. The bottom component is a dashpot, representing the viscous component of stress. We see here that the components are in parallel, which implies that the stresses are summed in series. In addition, FFEA applies a thermal stress to the system.

2.3 Finite Element Analysis

The potential geometric complexity of biomolecules means that there isn't necessarily an analytical solution for the velocity distribution, the solution to Equation 2.1, throughout the structure. We settle instead for a generalised numerical procedure for which an approximate solution can be obtained via computational techniques.

Finite Element Analysis (FEA) provides this functionality. Applying the procedure shown in Section 1.2.4.4 to Equation 2.1 generates the following algebraic system of equations,

$$\mathbf{M}\frac{D\vec{u}}{Dt} + \mathbf{\Lambda}\vec{u} + \vec{C} = \vec{N} + \vec{f}^{e}, \tag{2.8}$$

where $\mathbf{M}$ is a mass matrix, a function of the density which describes how mass is distributed between elements of the mesh, $\mathbf{\Lambda}$ is a viscosity matrix which describes the dissipation of energy due to relative motion of the nodes within the object, and $\vec{C}$ is an elasticity vector, a non-linear function of the deformation of the object that describes its tendency to return to its mechanical equilibrium. $\vec{N}$ is the thermal noise vector with the required equipartition statistics and $\vec{f}^{e}$ is a vector representing all additional body and external forces. Equation 2.8 describes how a continuum object responds to any applied deformation or external stress.

It must be stated that although FEA brings us back into a discrete solution environment, this is purely for numerical purposes. It is developed from continuum mechanics, which is fundamentally different from discrete particle methods such as MD. It can be shown that as the number of elements $N_e \to \infty$, the analytical continuum solution will be recovered (if such a solution exists) [66], but the motions of individual atoms, residues or any 'underlying structure' will not.

2.4 External and Inter-Molecular Forces

In addition to the internal stresses that describe the mechanical components of a globular biomolecule, FFEA also includes functionality to model the interactions of biomolecules with a surrounding solvent, as well as interactions with other FFEA objects.

2.4.1 External Solvent Interactions

FFEA currently models the interactions with external solvent material as local interactions about each of the discretised nodes following the application of finite element analysis to the original equation. An additional viscous drag force $\vec{F}_i^D$ is applied directly to each node with the following form,

$$\vec{F}_i^D = -(6\pi\mu r_i^s)\vec{u}_i, \tag{2.9}$$

which therefore opposes the motion of the node. Here, r_i^s is an effective Stokes' radius assigned to each node i with velocity $\vec{u}_i$. This force is included as an additional contribution to the viscosity matrix $\mathbf{\Lambda}$,

$$\mathbf{\Lambda} = \mathbf{\Lambda}^{int} + \mathbf{\Lambda}^{ext}, \tag{2.10}$$

where $\mathbf{\Lambda}^{int}$ is formed from the application of the finite element method to the term of Equation 2.8 involving the viscous stress, and the diagonal elements of $\mathbf{\Lambda}^{ext}$ are the elements of the set of vectors $\vec{F}_i^D$ for all nodes i.

By setting the Stokes radii in such a way that they approximate the length scale of the local environment, we can model the correct magnitude of viscous drag on the system. However, there is no fluid flow field defined within the external solvent, it is simply a local effect, and so hydrodynamic communication between different biomolecules is not currently modelled within FFEA.

2.4.2 Van der Waals Interactions

In order to model non-specific, short-range repulsive (and in some cases, attractive) behaviour, FFEA includes a force term which approximates Van der Waals interactions through use of the Lennard-Jones interaction potential. This was implemented by Richardson [73] within the finite element analysis procedure as a surface-surface interaction, using Gaussian quadrature to evaluate the required surface integrals. The total force exerted by the continuous surface Γ_q onto a point $\vec{p}$ can be written as follows,

$$\vec{F}^{int}(\vec{p}) = \int_{\Gamma_q} \vec{f}(\vec{p}, \vec{q}) dA_q, \tag{2.11}$$

where $\vec{q}$ is a point on the surface Γ_q, and $\vec{f}(\vec{p}, \vec{q})$ is a force per unit area per unit area. To implement VdW interactions, $\vec{f}$ takes the Lennard-Jones form,

$$\vec{f}(\vec{p}, \vec{q}) = \frac{12\epsilon}{r_{eq}} \left(\left(\frac{r_{eq}}{r(\vec{p}, \vec{q})} \right)^{13} - \left(\frac{r_{eq}}{r(\vec{p}, \vec{q})} \right)^{7} \right) \hat{\vec{r}}, \tag{2.12}$$

where $r(\vec{p}, \vec{q}) = |\vec{p} - \vec{q}|$, the distance between the two points $\vec{p}$ and $\vec{q}$, $\hat{\vec{r}}$ is the unit vector corresponding to these points and ϵ and r_{eq} are the equilibrium interaction strength and separation respectively. A further integration of $\vec{F}^{int}$ over the surface Γ_p containing $\vec{p}$ gives the total force between the two surfaces. We can use this in conjunction with the finite element method to discretise these continuous surface effects onto the surface nodes of the mesh. These forces then form part of the external force term $\vec{f}^e$ from Equation 2.8.

Within FFEA, different instances of the parameters ϵ and r_{eq} can be assigned to the different surface faces, allowing multiple different interaction potentials between the two surfaces, all of which correspond to VdW type interactions. As well as this, the discretisation of the surfaces into a set of triangles means that Equation 2.11 can be easily modified to allow surface-surface interactions within the same molecule, if necessary. More detail on this method is given by Richardson [73].

2.5 Fluctuating Finite Element Analysis

Following application of the finite element method, the inclusion of all additional solvent and inter-molecular interactions and the addition of the new thermal stress term,

we obtain the following equation of motion, written in index notation for clarity,

$$M_{ij}\ddot{x}_j + \Lambda_{ij}\dot{x}_j + C_i = N_i + f_i^e, \tag{2.13}$$

where the indices i and j run over both the nodes and directional components.

Equation 2.13 is the equation of motion used in FFEA simulations prior to the work presented in this thesis. With M_{ij}, Λ_{ij} and C_i evaluated at the beginning of each simulation time step, standard linear algebra and numerical integration techniques can be applied to change the velocity field of the system by a small amount Δv_i, and update the configuration of the system by a small amount Δx_i, over a time step Δt.

The vector $\vec{N}$ in the final discretised model has the effect of applying random forces to each node such that each degree of freedom has $\frac{1}{2}k_BT$ of energy on average, with Gaussian fluctuations about that value. With a time step, Δt, the vector components N_i have the following covariance coupling to the viscosity matrix Λ_{ij},

$$\langle N_i N_j \rangle = \frac{k_B T}{\Delta t} \left(\Lambda_{ij} + \Lambda_{ji} \right), \tag{2.14}$$

which shows implies that the thermal noise is delta-correlated in time, but also in space. This is due to the spatial discretisation of the solution from the finite element method, leading to the matrix $\mathbf{\Lambda}$ being block-diagonal. This lack of global viscous coupling allows the FFEA calculations to be performed on an 'element by element' basis[1], enabling an approximately linear scaling of simulation speed with system size, whilst still preserving the required local coupling between the elements and the correct statistical behaviour of the stochastic noise with respect to the viscosity.

2.6 Summary

We have introduced Fluctuating Finite Element Analysis, a novel and robust method for the simulation of globular macromolecules. The method treats these molecules as continuous soft matter systems, with a visco-elastic response to thermally induced deformation. Inter-particle forces are implemented to model both steric repulsion and Van der Waals type forces.

Tests performed on simple continuum structures by Oliver *et al.* [71] show that the technique successfully equilibrates to the correct distribution of energy, in line with

[1] Computationally, each element can treated as independent, with no pair-wise, long-range coupling to account for.

the equipartition theorem, and gives the correct thermodynamic properties for simple test systems. Yet as with any emerging technique, there is room for improvement.

The following chapter will discuss the developments I have made to the FFEA software package, including additional algorithmic functionality within the software to allow a much broader range of length and time scales to be accessed by a potential user. These improvements will prove to be vital for the applications of FFEA to the molecular motor cytoplasmic dynein in Chapter 5.

Chapter 3

Development of Fluctuating Finite Element Analysis

As a continuum mechanical method, finite element analysis applied to the Cauchy momentum equation naturally applies to macroscopic systems. FFEA extends this to simulations of mesoscopic systems through the inclusion of a stochastic stress, which produces thermal fluctuations that naturally reduce in relative magnitude as the system size increases. However, the *general* forms of dynamic motion which can result from the application of FFEA to biological systems have yet to be analysed in detail.

This chapter presents an analysis of the different types of stress within FFEA, and how they combine to produce different regimes of motion. We develop an inertialess solution protocol for FFEA simulations, which takes into account the over damped nature of motion specific to the mesoscale, and discuss the advantages of both solution methods and in what circumstances each should be used. Using this analysis, we also develop two network models for FFEA systems, similar in nature to Gaussian Network Models, which can determine the relative propensities of different normal modes of motion within specific systems. Finally, we optimise the surface-surface interactions within FFEA to obtain a robust transition between steric repulsion and Van der Waals attraction without affecting the computational efficiency of an FFEA simulation.

3.1 A New Constitutive Model

The continuum mechanical equations of motion within FFEA take into account all forces in the form of stresses. We saw in Chapter 2 that the total mechanical stress,

$\boldsymbol{\sigma}$, is formed as a series summation of all of the individual components [71],

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^e + \boldsymbol{\sigma}^v + \boldsymbol{\sigma}^t, \tag{3.1}$$

where the components of stress are elastic, viscous, and thermal. Here we consider the current state of the elastic stress, $\boldsymbol{\sigma}^e$.

3.1.1 Elastic Stress - Current Formulation

An elastic stress can be calculated from a strain energy density functional, ψ, to model the material as hyperelastic (see Section 2.2). The hyperelastic property within FFEA means that the stress-strain relationship for biomolecules is non-linear with respect to deformation from the equilibrium configuration, and the stress is independent of the strain rate.

ψ can take a variety of functional forms depending on the properties of elasticity to be modelled. The original form developed and implemented within FFEA, ψ_o, is,

$$\psi_o = \frac{G}{2}\left(F_{ij}F_{ij} - 3\right) + \frac{B}{2}\left(\left(|\mathbf{F}| - \alpha\right)^2 - \left(\frac{G}{B}\right)^2\right), \tag{3.2}$$

where G is the shear modulus, $K = B - \frac{G}{3}$ is the bulk modulus and $\mathbf{F}$ is the deformation gradient tensor. The first term in Equation 3.2 is drawn from the neo-Hookean model for incompressibility [82] and the second term adds energy proportionally to fractional volumetric change. The addition of a constant, α, then gives both zero elastic stress and zero elastic energy at zero deformation. We saw in Chapter 2 that the form of stress derived from this functional, now given in index notation, is,

$$\sigma^e_{ij} = \frac{G}{|\mathbf{F}|}F_{ik}F^T_{kj} + B(|\mathbf{F}| - \alpha)\delta_{ij}. \tag{3.3}$$

For $\boldsymbol{\sigma}^e = \mathbf{0}$ at $\mathbf{F} = \mathbf{I}$, it follows that $\alpha = 1 + \frac{G}{B}$.

This formulation was developed for small fluctuations about an equilibrium structure, for which previous uses of the model have worked well [83]. However, recent uses of FFEA have required smaller elements to capture structural detail $\leq$ 1nm, which is near the lower limit of the mesoscale. This means that relative deformations, and therefore the strains within these elements, are much larger than in previous applications. We can examine the effect of large, compressive strains by considering pure isotropic

deformation, such that $F_{ij} = \zeta\delta_{ij}$ then Equation 3.2 simplifies to,

$$\psi_o(\zeta) = \frac{3G}{2}\left(\zeta^2 - 1\right) + \frac{B}{2}\left(\left(\zeta^3 - \alpha\right)^2 - \left(\frac{G}{B}\right)^2\right), \tag{3.4}$$

which has a single variable, ζ. Figure 3.1 shows the effect of isotropic expansion on the current strain energy density for $G = 1$GPa and $K = 3$GPa. We can see that for small fluctuations about equilibrium and indeed at large expansion the model has an acceptable form, being approximately harmonic about $\zeta = 1$ and tending to infinity as $\zeta \to \infty$. However, we may expect that for large compressions ($\zeta \to 0$), the strain energy density would also tend towards infinity. The original model does not provide this behaviour, rather, $\psi_0(0) = \frac{1}{2}\left(K - \frac{2G}{3}\right) = \frac{1}{2}\lambda$, where λ is an additional material property known as Lamé's first parameter. Additionally, we chose our values of G and K to emphasise that, depending on the material parameters, turning points can form in the region $0 < \zeta < 1$. In our example, the strain energy density actually begins to decrease with increasing compression, or, for decreasing ζ at $\zeta < 0.506$. This is an unrealistic model for large compressions of an isotropic and homogeneous material such as this. Although the resulting stress does give the required tendency to infinity, this is only a result of the compressibility of the object, with a fractional volumetric change determined by $|\mathbf{F}| = \frac{V}{V_0}$, and not due to the intrinsic energy storage ability of the system itself. We require a more robust functional form for ψ to suitably model large compressive deformations.

3.1.2 Elastic Stress - New Formulation

Shontz *et al.* [84] provide a strain energy density which is valid for large compressional deformations as well as expansion. The form is as follows,

$$\psi_n = \frac{G}{2}\left(F_{ij}F_{ij} - 3\right) + \frac{\lambda}{4}\left(|\mathbf{F}|^2 - 1\right) - \left(\frac{\lambda}{2} + G\right)\ln(|\mathbf{F}|). \tag{3.5}$$

From this, we can calculate the Cauchy stress tensor required by FFEA using the following relation [82],

$$\sigma^e_{ij} = \frac{1}{|\mathbf{F}|}\frac{\partial\psi}{\partial F_{ik}}F_{jk}. \tag{3.6}$$

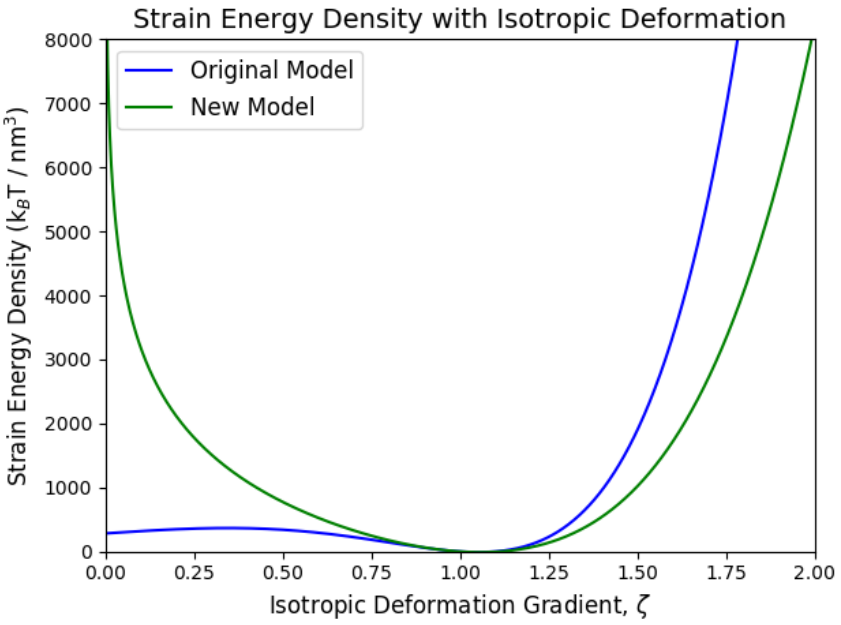


FIGURE 3.1: The effect of isotropic expansion on the strain energy density for both the original and new FFEA constitutive elastic models. We can see that although both models approximate the correct behaviour about $\zeta = 1$, the undeformed state, the original model incorrectly models the energy density at large compressions.

Firstly, considering only the derivative from Equation 3.6, substitution of Equation 3.5 gives us,

$$\begin{aligned}\frac{\partial \psi_n}{\partial F_{ik}} &= \frac{G}{2}\frac{\partial}{\partial F_{ik}}F_{lm}F_{lm} + \frac{\lambda}{4}\frac{\partial}{\partial F_{ik}}|\mathbf{F}|^2 - \left(\frac{\lambda}{2}+G\right)\frac{\partial}{\partial F_{ik}}\ln|\mathbf{F}|\,, \\ &= GF_{ik} + \frac{\lambda}{2}|\mathbf{F}|^2 F^{-1}_{ki} - \left(\frac{\lambda}{2}+G\right)F^{-1}_{ki}.\end{aligned}$$

Now, fully evaluating Equation 3.6,

$$\frac{1}{|\mathbf{F}|}\frac{\partial \psi_n}{\partial F_{ik}}F_{jk} = \frac{G}{|\mathbf{F}|}F_{ik}F_{jk} + \frac{\lambda}{2}|\mathbf{F}|\,\delta_{ij} - \left(\frac{\lambda}{2}+G\right)\frac{1}{|\mathbf{F}|}\delta_{ij}, \tag{3.7}$$

and rearranging gives,

$$\sigma^e_{ij} = \frac{G}{|\mathbf{F}|}\left(F_{ik}F^T_{kj} - \delta_{ij}\right) + \frac{\lambda}{2}\left(|\mathbf{F}| - \frac{1}{|\mathbf{F}|}\right)\delta_{ij}. \tag{3.8}$$

Figure 3.1 also shows the behaviour of the new elastic model. We can see that as $\zeta \to 0$, $\psi_n \to \infty$ as required. However, the gradient and curvature of this model clearly differ from the original at large deformations from the equilibrium structure. Although previous applications of FFEA remained largely within the linear regime and therefore did not see this effect, it is possible that future simulations may need to

take into account these non-linear effects. Consistency with the constitutive models for subsequent simulations must be adhered to, as material parameters such as the shear and bulk moduli, even when constant between simulations, may have different effects within the different constitutive models for large deformations.

We validate the new constitutive model with a simple set of simulations performed on cuboid objects. We built 4 tetrahedral meshes of different coarsenesses, each with length 15nm, width 3nm and height 1nm, and with the length aligned along the z-axis (see Figure 3.2, inset). Each model had the same material parametrisation, with a homogeneous Young's modulus $E = 1\text{GPa}$, Poisson ratio, $\nu = 0.35$ and all internal and external viscosities $\mu = 1 \times 10^{-3}\text{Pa.s}$. For each cuboid model we ran a series of 15 simulations, where, in each simulation, we applied a different constant force to the nodes making up the faces with normal vectors in the $\pm z$ directions. These forces were in the same direction as the face normals, so as to induce tensile strain in the cuboids. The simulations were performed in the absence of thermal noise, allowing the cuboids to settle into a new static equilibrium under the action of the external forces. We then measured the induced strains and effective stresses from the final state of the simulation[1], and plotted the resulting stress-strain curves. These are shown in Figure 3.2.

We see that for each of the four different meshes we obtain exactly the same overall pattern in the strain response to applied stress. For the set of strain values $\epsilon < 1.0$ we calculated a linear regression with respect to stress. The graphs show that the gradient of this line is within 0.1% of the true value of the Young's modulus, implying that the linear elastic regime is adhered to for extensions up to approximately the original length of the object. From there, we see a continual increase in the magnitude of the elastic response as the cuboid enters the non-linear regime from the underlying hyper-elastic model. This is exactly the behaviour we expect given the underlying energy curve for isotropic expansion.

3.2 Inertialess Formulation of FFEA

The original formulation of FFEA naturally takes into account the inertia within each molecule [71]. This was partly because our main reference model, the standard molecular dynamics algorithm, takes the mass into account by default due to account for the types of oscillatory motion occurring at atomic resolution [85], but also because the

[1] As FFEA objects are compressible, the surfaces reduced in area as strain increased, giving a varying amount of stress under the action of a constant force.

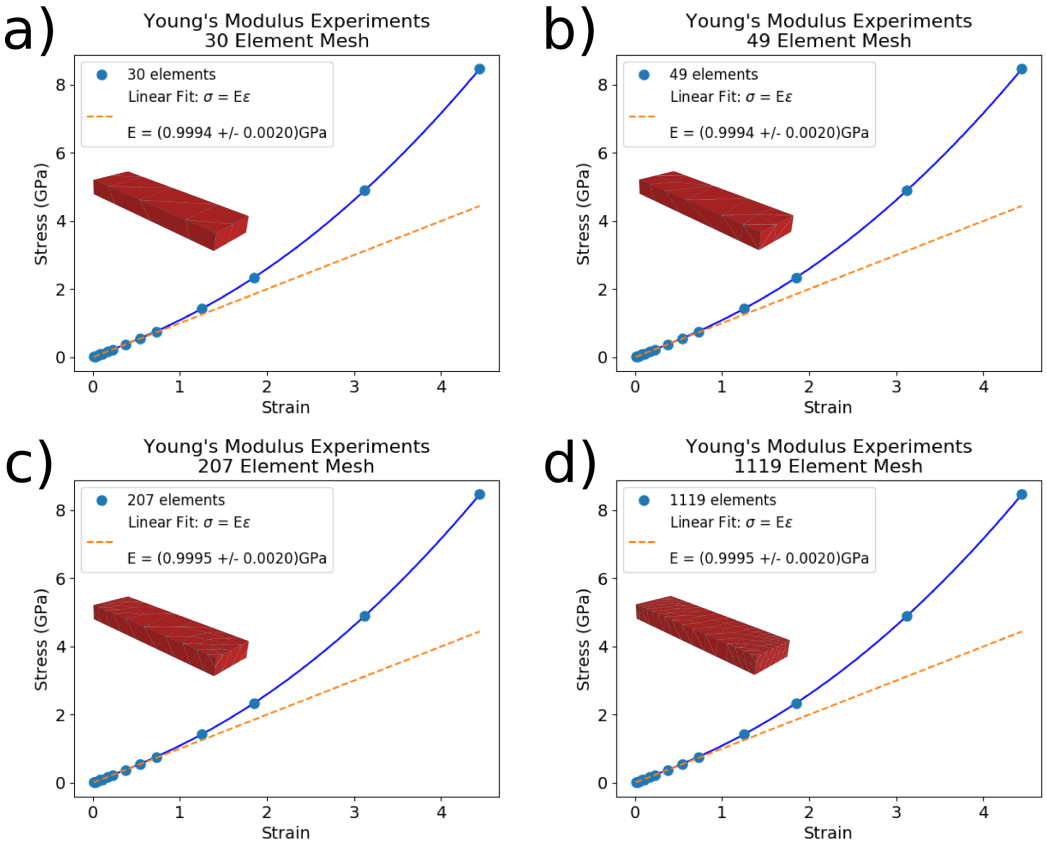


FIGURE 3.2: The stress-strain curves obtained by applying varying levels of stress to cuboid structures of different levels of mesh coarseness. **a)** is a 30 element mesh, **b)** a 49 element mesh, **c)** a 207 element mesh and **d)** a 1119 element mesh. The inset images in each graph are FFEA visualisations of the undeformed cuboids.

Cauchy momentum equation itself contains mass as a consequence of Newton's Second Law,

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\right) = \frac{\partial \sigma_{ij}}{\partial x_j}, \tag{3.9}$$

where the density, $\rho = \rho(x_i)$, gives the mass distribution throughout the continuum.

However, the following section will show that mass has little effect on the resultant dynamics of many mesoscopic biological systems. Not only that, the inclusion of mass within FFEA places a relatively low upper bound on the allowed size of the time step required for stable numerical integration, such that it is actually a computational hindrance to take inertia into account given that the relevant dynamics do not depend on it.

We have successfully reformulated the FFEA algorithm by removing the inertia from all calculations, and in the following section we discuss the inertialess approximation to the FFEA equation of motion, and perform simulations on simple, regular geometric structures to verify the software implementation of inertialess FFEA simulations.

3.2.1 Time Scales within FFEA

In order to determine whether it is necessary to consider inertia within FFEA, we must look at how the mass affects the dynamics of interest within an FFEA system i.e. a mesoscopic biological system. FFEA is designed to study the long-time dynamics of these systems, such as the dynamical modes of motion of our biomolecule associated with large scale deformations. As an example, consider the bending motions of a mesoscopic cantilever of length L. Theoretical analysis of beam bending shows us that the motion of an object fixed at one end can be readily decomposed into Fourier modes with associated wavelengths, λ_i [86]. From this decomposition we find that the fundamental mode corresponding to bending fluctuations of the entire cantilever, $\lambda_0 \approx 2L$, has a longer associated time scale, τ_0 than any of the higher order fluctuations, $\lambda_{i\neq 0} < 2L$. If we order these modes such that $\lambda_{i+1} \leq \lambda_i$, then we find that $\tau_{i+1} \leq \tau_i$.

To find the equivalent set of dynamical modes for a three-dimensional FFEA system, we begin by making a linear approximation to the elastic force vector in the FFEA equation of motion. Following the finite element approximation, the core equation of motion solved within FFEA in the absence of external forces is,

$$M_{ij}\ddot{x}_j + \Lambda_{ij}\dot{x}_j + C_i = N_i, \tag{3.10}$$

which includes the mass matrix, M_{ij}, the viscosity matrix, Λ_{ij} and the elasticity vector, C_i, and the indices i and j extend over all nodes and also their directional components. Each of these terms describe a different type of coupling between each pair of nodes in the system. We have seen that the elasticity vector, C_i, has a non-linear response at large deformations from equilibrium (see Figure 3.1). This means that it cannot be written exactly as a matrix applied to a position vector. But, for ease of analysis, we will approximate the force as linear about the equilibrium structure, $\vec{x}_0$, at which the elastic stress is $\vec{0}$. This gives,

$$M_{ij}\ddot{x}_j + \Lambda_{ij}\dot{x}_j + \cancel{C_i\Big|_{\vec{x}_0}} + \frac{\partial C_i}{\partial x_j}\Big|_{\vec{x}_0}(x_j - x_{0,j}) = N_i. \tag{3.11}$$

Redefining our position variable $\vec{x}$ to measure deviation from equilibrium, such that $\vec{x} - \vec{x}_0 \to \vec{x}$,

$$M_{ij}\ddot{x}_j + \Lambda_{ij}\dot{x}_j + K_{ij}x_j = N_i, \tag{3.12}$$

where $K_{ij} = \frac{\partial C_i}{\partial x_j}\Big|_{\vec{x}=\vec{0}}$ is the local stiffness matrix. By making this approximation, we have reduced the FFEA equation of motion to a simpler system of linear, second order stochastic differential equations.

We can simplify the system further by making the assumption that $\mathbf{M}$, $\mathbf{\Lambda}$ and $\mathbf{K}$ are simultaneously diagonalisable, that is, that they have a mutual set of eigenvectors. Using the matrix of arranged eigenvectors, $\mathbf{V}$, the equation of motion may be rewritten as follows,

$$\mathbf{V}^{-1}\mathbf{M}\mathbf{V}\ddot{\vec{y}} + \mathbf{V}^{-1}\mathbf{\Lambda}\mathbf{V}\dot{\vec{y}} + \mathbf{V}^{-1}\mathbf{K}\mathbf{V}\vec{y} = \mathbf{V}^{-1}N_i,$$
$$\mathbf{m}\ddot{\vec{y}} + \boldsymbol{\lambda}\dot{\vec{y}} + \mathbf{k}\vec{y} = \vec{n},$$

where $\mathbf{m}, \boldsymbol{\lambda}$ and $\mathbf{k}$ are diagonal matrices of eigenvalues, $\vec{y} = \mathbf{V}^{-1}\vec{x}$ and $\vec{n} = \mathbf{V}^{-1}\vec{N}$. By replacing our coordinates $\vec{x}$ with the new set $\vec{y}$, our system of equations of motion become separable. Each of these equations describe a different *normal mode* of the system and it is from these that we may estimate the time-scales associated with this system. Taking a single equation from the set, we reduce the single mode to a one-dimensional equation of motion with a single coordinate y,

$$m\ddot{y} + \lambda\dot{y} + ky = n, \tag{3.13}$$

where m, λ and k are respectively the mass, drag and elasticity eigenvalues associated with this particular mode, and n is the appropriate thermal noise function in this coordinate system. In the absence of the stochastic noise, the equation is linear and has well known solutions of the form $y = A\exp(b_+t) + B\exp(b_-t)$, where A and B are constants, and,

$$b_\pm = \frac{\lambda}{2m}\left(1 \pm \sqrt{1 - 4\frac{km}{\lambda^2}}\right), \tag{3.14}$$

which can be rewritten as,

$$b_\pm = \frac{1}{2\tau_m}\left(1 \pm \sqrt{1 - 4\frac{\tau_m}{\tau_k}}\right), \tag{3.15}$$

where $\tau_m = \frac{m}{\lambda}$ and $\tau_k = \frac{\lambda}{k}$ are the time constants associated with the inertial damping and elastic damping respectively, or, the evolution of velocity and positional degrees of freedom respectively. The three possible solution regimes which exist here are given in Table 3.1. Equation 3.15 shows that the regime limits are determined by the ratio of the two time scales, their relative size determining whether our system has oscillatory motion or pure exponential decay. For such a simplified system, representing a single dynamical mode within a biomolecule, we can estimate the values of these time scales using values representative of the mesoscale and therefore determine which regime is applicable for a generic mesoscopic biomolecule.

$4\frac{\tau_m}{\tau_k}$	Discriminant	Regime	Type of Motion
< 1	> 0	Over Damped	Exponential Decay
$= 1$	$= 0$	Critically Damped	Exponential Decay
> 1	< 0	Under Damped	Oscillatory

TABLE 3.1: The three regimes of dynamical motion in FFEA, written in terms of the intrinsic time constants. With these constants, we can see that dynamical regimes are more effectively defined by their dynamical time scales.

For a highly over damped system, the long-time solution to Equation 3.13 is,

$$y = y_0 \exp\left(-\frac{t}{\tau_k}\right), \tag{3.16}$$

showing that the over damped dynamics are not affected by the mass at all. Hence, the dynamical regime in which mesoscopic biomolecules exist will in turn determine whether or not the inclusion of mass is necessary for an FFEA simulation. We can estimate which regime this is for a given system by taking the normal mode with the largest associated length scale, and hence the largest expected time scales, and calculating the ratio between τ_k and τ_m for material parameters representative of the biological mesoscale.

We will consider two geometrically simple objects representative of the mesoscale and study their resulting dynamics. Since the drag on, and effective elasticity of a biomolecule depend strongly upon its shape, we will discuss both globular and elongated biomolecules by defining the aspect ratio, $a = \frac{L}{\sqrt{A}}$; the axial length of a biomolecule divided by the square-root of its corresponding cross-sectional area. Our two test models, with clearly differing aspect ratios, are shown in Figure 3.3.

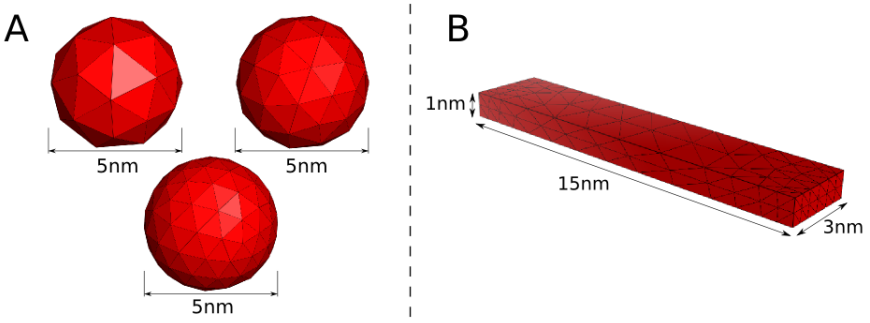


FIGURE 3.3: Structures used to verify the inertialess approximation to FFEA. A. 3 spheres of differing mesh coarseness, each with radius, $R = 5nm$. B. A single cuboid, $1nm \times 3nm \times 15nm$

3.2.1.1 Globular Biomolecules - Small Aspect Ratio

For $a \approx 1$, we take a spherical object of radius R as our test system. Provided we have similar bulk and shear moduli, we would expect the slowest mode within this object to be a uniform radial expansion. To calculate a linear elastic parameter, k, we first use the definition of the bulk modulus to calculate the change in pressure required for the sphere to contract,

$$\Delta P = -K \int_{V_i}^{V_f} \frac{1}{V} dV = -K \ln \frac{V_f}{V_i}, \tag{3.17}$$

where P is the pressure applied to the surface, and V_i and V_f are the initial and final volumes respectively. Since the object remains spherical when undergoing radial expansions, we can write Equation 3.17 in terms of the radius only,

$$\Delta P = -3K \ln \frac{R + \Delta r}{R}, \tag{3.18}$$

where the initial radius corresponding to V_i, $r_i = R$ and the final radius $r_f = R + \Delta r$. We can linearise the logarithm around $\Delta r = 0$,

$$\Delta P = -3 \frac{K}{R} \Delta r, \tag{3.19}$$

which gives us a linear approximation for the pressure change required for pure radial contractions. Rewriting this in terms of a linear restoring force at the surface allows us to introduce the parameter k,

$$\Delta F = -k\Delta r = -A\Delta P, \tag{3.20}$$

where A is the representative area of the sphere. Substituting in our expression for ΔP, and using an area $A = 4\pi R^2$ gives us our linear elastic parameter for this system,

$$\begin{aligned} k\Delta r &= \frac{3K}{R} \Delta r \times 4\pi R^2, \\ k &= 12\pi RK, \end{aligned} \tag{3.21}$$

giving us a linear elastic constant in terms of the intrinsic object parameters. It is important to note at this stage that the effective elasticity for a given motion is not independent of the geometry of the object.

To determine the effect of viscosity on the object, we simplify the drag experienced by the sphere when undergoing radial expansion by assuming it to be equivalent to

Stokes' drag on a sphere moving through a fluid [80],

$$\lambda = 6\pi\mu R. \tag{3.22}$$

Finally, writing the mass in terms of an homogeneous density, $m = \rho\frac{4}{3}\pi R^3$, we can calculate the form of the ratio of time constants as,

$$\begin{aligned} 4\frac{\tau_m}{\tau_k} &= 4\frac{mk}{\lambda^2}, \\ &= \frac{16}{9}\frac{\rho K}{\mu^2}R^2, \end{aligned} \tag{3.23}$$

which shows a relationship between the dynamics of radial expansion and the intrinsic material properties of the system.

To specify Equation 3.23 to a mesoscopic biomolecule, we substitute values representative of the mesoscale in general. The viscosity of the surrounding medium is close to that of water, $\mu = 1 \times 10^{-3}$Pa.s, and the densities are also similar to that of water, $\rho = 1 \times 10^{3}$kgm^{-3}. For the bulk modulus, a representative value obtained though experiments on mesoscopic biomolecules is $K = 1$GPa [87]. Finally, we choose the equilibrium radius as $R = 5$nm, a value sufficiently within our definition of the mesoscale [71]. Substituting in these values gives us our dynamical regime,

$$\begin{aligned} 4\frac{\tau_m}{\tau_k} &= \frac{16}{9}\frac{10^3 \times 10^9}{\left(10^{-3}\right)^2} \times \left(5 \times 10^{-9}\right)^2, \\ &= 44.4. \end{aligned} \tag{3.24}$$

With reference to Table 3.1, we see that this type of motion is reasonably under damped and therefore oscillatory[2]. For this level of under damping, energy is dissipated very slowly to the surroundings. For hydrodynamically realistic solvent environments, periodic waves would be formed and potentially sustained over long distances through the surrounding medium and so for a highly globular system it is necessary to include inertia if we are interested in how it fluctuates as a whole and the effects of those fluctuations on the surroundings.

[2]We have neglected the effects of internal viscosity in this example. However, we are far enough into the under damped regime that it is unlikely that the addition of internal friction will be sufficient to transition to the over damped regime, and so inertia should still be included for dynamics of small aspect ratio objects

3.2.1.2 Elongated Molecules - Large Aspect Ratio

Many mesoscopic biomolecules are not as isotropically globular as a sphere, but are formed of long, thin regions for which the aspect ratio affects the drag and effective elasticity in a drastic way. Suppose instead that our test object is a cuboid of length L, height h and width w in the x, y and z directions respectively, and $L >> h$ (see Figure 3.3). Our specific case of $w = 3h$ and $L = 15h$ gives us a primary aspect ratio $a = \frac{L}{\sqrt{hw}} = 8.66$, much greater than that of a sphere. The most flexible elastic mode for this object will be a first order bend about the x-axis.

We can calculate an effective spring constant for the bending motion using Euler-Bernoulli beam bending theory. For a first order deflection of size Δr, elastic beam theory gives the following expression for static linear deformation in terms of the Young's modulus, E [88],

$$\begin{aligned} F &= \frac{48EI}{L^3}\Delta r, \\ &= \frac{4wh^3}{L^3}E\Delta r. \end{aligned} \tag{3.25}$$

where F is an applied force and I is the appropriate second moment of area. Again assuming the linear relation $F = k\Delta r$, we find that this motion has an effective linear elasticity given by,

$$k = \frac{4wh^3}{L^3}E, \tag{3.26}$$

and substituting in the values for w and h in terms of L,

$$k = \frac{4L}{16875}E, \tag{3.27}$$

where again, we note the strong effect of the geometry of the object on its effective elasticity.

We can calculate an approximate drag on the cuboid object using the drag coefficient on a long, slender body moving perpendicular to the primary axis, obtained by Cox *et al.* [89]

$$\lambda \approx \frac{4\pi\mu L}{\ln\left(\frac{L}{w}\right) - 0.806} \approx 5\pi\mu L, \tag{3.28}$$

which shows that the hydrodynamic drag is mostly determined by the largest length scale, L.

For a mass again being a function of the equilibrium density, $m = \rho hwL$, allowing us to estimate the time constants for the bending of an elongated molecule,

$$\begin{aligned} 4\frac{\tau_m}{\tau_k} &= 4\frac{mk}{\lambda^2}, \\ &\approx \frac{1}{4.68 \times 10^7}\frac{\rho E}{\mu^2}L^2. \end{aligned} \tag{3.29}$$

For our representative values $\rho = 1 \times 10^3 \text{kgm}^{-3}$, $E = 1\text{GPa}$, $\mu = 1 \times 10^{-3}\text{Pa.s}$ and $L = 15\text{nm}$, we find,

$$\begin{aligned} 4\frac{\tau_m}{\tau_k} &= \frac{1}{4.68 \times 10^7}\frac{10^3 \times 10^9}{(10^{-3})^2} \times \left(15 \times 10^{-9}\right)^2, \\ &\approx 4.81 \times 10^{-6}. \end{aligned} \tag{3.30}$$

We note that the pre-factor present in Equation 3.29, when compared to that for the spherical oscillations is Equation 3.24, is very small. This pre-factor emerged from the geometry of the object, and so we see that that for the bending motion of a molecule with a large aspect ratio, the dominant modes of motion are highly over damped. Although the drag increases with the largest length scale in the molecule, the elastic constant for bending scales inversely with approximately the third power of the largest length scale, and approximately the third power of the aspect ratio such that an increase in L with no increase in h or w would make the motion even more over damped. Consequently, the deflection simply decays exponentially, with no oscillatory motion at all. The energy in the system is quickly lost to the surrounding medium through dissipation and as such, velocities are not sustained within the system. As we will see in Chapter 5, many functional proteins within biological organisms have sub-structures of exactly this type; long, thin and flexible regions for which the motion is over damped. It is exactly these types of molecules that FFEA was designed to simulate.

We also note that for both the globular and elongated objects, the time scale ratio was proportional to L^2. This means that when the system approaches macroscopic sizes, the mass of the object dominates the geometric considerations in terms of these time scales, and inertia again becomes important, this time in terms of the global dynamics. It is at the mesoscale that the geometry of the object becomes the deciding factor in the type of motion the object can undergo.

The above arguments lead us to conclude that a large number of potential FFEA simulations will not require inertia in order to simulate the dynamics of interest for

the mesoscopic systems. Therefore, an inertialess solution protocol for this type of simulation is required.

3.3 Numerical Implementation of an Inertialess Formulation

Up until this point we have provided specific examples showing that long-time dynamics at the mesoscale are independent of inertia. A more general formalism can be derived through an alternate method of numerical integration of the FFEA equation of motion.

3.3.1 Solving the FFEA Equation of Motion using Numerical Integration

Even following the finite element approximation, there is no general analytical solution to Equation 3.10. In order to simulate our FFEA system, we must use numerical integration to update the positions of the nodes, and it is this process which is limited by the dynamical time scales of each specific system.

The original numerical integration method for FFEA is a two step forward Euler integration of the acceleration vector. Rewriting Equation 3.10 in matrix notation and in terms of velocities, $\vec{v}$,

$$\mathbf{M}\dot{\vec{v}} + \mathbf{\Lambda}\vec{v} + \vec{C} = \vec{N}. \tag{3.31}$$

When applying forward integration we use the velocities at the current time t on the RHS of Equation 3.31 to calculate the new velocities $\vec{v}(t + \Delta t)$,

$$\begin{aligned} \frac{\vec{v}(t+\Delta t) - \vec{v}(t)}{\Delta t} &= \mathbf{M}^{-1}\left(\vec{N}(t) - \mathbf{\Lambda}\vec{v}(t) - \vec{C}(t)\right), \\ \Rightarrow \vec{v}(t+\Delta t) &= \left(\mathbf{I} - \Delta t\boldsymbol{\tau}_m^{-1}\right)\vec{v}(t) + \Delta t\mathbf{M}^{-1}\left(\vec{N}(t) - \vec{C}(t)\right), \end{aligned} \tag{3.32}$$

where $\boldsymbol{\tau}_m = \mathbf{\Lambda}^{-1}\mathbf{M}$ is the matrix of inertial time constants associated with the relative motion between each node, and each direction, of the FFEA object. Further numerical expansion of $\vec{v}(t + \Delta t)$ also allows us to update the positions of the nodes,

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \Delta t\left(\mathbf{I} - \Delta t\boldsymbol{\tau}_m^{-1}\right)\vec{v}(t) + (\Delta t)^2\mathbf{M}^{-1}\left(\vec{N}(t) - \vec{C}(t)\right). \tag{3.33}$$

Equations 3.32 and 3.33 make up the entire current numerical integration scheme with FFEA. By linearising $\vec{C}(t)$, we see the effect of both time constant matrices,

$$\begin{aligned}\vec{x}(t+\Delta t) &= \vec{x}(t) + \Delta t\left(\mathbf{I} - \Delta t\boldsymbol{\tau}_m^{-1}\right)\vec{v}(t) + (\Delta t)^2\mathbf{M}^{-1}\left(\vec{N}(t) - \mathbf{K}\vec{x}(t)\right), \\ &= \left(\mathbf{I} - (\Delta t)^2\boldsymbol{\tau}_m^{-1}\boldsymbol{\tau}_k^{-1}\right)\vec{x}(t) + \Delta t\left(\mathbf{I} - \Delta t\boldsymbol{\tau}_m^{-1}\right)\vec{v}(t) + (\Delta t)^2\mathbf{M}^{-1}\vec{N}(t), \quad (3.34)\end{aligned}$$

where $\boldsymbol{\tau}_k = \mathbf{K}^{-1}\boldsymbol{\Lambda}$ is the viscous time constant matrix.

It can be shown that Equation 3.32 is numerically unstable over repeated integration steps if $\frac{\Delta t}{e_m} > 1$, where e_m is the smallest eigenvalue of $\boldsymbol{\tau}_m$, Equation 3.34 is unstable for both $\frac{\Delta t}{e_m} > 1$, and $\frac{\Delta t^2}{e_m e_k} > 1$, where e_k is the smallest eigenvalue of $\boldsymbol{\tau}_k$. Hence, a forward integration method is numerically limited by the smallest time scale within our system. It can be shown, by similar geometric arguments as those presented in Section 3.2.1, that the inertial time scales have all of the length scale dependence of the calculated time step ratios, and so with biologically representative material parameters, the smallest time scales almost always correspond to smallest length scales. In FFEA, the smallest time scales are therefore rapid, oscillatory, inertial motion between adjacent nodes.

An alternative approach to the solution of the FFEA equation of motion is to use implicit backward integration, in which we use the velocities $\vec{v}(t+\Delta t)$, on the RHS of Equation 3.31. This leads to a substantially different numerical expansion,

$$\begin{aligned}\mathbf{M}\frac{\vec{v}(t+\Delta t) - \vec{v}(t)}{\Delta t} + \boldsymbol{\Lambda}\vec{v}(t+\Delta t) + \vec{C}(t) &= \vec{N}(t), \\ \left(\frac{1}{\Delta t}\mathbf{M} + \boldsymbol{\Lambda}\right)\vec{v}(t+\Delta t) &= \frac{1}{\Delta t}\mathbf{M}\vec{v}(t) + \vec{N}(t) - \vec{C}(t). \quad (3.35)\end{aligned}$$

Multiplying though by $\boldsymbol{\Lambda}^{-1}$ allows us to see the effect of the inertial time scales,

$$\left(\frac{1}{\Delta t}\boldsymbol{\tau}_m + \mathbf{I}\right)\vec{v}(t+\Delta t) = \frac{1}{\Delta t}\boldsymbol{\tau}_m\vec{v}(t) + \boldsymbol{\Lambda}^{-1}\left(\vec{N}(t) - \vec{C}(t)\right). \quad (3.36)$$

Upon repeated iterations, Equation 3.36 is numerically stable for all values of Δt. Taking Δt to be large in comparison to the largest eigenvalue of $\boldsymbol{\tau}_m$,

$$\begin{aligned}\vec{v}(t+\Delta t) &= \boldsymbol{\Lambda}^{-1}\left(\vec{N}(t) - \vec{C}(t)\right), \\ \boldsymbol{\Lambda}\vec{v}(t+\Delta t) + \vec{C}(t) &= N(t), \quad (3.37)\end{aligned}$$

which is the original equation of motion, Equation 3.10, with the inertial term neglected. As backward iteration allows us to take any size time step whilst retaining numerical stability, the effect on the FFEA equation of motion is that we can 'skip

over' all of the dynamics that FFEA was not designed to simulate, i.e. the small length scale inertial motion, whilst still retaining the numerical accuracy required to simulate motion at the mesoscale. Therefore, we can implement Equation 3.37 as our new equation of motion.

Using a single forward Euler integration, we can calculate the new positions of all of the nodes in the FFEA object as a function of their instantaneous velocities,

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \Delta t \mathbf{\Lambda}^{-1} \left(\vec{N}(t) - \vec{C}(t) \right), \tag{3.38}$$

which is the complete numerical integration scheme[3].

In order to validate the inertialess approximation and its implementation into FFEA, we must ensure that the dynamics are equivalent in both space and time between inertial and inertialess simulations of the same processes.

3.4 Verification - Cuboid Normal Modes

Validation of the spatial dynamics can be achieved by comparing the normal modes of motion for a test object. We will use the test model from Section 3.2.1.1, an elongated cuboid, for which we expect the slowest modes (associated with the smallest eigenvalues) to be identical in the inertial and inertialess cases due to their over damped nature. As the modes become faster and more localised, we expect the mass to play a larger role leading to the modes no longer being identical.

3.4.1 Theoretical Background

The stochastic noise function within FFEA derived by Oliver *et al.* is such that that there is only δ -correlation in both space and time, and that the equipartition theorem is fulfilled [71]. Hence, we have a Boltzmann distribution for the probability that the

[3] We must use a forward method in the final numerical integration step because the elastic force vector $\vec{C}$ is non-linear with respect to $\vec{x}$, and so we cannot easily use a factor of $\vec{x}(t+\Delta t)$ for stable, backwards integration

system is in any energy state,

$$\begin{aligned} P(E) &= \frac{1}{Z}\exp\left(-\frac{E}{k_BT}\right), \\ &= \frac{1}{Z}\exp\left(-\frac{E_k}{k_BT}\right)\exp\left(-\frac{E_s}{k_BT}\right), \\ \Rightarrow P(E_s) &= \frac{1}{Z_s}\exp\left(-\frac{E_s}{k_BT}\right), \end{aligned} \tag{3.39}$$

where the total energy, $E = E_k + E_s$, the sum of kinetic and strain energies respectively, and the total partition function $Z = Z_k Z_s$ is the product of the two independent partition functions for strain and kinetic energies.

As Equation 3.39 shows that the strain and kinetic energy states form separable subsystems, the form of the strain energy partition function must be,

$$\begin{aligned} Z_s &= \int_0^\infty \exp\left(-\frac{E_s}{k_BT}\right) dE_s, \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left(-\frac{x_i K_{ij} x_j}{k_BT}\right) dx_i dx_j, \end{aligned} \tag{3.40}$$

showing that the distribution of strain energy, in the linear elastic regime, is purely a function of the linear elasticity matrix $\mathbf{K}$. This allows us to calculate the normal modes of motion available to the system, those with respect to x_i, through analysis of $\mathbf{K}$ alone. However, we must also verify that our software implementation is correct by extracting these modes from a simulation which implements $\mathbf{K}$ directly, i.e. an FFEA simulation. This can be achieved using Principal Component Analysis (PCA), a generalised method for compressing large data sets via a relatively small amount of analysis of the statistical properties of that data [90]. For our purposes, PCA can calculate the covariance matrix of nodal positions using the set of simulation frames,

$$C_{ij} = \langle X_i X_j \rangle_t = \frac{1}{N}\sum_t^N (X_i(t) - \bar{X}_i)(X_j(t) - \bar{X}_j), \tag{3.41}$$

where the sum is taken over all N frames present in the simulation. If the simulation has run for an appropriately long period of time (longer than the longest time scale, τ_k) then we can assume the ergodic theorem holds, and postulate that the time average covariance matrix from Equation 3.41 is equal to the spatial average covariance matrix. We can then use equipartition of energy to equate the following,

$$\begin{aligned} \frac{1}{2}K_{ik}\langle X_k X_j \rangle_t &= \frac{1}{2}k_BT\delta_{ij}, \\ \langle X_i X_j \rangle_t &= k_BT K^{-1}_{ij}. \end{aligned} \tag{3.42}$$

So we find that over a fully converged simulation, the elasticity matrix is linked via equipartition to the covariance matrix. Taking the eigen-decomposition of K_{ij},

$$\begin{aligned}\langle X_i X_j \rangle_t &= k_B T \left(e_{i\alpha} D_{\alpha\beta} e^{-1}_{\beta j} \right)^{-1}, \\ &= k_B T e_{i\alpha} D^{-1}_{\alpha\beta} e^{-1}_{\beta j}, \\ &= e_{i\alpha} \left(k_B T D^{-1}_{\alpha\beta} \right) e^{-1}_{\beta j}, \end{aligned} \tag{3.43}$$

where the matrix $e_{i\alpha}$ is the matrix of eigenvectors (each eigenvector denoted by α) and $D_{\alpha\beta}$ is the diagonal matrix of eigenvalues. This means that the eigenvectors of $\mathbf{C}$ and $\mathbf{K}$ are identical in the limit of simulation convergence and the respective eigenvalues are simply the inverse of one other, multiplied by a factor of $k_B T$. Therefore, PCA allows us to calculate the normal modes of motion as a function of a simulation itself.

For the dominant modes in which the inertialess approximation is valid, we would expect to find that the PCA eigenvectors from both an inertial simulation (using Equation 3.31 as the equation of motion) and an inertialess simulation (using Equation 3.37 as the equation of motion) are the same. We denote these eigen-systems using the eigenvector matrices $\mathbf{e}_1$ and $\mathbf{e}_2$ respectively. As the matrix $\mathbf{K}$ is symmetric, so too is $\mathbf{C}$ and therefore the following identity must be true of their respective orthogonal eigen-systems,

$$\mathbf{e}_1^T \mathbf{e}_2 = \mathbf{I}, \tag{3.44}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{e}_1$ and $\mathbf{e}_2$ correspond to the eigenvector matrices from inertial and inertialess simulations respectively.

3.4.2 Results

To verify the expected diffusion result derived in Section 3.4.1, we built a single cuboid mesh with 966 elements and dimensions 1nm ×3nm ×15nm to avoid degeneracy between dimensions (see Figure 3.3). Simulations of both an inertial and non-inertial system were performed until the dynamics were sufficiently converged to show the expected result. Principal component analysis was then performed on the resulting trajectories in order generate the set of elastic normal modes of the system from the covariance matrix. The PCA analysis was performed using the pyPcazip software package [91]. pyPcazip was was designed primarily for atomistic trajectory data from MD simulations, but we were able to compensate for this simply by converting the nodes of our FFEA system into the PDB format.

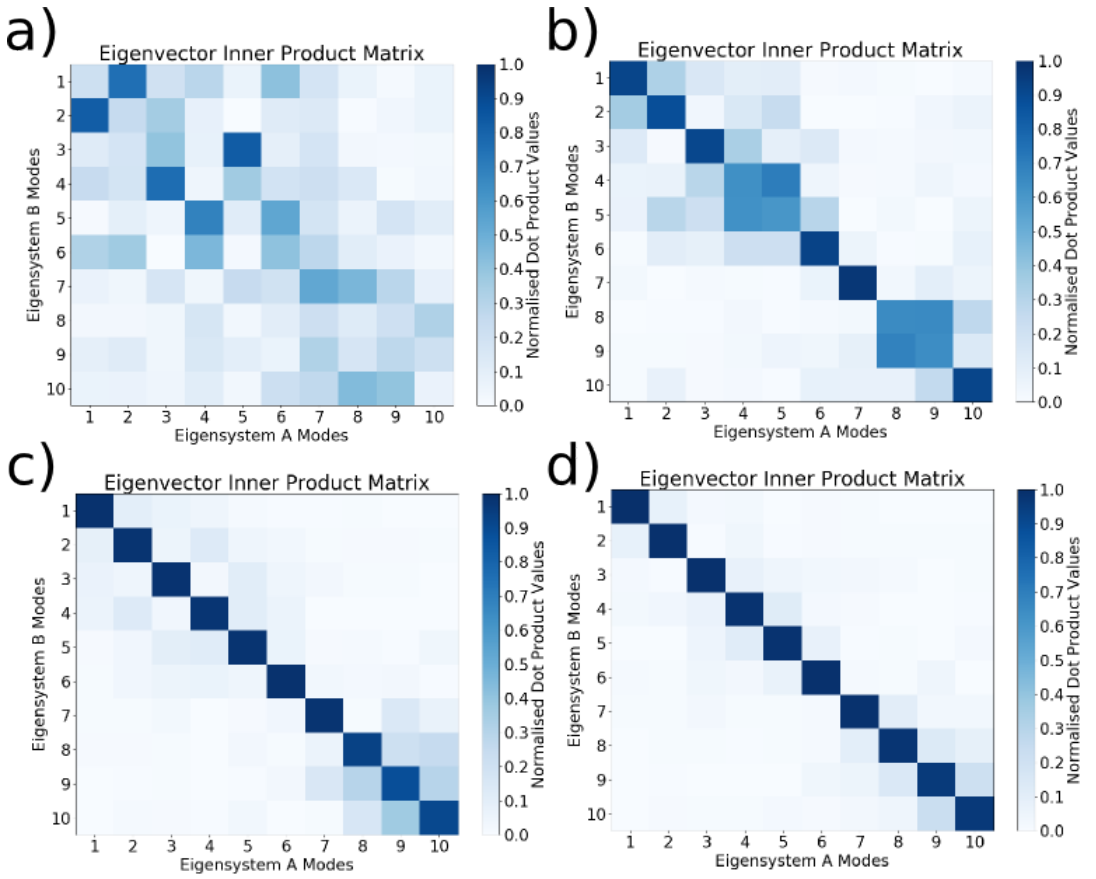


FIGURE 3.4: Eigen-system inner product matrices for the dominant 10 modes of both an inertial (eigen-system A) and inertialess (eigen-system B) cuboid. **a)**, **b)**, **c)** and **d)** represent analysis of $100, 1000, 5000$ and 10000 simulation frames respectively from a simulation of total length 0.5μs. We can see that as dynamical convergence occurs, this matrix tends toward the identity matrix as expected, showing that biological motion is indeed over damped.

The slowest 10 modes of each system were the used to form the eigenvector matrices, which were multiplied as in Equation 3.44 to form the following matrix,

$$A_{\alpha\beta} = e^T_{1,\alpha i} e_{2,i\beta}. \tag{3.45}$$

The evolution of the structure of $A_{\alpha\beta}$ throughout a simulation is visualised as a heat map in Figure 3.4.

We clearly see convergence towards the identity matrix for the lowest ten normal modes, which is enough to support the notion that FFEA simulations of biological molecules with physically realistic properties are highly over damped. To get a numerical value for the level of diagonalisation, we calculated the Pearson sample correlation coefficient, r, of the matrix by modelling the elements of the matrix as probabilities of a discrete data set i.e. $A_{\alpha\beta}$ is modelled as the probability of eigenstate α being equal to eigenstate

β. This analysis gives a value of 1 for perfectly diagonal and -1 for perfectly anti-diagonal. For the four examples in Figure 3.4, we obtained the values $r = 0.458$, $r = 0.789$, $r = 0.890$ and $r = 0.916$ for 5ns, 50ns, 250ns and 500ns of simulation time respectively.

For completeness, we extracted the 100 most flexible modes from each trajectory, and plotted their inner product matrix as a heat map in Figure 3.5. This larger data set shows the identity property becoming less well defined at higher modes, with $r = 0.620$ after 0.5μs of simulation time. As these higher modes should have converged much more quickly than the lower modes, being associated with smaller length scales, we believe this loss of identity to be due to the expected breakdown of inertial behaviour for faster modes as described in Section 3.3.1. However, as the slowest 10 modes contain 80.31% of the calculated variance, we maintain that this is a sufficient level of overlap to justify the claim that the two eigen-systems are equivalent with respect to the global dynamical behaviour.

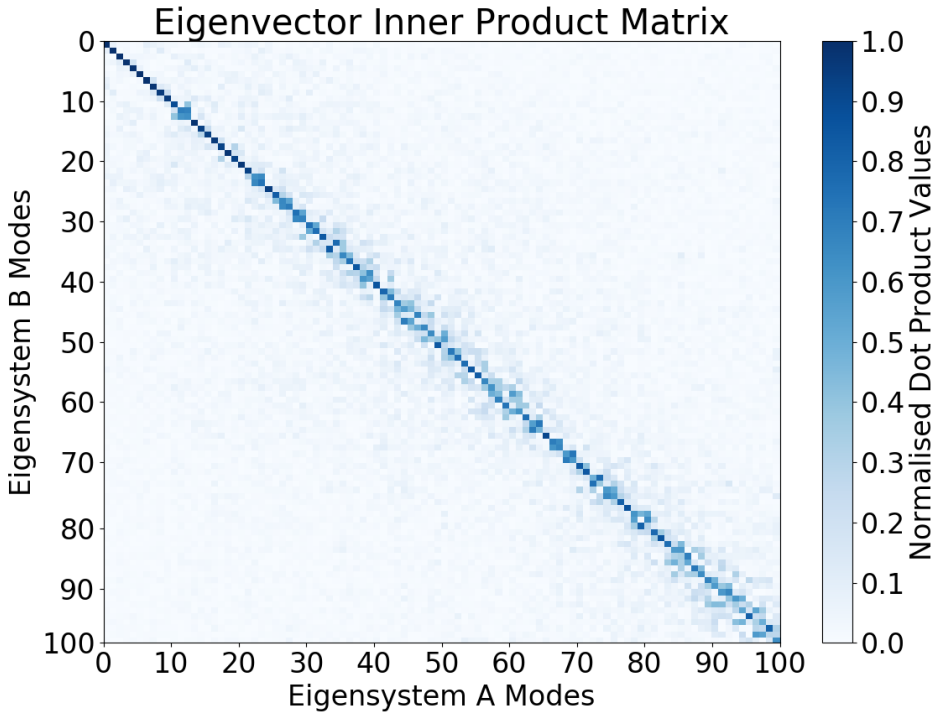


FIGURE 3.5: Eigen-system inner product matrix for inertial and non-inertial cuboids for the top 100 most flexible modes.

3.5 Verification - Spherical Diffusion

To verify the dynamics of FFEA processes which depend on time, we can measure the diffusive properties of a mesoscopic spherical object. Here we present a small theoretical background to diffusive processes.

3.5.1 Theoretical Background

As our inertialess approximation is based on the assumption that the dynamics of mesoscopic biological molecules are highly over damped, the diffusion of a mesoscopic sphere through a viscous background medium of viscosity μ should have the well known solution [21],

$$\begin{aligned} \langle x^2 \rangle &= 2Dt = 2\frac{k_B T}{\lambda}t, \\ \Rightarrow \langle r^2 \rangle &= 6\frac{k_B T}{\lambda}t, \end{aligned} \tag{3.46}$$

where $\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle$ is the three dimensional squared displacement, and λ is the drag on the sphere. In our case, for a sphere with radius R, this drag force is described by Stokes' formula (Equation 3.22). Thus, for an inertialess sphere, we would expect the squared displacement of the sphere from its starting position to be a function of the drag and to increase linearly with time. We can better express this linearity by taking logarithms of Equation 3.46,

$$\ln\left(\langle r^2 \rangle\right) = \ln(t) + \ln\left(6\frac{k_B T}{\lambda}\right), \tag{3.47}$$

which allows us to view over damped diffusion as a curve with a gradient $g = 1$ and an associated y-intercept $y_0 = \ln\left(6\frac{k_B T}{\lambda}\right)$.

For the same over damped system, but with inertia explicitly included, the over damping means that after a long period of time, the diffusion should tend toward that of an inertialess system, and be described by Equation 3.47. However, for small periods of time, we would expect the diffusion to have the following form,

$$\begin{aligned} \langle x^2 \rangle &= \langle v^2 \rangle t^2 = \frac{k_B T}{m}t^2, \\ \Rightarrow \langle r^2 \rangle &= 3\frac{k_B T}{m}t^2, \end{aligned} \tag{3.48}$$

due to the fact that the inertia of an object allows it to retain a velocity v for a time $t < \tau_m$. In an over damped system the effects of these retained velocities would become

Sphere Model	Number of Elements	Number of Nodes
1 - Coarse	70	38
2 - Mid	122	64
3 - Fine	336	134

TABLE 3.2: Three different models of the same sphere, radius $R = 5nm$. These three allow us to verify the diffusion calculated by FFEA is independent of the coarseness of the model.

negligible over long periods of time. Again taking logarithms,

$$\ln\left(\langle r^2\rangle\right) = 2\ln\left(t\right) + \ln\left(\frac{3k_BT}{m}\right), \tag{3.49}$$

which allows us to view inertial diffusion as a curve with gradient $g = 2$ and y-intercept $y_0 = \ln\left(3\frac{k_BT}{m}\right)$. Notice that the curves of Equation 3.47 and Equation 3.49 must intersect for some value of t. Equating the two regimes using Equation 3.46 and Equation 3.48 gives us the time around which inertial diffusion tends towards inertialess diffusion for an over damped object,

$$\begin{aligned}\langle r^2\rangle &= 6\frac{k_BT}{\lambda}t = 3\frac{k_BT}{m}t^2,\\ t &= 2\frac{m}{\lambda},\\ &= 2\tau_m.\end{aligned} \tag{3.50}$$

So the time scale over which the effects of inertia become negligible is the inertial time constant, as we would perhaps expect.

3.5.2 Results

To verify the expected diffusion result derived in Section 3.5.1, we built three spherical tetrahedral meshes of the same radius, $R = 5nm$, but 3 different levels of coarseness (see Table 3.2). We performed twenty identical simulations of each model to allow the necessary averages to be taken. Each simulation was $20ns$ with an integration time step $dt = 1fs$, to allow for both the necessary temporal resolution to capture inertial effects for $t < 2\tau_m$ as well the transition to inertialess motion for $t > 2\tau_m$. FFEA currently simulates drag on an object by applying a local drag to each node, which has a characteristic radius r_i called the Stokes radius assigned to it during the initialisation routines. For a sphere of radius R and containing N nodes, we are able

to exactly simulate the drag it should experience as follows,

$$\begin{aligned} \lambda_{sphere} &= N\lambda_{node}, \\ 6\pi\mu R &= 6\pi\mu N r_s, \\ r_s &= \frac{R}{N}, \end{aligned} \tag{3.51}$$

where the characteristic radius $r_i = r_s \; \forall \; i$. For the mass, we set the density of each sphere to be the same for each model, $\rho = 1.5 \times 10^3 kg$, a value representative of globular biomolecules [87].

The results from the simulations are presented in Figures 3.6, 3.7 and 3.8. We can clearly see that for each model the theoretical diffusion trace is reproduced within the FFEA framework, with the explicit inertial diffusion trajectory transitioning from the graph with a gradient, $g = 2$ to the theoretical inertialess graph with $g = 1$ at exactly the point at which the graphs cross, $t = 2\tau_m$.

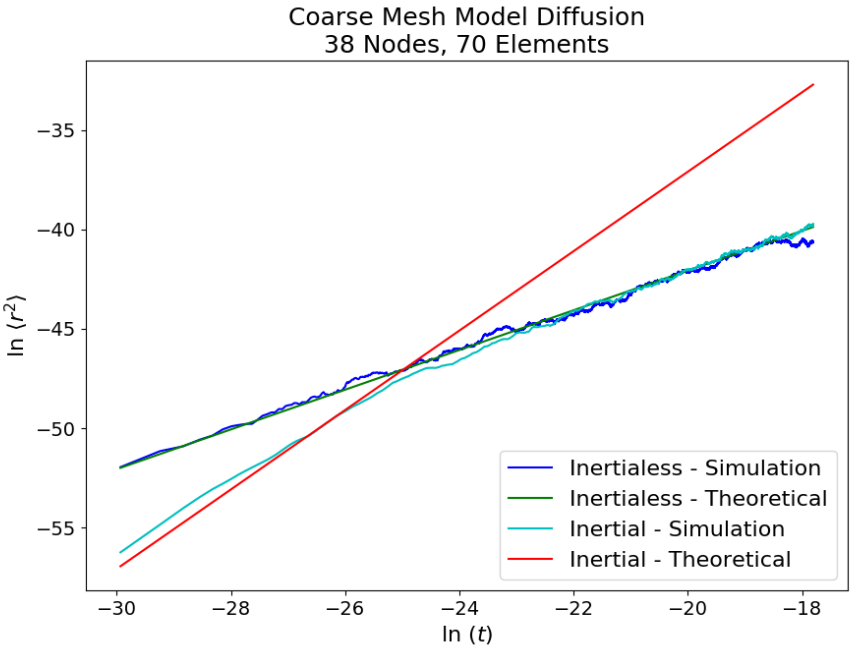


FIGURE 3.6: Diffusion trace of a 70 element sphere, radius $R = 5nm$.

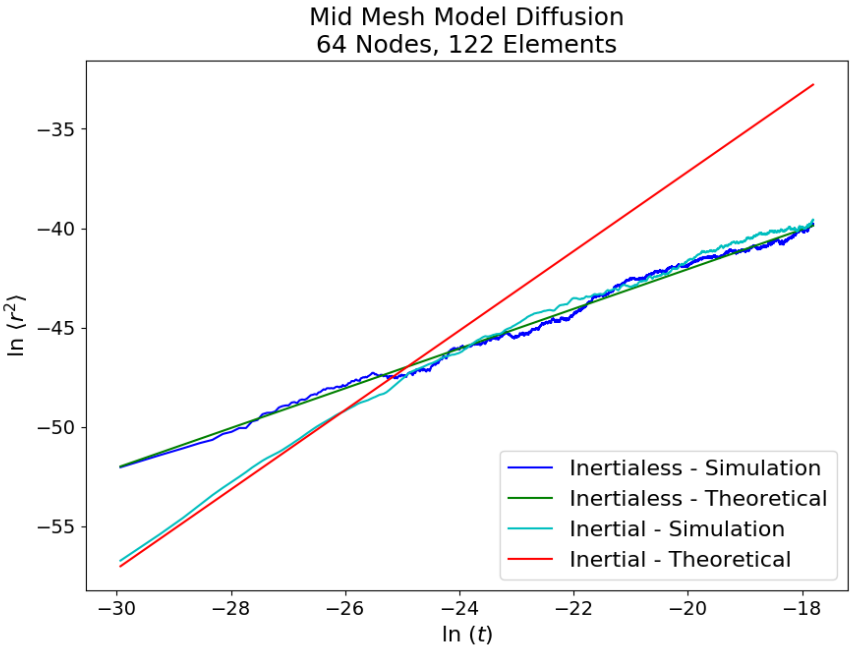


FIGURE 3.7: Diffusion trace of a 122 element sphere, radius $R = 5nm$.

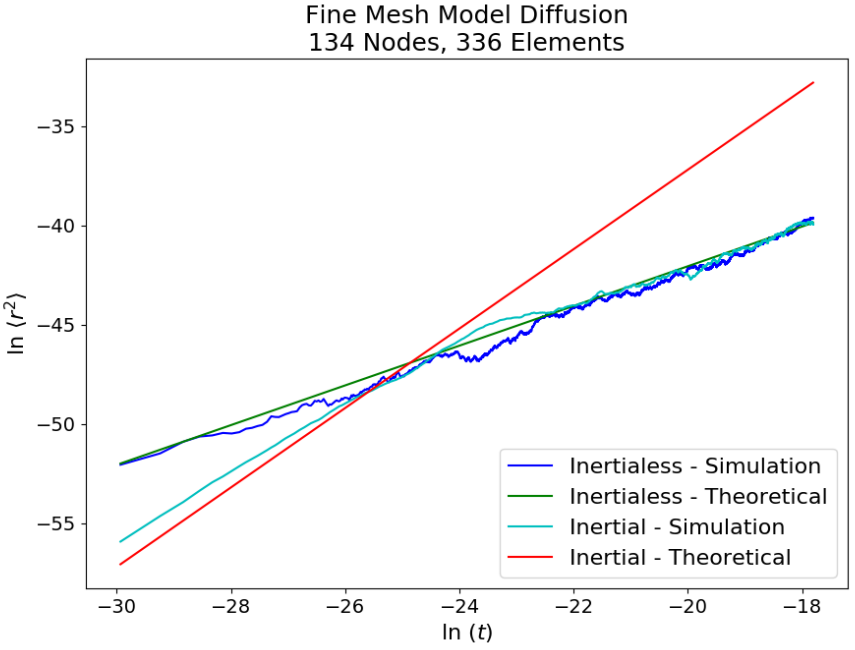


FIGURE 3.8: Diffusion trace of a 336 element sphere, radius $R = 5nm$.

3.6 Performance Increase

Section 3.4 and Section 3.5 provided the necessary verification to justify the usage of a non-inertial equation of motion (Equation 3.37) by default in an FFEA simulation. The equation has been reduced to a first order differential equation, and so we must now solve the equation of motion by inverting the viscosity matrix $\mathbf{\Lambda}$ in our solution regime, as opposed to the mass matrix $\mathbf{M}$ in the original formulation. We obtain the solution using the conjugate gradient method, an iterative steepest decent method for the solution of linear algebraic systems with symmetric, positive definite matrices [92]. The sparsity of our viscosity matrix makes such an iterative algorithm orders of magnitude faster than alternative, direct inversion methods.

However, compared with the mass matrix, the viscosity matrix has additional directional dependencies due to the strain rate tensor having non-zero off-diagonal components [80]. This means that for a system with N nodes, $\mathbf{M}$ has dimensions $N \times N$ whereas $\mathbf{\Lambda}$ is $3N \times 3N$. Not only this, unlike $\mathbf{M}$, which remains constant throughout the simulation[4], $\mathbf{\Lambda}$ must be continuously recalculated and rebuilt at each simulation time step to take into account varying strain rates between nodes. We already know that this has no implications on the overall dynamics of an FFEA simulation, but it will affect the speed of the program itself.

If we define our runtime as t_r and our simulation time as t_s, the goal of optimizing any piece of simulation software is to maximise the ratio $v_{sim} = \frac{t_s}{t_r}$, i.e. to get as much simulation time completed in as little real time as possible. Although the inertialess approximation means we can theoretically take much larger integration time steps whilst still retaining numerical stability, if the simulation itself takes longer due to the slower matrix building and inversion procedures at each time step, then we may as well continue to include the mass matrix.

To determine whether or not the inertialess approximation increases the speed of an FFEA simulation, we performed a parameter sweep over a number of variables contributing to the speed of the simulation. We firstly varied the Young's modulus, E, of the system as this parameter directly affects the size of the allowed time step by changing the viscous time scale. Additionally, we were able to check whether or not larger / smaller values affected the numerical efficiency of the conjugate gradient solver.

We have found that although numerical stability is theoretically retained for much larger time steps with the non-inertial system, the smallest length scale in the system, l_{min}, still affects the size of the time step that we can take. The connectivity of an

[4]Which is because the density of each element also remains constant throughout a simulation

FFEA mesh is constant throughout a simulation, and therefore each element must retain its relative topology. In other words, the normal vectors of each of the element faces are initially defined to point outwards when calculated as a cross product of the edge vectors.

However, if a node moves too far in a single direction, then it may pass through the opposite face of one of its associated elements. If this occurs, the normal vector no longer point outwards with respect to the element when calculated from its edge vectors. We refer to this effect in general as element inversion, and it is a problem due to physically unrealistic properties being calculated, such as negative volumes, which in turn lead to incorrect dynamics. Element inversion is the end result of numerical instabilities, where position and velocity fluctuations gradually build up in magnitude but it also occurs if a node simply doesn't have time to physically relax under the action of the equation of motion. This can occur if the magnitude of the thermal noise is too high, which we know from Equation 2.5 is itself a function of the time step. This is not a numerical stability problem but a limit of the mesoscale simulation itself in that our mesh must be at least as fine as the finest level of detail we wish to investigate. We cannot escape this limit, or the size of the thermal noise which it imposes. We refer to these two different occurrences of element inversion as unstable and stable inversion respectively.

To look at the effect of element inversion, we also varied l_{min} in our performance analysis by keeping the same mesh connectivity and varying the size of the whole molecule. For completeness, we also performed the iteration for three different mesh structures, a coarse, mid and finely detailed cubic structure, to see whether the structure of the matrices themselves affected the speed of their inversion.

For any of these simulations we can use any simulation time step Δt less than the smallest time scale within the system and get a numerically stable simulation. This gives us a spectrum of possible speeds for any simulation. We found the maximum speed of any given simulation, v_{sim}, by performing a bisection iteration procedure for the time step. The bisection method is as follows:

- Initialise:
  - Select Bisection Range ($\Delta t_{min} = 1 \times 10^{-17}$, $\Delta t_{max} = 1 \times 10^{-3}$)
  - Select Initial Time step ($\Delta t = \Delta t_{max}$)
  - Select Mesh Coarseness (Number of Elements $= 6, 93, 678$)
  - Select Young's Modulus ($1 \times 10^{7} Pa < E < 1 \times 10^{11} Pa$)

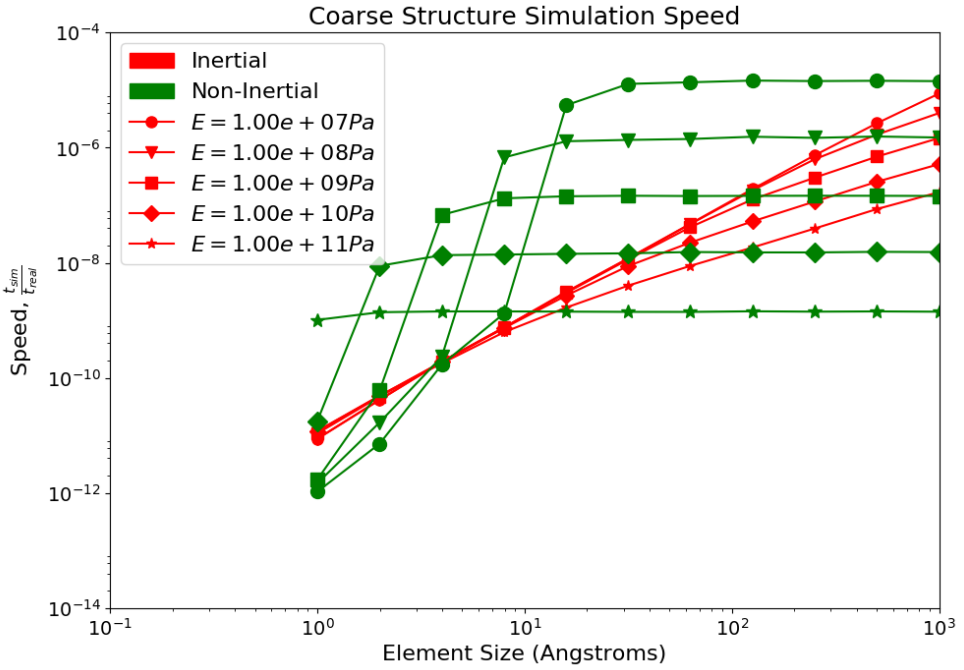


FIGURE 3.9: Maximum simulation speed for a 6 element cubic object as a function of the smallest element size, for a variety of different Young's Moduli

  - Select System Smallest Element Length ($1 \times 10^{-10}m < l_{min} < 1 \times 10^{-7}m$)

- Iteration:
  - Run FFEA simulation for 10000 time steps
  - If simulation successful:

    $\Delta t_{min} = \Delta t$ i.e. Δt could potentially be larger
  - Else:

    $\Delta t_{max} = \Delta t$ i.e. Δt must be smaller
  - Set new time step, $\Delta t = \sqrt{\Delta t_{max} \Delta t_{min}}$
  - Exit Condition: If $\left|1 - \frac{\Delta t_{max}}{\Delta t_{min}}\right| < 0.05$:

    End Iteration

This iteration procedure was performed for 5 different values of the Young's modulus and 11 values of the minimum length scale, constituting 55 independent iteration procedures for each mesh. To remove the effects of software parallelisation from these measurements, we performed all simulations on a single core. The results of the iterative procedures for each value of E, l_{min} and each mesh structure are presented in Figures 3.9, 3.10 and 3.11

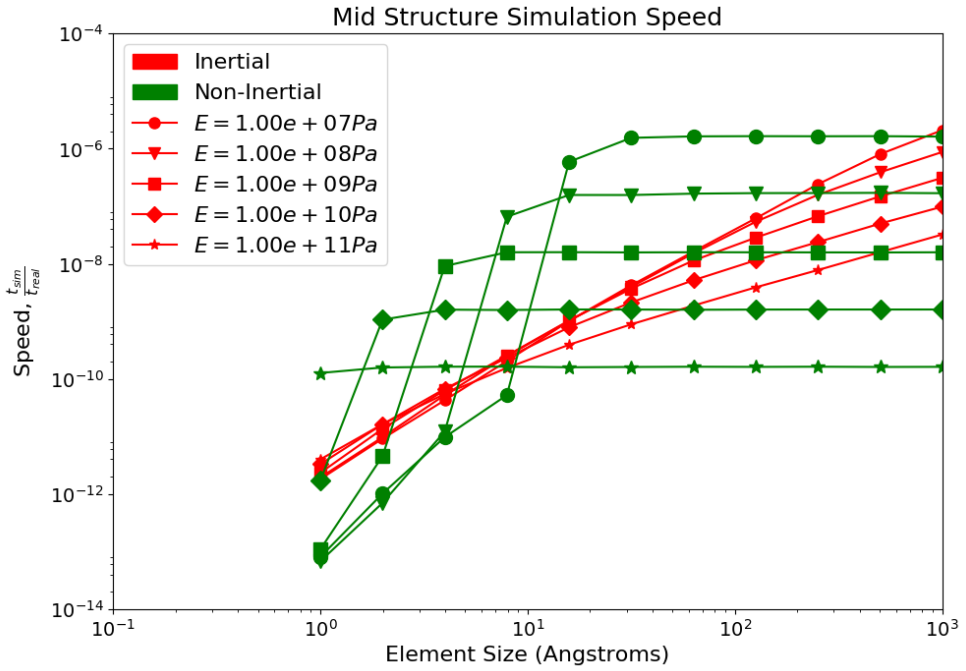


FIGURE 3.10: Maximum simulation speed for a 93 element cubic object as a function of the smallest element size, for a variety of different Young's Moduli

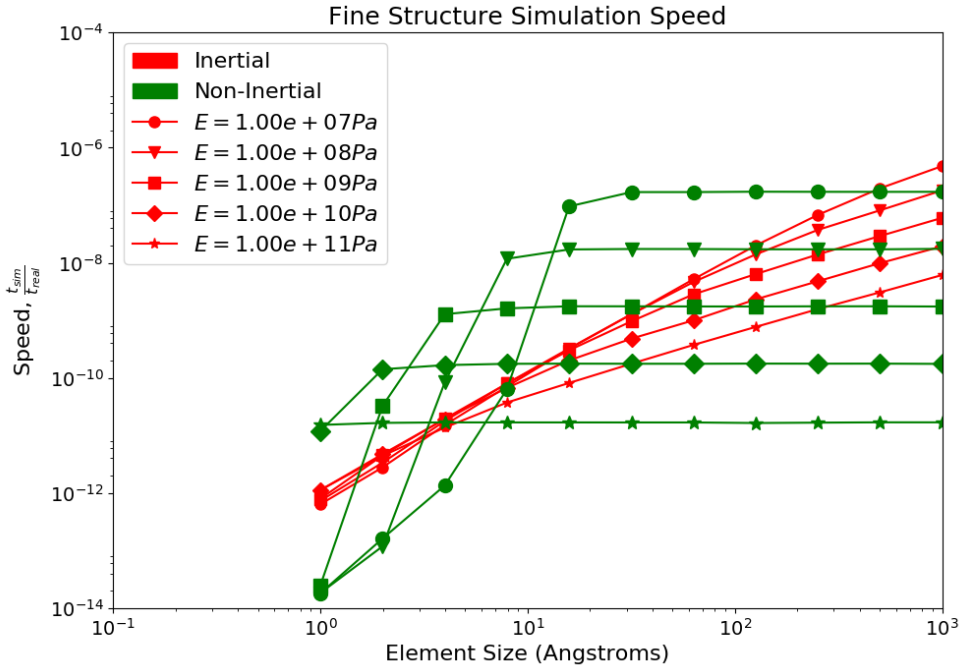


FIGURE 3.11: Maximum simulation speed for a 678 element cubic object as a function of the smallest element size, for a variety of different Young's Moduli

Let us begin by considering only the coarse structure, Figure 3.9, and within that, the single inertialess graph corresponding to $E = 1 \times 10^7$Pa. We see an almost linear initial increase in the maximum speed as we increase the value of l_{min}. This is the stable inversion regime, in that individual thermal fluctuations are of the same magnitude as l_{min} and so we are limited in our value of Δt, and hence the simulation speed, by the size of the smallest element in the system. However, the graph comes to a plateau and for all increases of l_{min} after that point, the simulation cannot sustain a larger time step. This is the unstable inversion regime, where the numerical instabilities iteratively build up and cause element inversion even though the individual thermal fluctuations are much smaller than the size of the element. For the values of E, which increase by a factor of exactly 10, the plateau in the simulation speed decrease by exactly the same factor. As the time scale $\tau_k \propto E^{-1}$, this is exactly the behaviour we would expect if the simulation speed is dependent only on the size of the time step, which must be lowered by exactly the same factor again to retain numerical stability. Hence, we can conclude from just the inertialess graphs that the simulation runtime, and by extension the efficiency of the conjugate gradient solver, is not dependent upon the magnitude of the elasticity parameters.

Let us now consider the inertial simulations. From Equation 3.34, we define the smallest eigenvalues from each of the inertial and viscous time constant matrices as τ_m and τ_k respectively. Equation 3.34 shows us that there are actually two ways for numerical instabilities to occur within an inertial solution protocol,

$$\begin{aligned} 1 - \frac{\Delta t}{\tau_m} &> 0, \\ \Delta t &> \tau_m, \end{aligned} \tag{3.52}$$

$$\begin{aligned} 1 - \frac{\Delta t^2}{\tau_m \tau_k} &> 0, \\ \Delta t^2 &> \tau_m \tau_k. \end{aligned} \tag{3.53}$$

For small systems, which correspond to smaller values of l_{min}, τ_m will generally be much smaller than τ_k. Therefore, it is Equation 3.52 that is responsible for numerical instabilities in the small l_{min} regime. τ_m is independent of the elasticity parameters, and in Figure 3.9 we see that the maximum speed for all values of the Young's modulus is approximately the same for small l_{min}. However, as the system size increases, so too does the mass of the system (as we have kept the density constant) and hence, the inertial time scale increases. As $\tau_m \propto l_{min}^2$ (see Section 3.2.1), the initial gradient of the

graphs is 2 due to their logarithmic axes. However, τ_k is approximately independent of the length scale of the system and so Equation 3.53 will become the dominant cause of numerical instability as the system size increases. If we define a new time scale $\tau_{mk} = \sqrt{\tau_m \tau_k}$, the geometric average of the two time scales, then we can rewrite Equation 3.53 as,

$$\begin{aligned} 1 - \frac{\Delta t^2}{\tau_{mk}^2} &> 0, \\ \left(1 - \frac{\Delta t}{\tau_{mk}}\right)\left(1 + \frac{\Delta t}{\tau_{mk}}\right) &> 0, \\ \Rightarrow 1 - \frac{\Delta t}{\tau_{mk}} &> 0. \end{aligned} \tag{3.54}$$

As $\tau_{mk} \propto l_{min}$, we see the gradient of each of the inertial graphs tend towards gradients of one on the logarithmic axes. This transition happens at higher values of l_{min} with lower values of the Young's modulus as $\tau_{mk} \propto E^{-0.5}$, so the system size, and therefore the mass, must increase further before τ_{mk} dominates the numerical instabilities.

Finally, we compare the different meshes. The overall pattern we have described up to now is approximately the same for each of the different meshes. We see that there is about an order a magnitude difference in speed between the coarse, mid and fine structure simulation speeds, with the fine structure being the slowest one. As there are no pair-pair interactions in these simulations, we can posit that this difference is simply due to the increase in number of degrees of freedom within the simulations i.e. the conjugate gradient algorithm, performed serially, requires more floating-point calculations for the fine structure simulation each step. It takes ~ 10 times longer for coarse, mid and fine simulations respectively as each has ~ 10 times as many elements as the previous model, which backs up this conclusion.

So should we replace the inertial solution procedure with the FFEA simulation protocol? Oliver *et al.* defined the mesoscale as beginning at ~ 5Å, so we will take that as our limiting case. At this value, we see that for the coarse system, an inertialess system runs faster for all values of E in our range. For the mid system, this is again true. For the fine system however, our simulation with $E = 1 \times 10^{11} Pa$ actually ran slower than the inertial equivalent. But, the value of E at which the inertialess simulation became faster is $E \approx 10$GPa, ~ 10 times larger than our representative value of the biological mesoscale, $E = 1$GPa (see Section 3.2.1). In our worst case scenario, that of a serial simulation of a highly detailed structure with very small features i.e. $l_{min} \approx 5$Å, an inertialess simulation gives us a simulation speed increase, $\frac{v_{sim,in}}{v_{sim,non-in}} \approx 50$ at $E = 1$GPa, a very promising result.

3.7 FFEA Normal Mode Analysis

FFEA is able to model not only the dynamics of individual molecules about their equilibrium structure, but the interactions of multiple large molecules such as steric or van der Waals interactions and completely generalised force-fields [93]. However, we may simply be interested in the range of motions that a specific parametrisation enables for an individual molecule, rather than the global effect of this molecule in its biological environment. We have developed a procedure within FFEA that provides the functionality to calculate the set of elastic and dynamic modes via network modelling [54].

3.7.1 FFEA Linear Elastic Model

As we saw in Section 3.2.1, the equation of motion of an FFEA simulation, Equation 3.10, can be linearised to form a multi-dimensional 2nd order SDE,

$$M_{ij}\ddot{x}_j + \Lambda_{ij}\dot{x}_j + K_{ij}x_j = N_i. \tag{3.55}$$

We also saw in our analysis of cuboid normal modes (Section 3.4) that the matrix $\mathbf{K}$ contains within it all of the information necessary to extract the available elastic normal modes from the structure, which can be written as a set of eigenvalues and eigenvectors. Each eigenvalue from the set, $\vec{e}_\alpha$, can be used to write the following equation,

$$\mathbf{K}\vec{e}_\alpha = k_\alpha \vec{e}_\alpha, \tag{3.56}$$

where the summation convention is not applied, meaning k_α is the corresponding eigenvalue to e_α. Dimensional analysis shows that each eigenvalue has the same units as $\mathbf{K}$, and is therefore an effective elastic constant describing the linear flexibility associated with this particular mode of motion. Each eigenvector has units of length, and so represents the relative motion of each node within this mode.

With insights from Bathe [94], we have been able to implement a method of determining elastic modes as a sub-routine of the FFEA framework. By linking to the Eigen C++ matrix libraries [95], FFEA performs this diagonalisation process in a highly efficient manner and exports the eigenvectors and eigenvalues in a standardised format for any FFEA system. We call this the FFEA Linear Elastic Model (LEM).

We can verify the implementation by comparing the eigenvectors calculated from an LEM with those calculated from PCA analysis of a fully converged trajectory. Using

the cuboid structure from Section 3.4 and the resulting PCA analysis of the simulation, we calculated the inner product of the two PCA and LEM eigen-systems, the matrix of which is shown as a heat map in Figure 3.12. We can see that again, we have obtained an approximate identity matrix, as the two systems are identical. Although still diagonally dominant, the small deviations from identity are due to the full simulation not having achieved dynamical convergence.

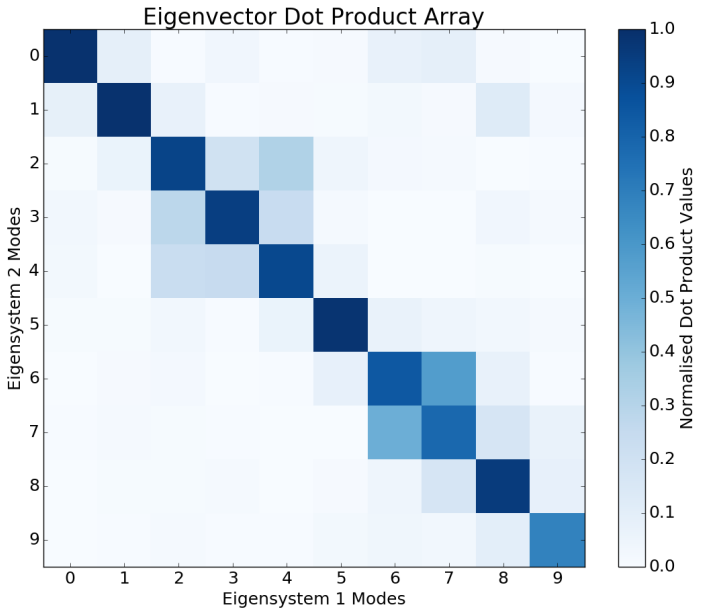


FIGURE 3.12: The inner product of an LEM analysis and PCA analysis of the same system, an elongated cuboid. We can clearly see the dominant feature is the diagonal, showing that the eigen-systems are the same.

We can use the eigenvectors from the LEM analysis to create animations of the modes by noticing that the following form of Equation 3.56 also holds,

$$\mathbf{K}\Delta\vec{e}_\alpha = k_\alpha \Delta\vec{e}_\alpha, \tag{3.57}$$

where $\Delta\vec{e}_\alpha$ represents a small change in the position of each node. Using equipartition, and the fact that k_α is a spring constant, we can write the following equation,

$$l_\alpha = \sqrt{\frac{k_B T}{k_\alpha}}, \tag{3.58}$$

where l_α is the standard deviation of the fluctuations associated with k_α. Using this characteristic length scale we can create an animation with each frame a multiple of $\vec{e}_\alpha$, with the amplitude of the motion within the animation made to be $\sim l_\alpha$. We use

20 frames to make the animation, which in our experience gives a clear idea of the motion with good spatial resolution.

We do note that this is simply a reimplementation of a standard Elastic Network Model as described in Section 1.2.3.1. However, the following two sections show how the additional mechanical considerations taken into account within FFEA allow more sophisticated normal mode calculations to be performed.

3.7.2 FFEA Dynamic Mode Model

Due to the visco-elastic nature of biological systems, local effects of viscous damping can modify the elastic modes of the system. For example, if we have a long, thin molecule which has homogeneous elastic parameters but highly inhomogeneous internal viscosity, then over any period of time the less viscous side will move much more quickly than the more viscous side, and the relative amounts of motion we see over time will not be the same as the motion theoretically available to the system due to the elasticity. So if we take time into account as well as position, and wish to know the relative frequency of motion in time as well as positional variance, then we should decompose our system into modes that take the viscosity into account as well as the elasticity. We refer to these modes as the *dynamic* modes.

In a similar fashion to Section 3.7.1, we linearise the system, but we keep the viscosity component of the equation, giving us the following equation,

$$\Lambda_{ij}\dot{x}_j + K_{ij}x_j = N_i. \tag{3.59}$$

In order to determine the modes due to the combination of the viscous and elastic components, we cannot simply invert the viscosity matrix and diagonalise the resultant matrix, as we would get the following equation,

$$\dot{x}_i + \Lambda^{-1}_{ik}K_{kj}x_j = \Lambda^{-1}_{ij}N_j. \tag{3.60}$$

The RHS of Equation 3.60 forms a vector of *coupled* thermal noise components with statistical moments which do not correspond to the fluctuation-dissipation theorem as applied to an FFEA system. Hence, the system of linear equations remain inseparable and so do not represent truly independent normal modes. In order to generate truly independent dynamical modes, we must find a new set of coordinates y_i for which the eigen-systems of $\mathbf{L}$ and $\mathbf{K}$ can be combined to form a system of independent linear equation.

We begin by diagonalising $\mathbf{\Lambda}$ to find its own independent eigen-system,

$$\mathbf{\Lambda} = \mathbf{e}_\Lambda \boldsymbol{\lambda}_\Lambda \mathbf{e}_\Lambda^{-1}, \tag{3.61}$$

where e_{ij}^{Λ} is the matrix of eigenvectors and λ^{Λ} is the diagonal matrix of eigenvectors for the viscosity matrix. We can then form the diagonal matrix $\mathbf{Q}$ such that,

$$Q_{ij} = \begin{cases} \lambda_{\Lambda,i} & i = j \\ 0 & i \neq j \end{cases}$$

This definition leads us to the following identity,

$$\mathbf{Q^T} \mathbf{e}_\Lambda^T \mathbf{\Lambda} \mathbf{e}_\Lambda \mathbf{Q} = \mathbf{I}, \tag{3.62}$$

which will be of use later on. Our next definition is the matrix $\hat{\mathbf{K}}$, defined as,

$$\hat{\mathbf{K}} = \mathbf{Q^T} \mathbf{e}_\Lambda^T \mathbf{K} \mathbf{e}_\Lambda \mathbf{Q}. \tag{3.63}$$

As $\mathbf{K}$ and $\mathbf{Q}$ are symmetric, and $\mathbf{e}_\Lambda$ forms an orthonormal basis, we note that $\hat{\mathbf{K}}$ itself is symmetric, and so can be diagonalised,

$$\hat{\mathbf{K}} = \mathbf{e}_K \boldsymbol{\lambda}_K \mathbf{e}_K^{-1}, \tag{3.64}$$

which forms another set of eigenvectors and eigenvalues. It is relatively intuitive that a combination of the two eigen-systems of the viscosity and elasticity matrices is necessary to form the dynamic mode eigen-system, so we define the final matrix $\mathbf{R}$,

$$\mathbf{R} = \mathbf{e}_\Lambda \mathbf{Q} \mathbf{e}_K, \tag{3.65}$$

and note the following identities,

$$\begin{aligned} \mathbf{R^T} \mathbf{\Lambda} \mathbf{R} &= \mathbf{I}, \\ \mathbf{R^T} \mathbf{K} \mathbf{R} &= \boldsymbol{\lambda}_K. \end{aligned} \tag{3.66}$$

Applying $\mathbf{R^T}$ to Equation 3.59,

$$\mathbf{R}^T \mathbf{\Lambda} \dot{\vec{x}} + \mathbf{K} \vec{x} = \mathbf{R}^T \vec{N}. \tag{3.67}$$

Finally, we can see how we must transform the coordinate system in order to find the dynamic normal modes,

$$\vec{x} = \mathbf{R}\vec{y}, \tag{3.68}$$

and so substituting into Equation 3.67 gives,

$$\mathbf{R}^T\mathbf{\Lambda}\mathbf{R}\dot{\vec{y}} + \mathbf{K}\mathbf{R}\vec{y} = \mathbf{R}^T\vec{N}, \tag{3.69}$$

and with the identity from Equation 3.66 we finally obtain

$$\dot{\vec{y}} + \mathbf{\lambda}_K\vec{y} = \mathbf{R}^T\vec{N}. \tag{3.70}$$

With respect to the new rotated coordinate system $\vec{y}$, the statistical moments of the effective thermal noise components $F_i = R^T_{ij}N_j$ are spatio-temporally independent,

$$\begin{aligned}\langle F_i(t)F_j(t')\rangle &= R^T_{ik}R^T_{jl}\langle N_kN_l\rangle,\\ &= R^T_{ik}R^T_{jl} \times 2k_BT\lambda_{kl},\\ &= 2k_BTR^T_{ik}\lambda_{kl}R_{lj}.\end{aligned} \tag{3.71}$$

Equation 3.66 shows that the matrices in Equation 3.71 reduce to the identity matrix, and so,

$$\langle F_i(t)F_j(t')\rangle = 2k_BT\delta_{ij}\delta(t-t'), \tag{3.72}$$

where the two delta functions show the required spatio-temporal independence between thermal degrees of freedom.

The numerical construction of the matrix $\mathbf{R}$ is relatively straightforward using the Eigen C++ libraries, and from there, diagonalisation gives the dynamic normal modes and associated eigenvalues for the coordinates $\vec{y}$. Using Equation 3.68 we can transform back to the regular spatial coordinates $\vec{x}$ used within FFEA to visualise the dynamic modes in the same manner as the elastic modes.

3.7.3 FFEA Time Scale Calculator

Given the large overall variation in possible time steps due to their dependence on the specific system to be modelled, we have found it useful to get a rough idea of what the time scales within an FFEA system are in advance of a simulation. Given the inertial equation of motion with linearised elasticity, Equation 3.55, there exist two

time constant matrices,

$$\boldsymbol{\tau}_m = \boldsymbol{\Lambda}^{-1}\mathbf{M},$$
$$\boldsymbol{\tau}_k = \mathbf{K}^{-1}\boldsymbol{\Lambda}.$$

Independent diagonalisation of these matrices provides the spectra of time constants associated with each type of motion, inertial and viscous. Again building and diagonalising these matrices using the Eigen C++ libraries, we can obtain the full spectrum of time constants relatively quickly for even the largest of FFEA systems by exploiting the sparsity of the matrices, using optimised Eigen algorithms for sparse matrix diagonalisation. The largest time-constant from either matrix tells us how long we must run our simulation to achieve full dynamical convergence, and the smallest gives us an idea of how small to make our integration time step to avoid numerical instabilities within a full simulation. However, this approach cannot determine whether or not stable element inversion will occur (see Section 3.6) as this is not a mechanical problem, rather, it is a geometric issue.

3.8 Further Additions to FFEA

3.8.1 Steric Interactions and Lennard-Jones

We saw in Chapter 2 that FFEA includes a procedure for modelling Van der Waals interactions. These interactions inherently include steric forces through the hard-core repulsive term between the two surfaces. Although realistic, in practise we have found that these hard-core interactions are prohibitively numerically unstable in that as the potential $V_{LJ} \to \infty$ with $r \to 0$, where r is the distance between the Gaussian integration points on each surface, we are unable to take large simulation timesteps due to the huge forces involved. This causes us to lose a major new benefit of FFEA, the ability to take long timesteps due to the over damped nature of the system (see Section 3.2). Solernou *et al.* have since developed a softer potential through a volume-volume interaction, which lessens the constraints on the system by allowing volume overlap. The local gradient of volume overlap is the metric used to scale the steric force repelling the two surfaces. This force is then applied to the two surfaces in the

direction of the gradient itself [93]. Although steric interactions are well modelled[5], the formation doesn't allow for a non-specific potential at the surface-surface interface, a required feature for an accurate representation of inter-molecular interactions. To account for this, we have developed an interpolative potential to model the steric interactions at close range using the volume overlap method, and a standard 6-12 potential at intermediate ranges.

We implement the function by considering two parallel surfaces being moved into contact with one another. For the two surfaces, separated by a single distance r with an energy minimum ϵ at r_{eq}, we have the following piecewise functional form

$$\begin{cases} \text{Volume Overlap Potential} & r < 0 \\ \text{Intermediate} & 0 < r < r_{eq} \\ \text{6-12 potential} & r > r_{eq} \end{cases}$$

where we must define the intermediate potential to interpolate between the two limiting regimes. The form of the intermediate function should be continuous in V and also in the first derivative, $\frac{dV}{dr}$, so that the forces are continuous as well as the energy. As such, we require the following boundary conditions to hold. To smoothly transition into the volume overlap function, we must have $V = 0$ and $\frac{dV}{dr} = 0$ at $r = 0$. To smoothly transition into the attractive component of a Lennard-Jones 6-12 potential at $r = r_{eq}$, we require $V = -\epsilon$ and $\frac{dV}{dr} = 0$, where ϵ is the potential energy minimum defined by the user. Our intermediate function, V^{int}, must have two turning points to form these boundary conditions and hence, the simplest form is a 3rd order polynomial,

$$\begin{aligned} V^{int} &= ar^3 + br^2 + cr + d, \\ \frac{dV^{int}}{dr} &= 3ar^2 + 2br + c. \end{aligned} \tag{3.73}$$

We can see immediately that $c = d = 0$ from the boundary conditions at $r = 0$. The final functional forms corresponding to the boundary conditions are,

$$\begin{aligned} V^{int} &= \epsilon \left(2 \left(\frac{r}{r_{eq}} \right)^3 - 3 \left(\frac{r}{r_{eq}} \right)^2 \right), \\ \frac{dV^{int}}{dr} &= \frac{6\epsilon}{r_{eq}} \left(\left(\frac{r}{r_{eq}} \right)^2 - \frac{r}{r_{eq}} \right). \end{aligned} \tag{3.74}$$

[5]The FFEA test suite shows that two colliding spheres with a relative velocity 30m/s do not intersect at all over the course of a collision simulation (in the absence of thermal noise) due to the steric potential. They also undergo only 2.70° of rotation, showing the potential to be highly conservative.

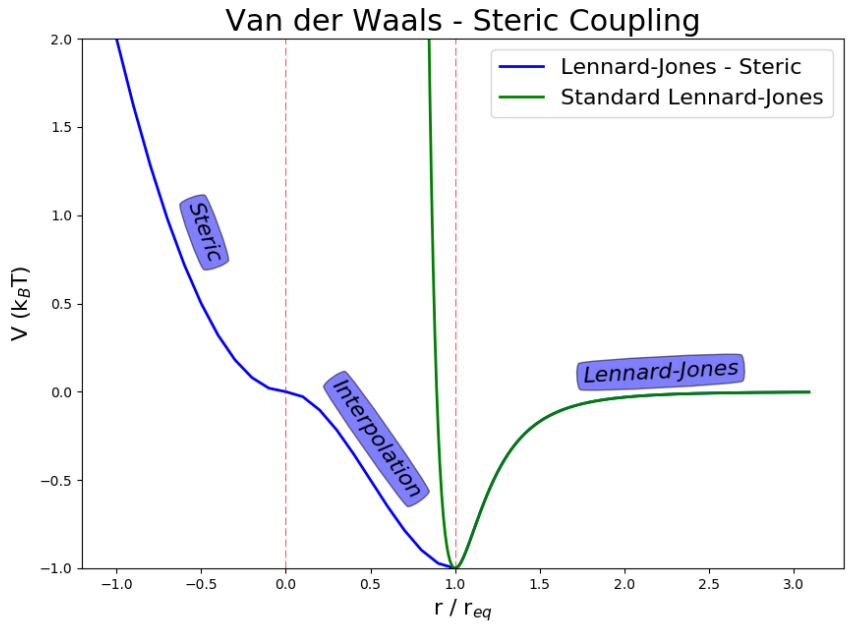


FIGURE 3.13: The new functional form for the Van der Waals potential, formed as an interpolation between the steric interactions at short range and Lennard-Jones interactions at mid-long range. A regular Lennard-Jones form is shown for comparison.

A graph of this potential is shown in Figure 3.13 for the specific case of two parallel surfaces, where we can clearly see the benefit of this softer potential in comparison with the original hard-core repulsion of a standard 6-12 potential. The gradient of the steric potential, and therefore the strength of the repulsion forces, can be tuned manually by the user, but always remains proportional to the gradient of volume overlap between the two structures.

For the general case of non-parallel interacting surfaces, there is no single distance r used to calculate surface-surface separation. As we saw in Section 2.4.2, the set of distances $r(\vec{p},\vec{q}) = |\vec{p}-\vec{q}|$ defined the continuous variation in surface-surface separation as a set of distances between the Gaussian integrations points $\vec{p}$ and $\vec{q}$ on each of the two interacting surfaces. The true potential is therefore a summation of the volume-volume and surface-surface components, which reduces to a pure interpolation for the specific case of parallel interacting surfaces. We therefore calculate the volume-volume steric repulsion in all cases of volume overlap, together with the Lennard-Jones attraction if $r(\vec{p},\vec{q}) > r_{eq}$, or the intermediate repulsive potential if $r(\vec{p},\vec{q}) <= r_{eq}$, between each the different pairs of Gauss points between the surfaces.

We have found through the simulations in this thesis involving van der Waals and steric interactions that this potential is soft enough to retain numerical stability at

even the largest time steps allowed by the time step limitations shown in Section 3.6, and so van der Waals / non-specific interaction potentials do not limit the speed of our simulation by adversely affecting the size of the time step. However, whether the specific geometry of each FFEA mesh, and therefore the layout of Gauss points, has any effect on the shape of the interaction landscape has yet to be studied in detail.

3.9 Summary

We have seen that the over damped nature of the biological mesoscale justifies using a non-inertial equation of motion within a mesoscopic simulation technique such as FFEA. This has allowed us to optimise the overall simulation protocol in terms of speed, taking larger time steps and obtaining an equivalent data set as a result. For interacting molecules, the new steric/Lennard-Jones coupled solver enables the simulation of effective surface-surface interactions without affecting this speed increase through local numerical instabilities. Finally, the network modelling and time scale calculator provide rapid insights of the approximate motion available to a given system. All of these tools together form the core of the FFEA method.

One significant result from the work presented in this thesis is the official publication and release of the FFEA software package to the general public [93]. In addition to the new techniques and models presented in this chapter, a substantial amount of work has been put into the software itself. The package contains a variety of initialisation tools to help a user build their desired simulation environment, and analysis tools which contain automatic methods for calculating and plotting common values of interest. We have also developed a visualiser as a plugin to the PyMOL molecular graphics program [96], which enables FFEA systems to be viewed alongside atomistic data, if available, and to interact with all of the built-in tools for analysis in PyMOL.

For software robustness, all of the tools are written using a centralised set of python classes that define the underlying structural components of an FFEA system. Examples include python objects which handle the structural topology, surface topology, material parameter distributions and, perhaps most importantly, the trajectories and measurement data that are produced by an FFEA simulation.

The open-access software is centralised under a git repository, and can be downloaded from `https://bitbucket.org/FFEA/ffea/downloads/` with associated documentation and installation instructions at `http://ffea.readthedocs.io/`.

Chapter 4

A Kinetic Scheme for FFEA

4.1 Kinetics at the Mesoscale

FFEA can currently be used to study protein equilibrium dynamics which, as we saw in Chapter 3, are driven by thermal fluctuations. The conformational space accessible to proteins is a function of their shape and elastic properties, represented in FFEA by a set of bulk and elastic moduli, which define how much a protein can deform away from a well defined equilibrium structure, or *native state* [97]. In addition to this however, many mesoscopic proteins perform their function in a much more directed manner; via the binding and hydrolysis of adenosine triphosphate (ATP), using the released energy to rearrange their internal structure [98]. The energy provided by ATP hydrolysis to the local environment is often much greater than that of the average thermal fluctuations in the same region, and so the local increase in energy can provide access to a much larger conformational space, or more commonly allows the protein to overcome internal energy barriers and rearrange their atomic structure. This process is very obvious in the family of transporter molecular motors, kinesin, myosin and dynein, which use ATP hydrolysis and subsequent structural rearrangement to generate directed bipedal motion (see Chapter 5).

This type of event is not currently modelled within the framework of an FFEA simulation as the dynamics involve the making and breaking of chemical bonds, which requires quantum mechanical considerations. This is of much higher resolution than FFEA was designed to simulate, and yet many of the large protein complexes that we would ideally want to model using FFEA, such as the molecular motors, are macromolecules whose overall function depends upon the large conformational changes which result from ATP hydrolysis. These molecules are fundamentally multi-scale systems,

with core processes acting at both the mesoscale, the regime in which FFEA was designed to simulate, and also at the nanoscale, more suited towards molecular dynamics and quantum mechanical techniques.

If we were to study the dynamics of these events together, we could perhaps employ multiple layers of resolution in a similar manner to QM/MM (Chapter 1). However, to retain the computational advantages of FFEA, we can consider only the free energy changes involved in ATP hydrolysis and other similarly non-equilibrium processes, and instead include these processes within the FFEA framework using kinetic theory.

4.2 Introduction to Kinetic Theory

Consider a physical system with two well defined states, A and B, and some pathway enabling a transition between them, for example, a standard, reversible chemical reaction. It is often the case that the full mechanism by which the reaction occurs is highly complex, yet the information required from a study of the reaction doesn't require knowledge of the mechanism at all. For example, for a given amount of reactant one may simply wish to determine the amount of product that will be formed given certain environmental conditions. In this situation we may ignore the specific atomistic processes involved over the course of the reaction and simply use a kinetic model, which uses the statistics of equilibrated systems to determine the end result of a given reversible reaction. All that is required is prior knowledge of the reaction rates, which can be measured via experimentation.

4.2.1 A Chemical Example - Table Salt

As a practical example, take the traditional example given in basic chemistry lessons for generating table salt,

$$\mathrm{NaOH} + \mathrm{HCl} \rightleftharpoons \mathrm{NaCl} + \mathrm{H_2O}. \tag{4.1}$$

Although the mechanism for the reaction is known, for the sake of example let us assume for now that it is not. Regardless, we are simply be interested in how much product we will obtain given an initial amount of reactant together with some set of environmental conditions. For a given reaction that has reached equilibrium, we can define the presence of reactant and the presence of product as separate, discrete *kinetic states*. From Equation 4.1, we will define state A as NaOH+HCl and state B as

$NaCl+H_2O$. If we assume stoichiometry in the reaction, the equilibrium concentrations of both states can then be obtained using the principle of detailed balance as follows,

$$c_A R_{AB} = c_B R_{BA}, \tag{4.2}$$

where c_A and c_B are the equilibrium concentrations of states A and B respectively, and due to our (assumed) lack of knowledge on the specific molecular mechanisms occurring during the reaction, we are required to include reaction probabilities in the form of statistical averages, the equilibrium reaction rates R_{AB} and R_{BA}, which are approximately constant at equilibrium. In actuality, the rates are a function of the specific molecular details and are therefore typically related to an energy barrier between states. Environmental conditions such as temperature, pressure, and presence of catalysts, as well as reactant concentrations and other factors all contribute to the rates by affecting the ability of the molecular ensembles to acquire enough energy to proceed in the reaction. But in principle for our study of interest, the specific molecular details can be bypassed in favour of the reaction rates, which contain the net result of all of the higher resolution detail and can be measured directly through relatively simple experimentation.

Once the rates are determined, we can solve Equation 4.2 to determine the equilibrium concentrations of both reactant and product. We can assume that the total concentration of initial mixture is known, such that,

$$c_A + c_B = c_T, \tag{4.3}$$

where c_T is the total initial concentration of both product and reactant. Substitution into Equation 4.2 gives,

$$\begin{aligned} c_A &= \frac{R_{BA}}{R_{AB} + R_{BA}} c_T, \\ c_B &= \frac{R_{AB}}{R_{AB} + R_{BA}} c_T. \end{aligned} \tag{4.4}$$

4.2.2 Kinetic Networks

Kinetic theory can be generalised to any number of discrete states forming a connected network. At equilibrium, for any two directly connected states within the network, indexed by i and j, detailed balance requires that,

$$c_i R_{ij} = c_j R_{ji}. \tag{4.5}$$

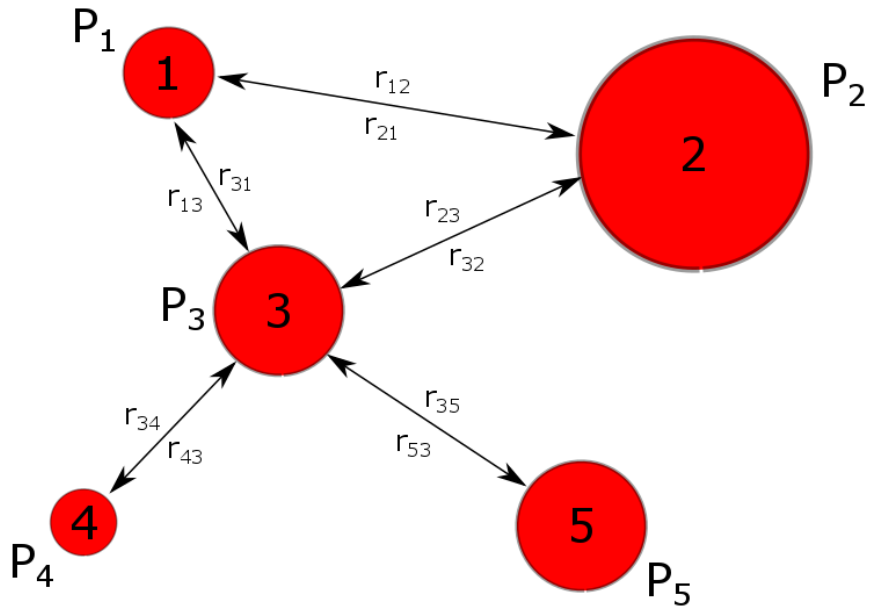


FIGURE 4.1: A visualisation of a generic kinetic network, where the occupancies of each state are visualised by their areas. The detailed balance condition is applicable to every pair of connected states in this network. Although each transition is reversible, we can see a clear inhomogeneity in the occupancies, implying that the transition rates towards state 2 are large, and those towards state 4 are small.

Such a network is illustrated in Figure 4.1.

The reaction rates within these networks can often be individually determined via experimentation. The complete set of rates means that the network corresponds to a fully parametrised Markov chain of events [99] for which we can determine the equilibrium distribution of concentration around the whole network using methods similar to Equation 4.4.

The idea of detailed balance can also be applied to the kinetics of single molecules. By dividing throughout by the total concentration, c_T, we can rewrite Equation 4.5 in terms of occupation probabilities,

$$P_i R_{ij} = P_j R_{ji}, \tag{4.6}$$

where P_i is the probability of being in state i at any given time once equilibrium is reached. For our example network, this represents having only a single state occupied within the system at a time. We can see how this kind of model can be useful by looking at studies of molecular motors.

4.2.3 A Biological Example - Molecular Motors

A set of molecules present within biological organisms responsible for generating force are the molecular motors. Molecular motors feel the effect of thermal fluctuations as all proteins do, and as a result dynamically explore conformational space about a well defined equilibrium configuration. The generation of force is not due to these fluctuations however, but due to internally metabolised ATP providing enough energy to change the equilibrium state itself. These large transitions between equilibrium structures enable the motor to do very specific work against any attached cargo and it is this cycle that generates force and the useful work done by the molecule. Both of these equilibrium structures have been solved to atomic resolution for a variety of motors but what is still largely unknown is exactly how the transitions proceed at the atomic level.

With incomplete structural detail regarding the transition process, a kinetic model is suitable for studying a molecular motor, and indeed, a purely kinetic model of the motor cytoplasmic dynein has been designed by Zhao *et al.* [100]. Cytoplasmic dynein is a highly complex dimer known to be responsible for intra-cellular transport of vital cargoes via a bipedal walking mechanism, with each step driven by an ATP hydrolysis fuelled 'power-stroke'. The 'pre' and 'post' power-stroke equilibrium structures were modelled by Zhao as discrete kinetic states. With very little *ab initio* knowledge of the underlying transition mechanism from pre-powerstroke to post-powerstroke or vice versa, their model is able to calculate a probability distribution of the step size (including direction) of the motor, which indicated a bias towards forward motion with the number of 'hand over hand' steps[1] being ~ 4 times less prevalent than so-called 'inchworm' steps[2]. The authors make a number of assumptions regarding the rates, yet we can still see how powerful a kinetic model is when parametrised only by a small number of transition rates. Each rate can be easily modified within the model to see the global effect of altering the rate at which processes occur, such as ATP hydrolysis (by varying the ATP concentration), or diffusion (by varying the background viscosity). We will perform our own study of cytoplasmic dynein in more detail in Chapter 5.

Following our discussion and development of the dynamical FFEA model in Chapter 3, one might consider whether the thermal fluctuations may have a significant effect on these reaction rates. In cytoplasmic dynein, for example, the structural deformations at equilibrium are relatively large and could therefore undergo associated energy changes that are potentially comparable to that released via ATP hydrolysis. If this is the case,

[1] Steps in which the trailing monomer moves past the leading monomer, or vice versa

[2] Steps in which the trailing monomer merely catches up to the leading monomer, or vice versa, without overtaking.

then the effect of these fluctuations on the kinetic transition rates of single molecules should be considered. The remainder of this chapter concerns the development of a coupled dynamic-kinetic model and its implementation within the FFEA framework.

4.2.4 A Note on Nomenclature

We define the 'kinetics' we have introduced in this chapter as the modelling of occupation probabilities of discrete states defined within a system, and the rates of transition between them. We explicitly distinguish this type of modelling from the inclusion of 'dynamics', by which we mean that the entire trajectory of the system is calculated from underlying, fundamental laws of motion. These two models are mutually exclusive when simulating a specific process, in the sense that if one has enough knowledge about the underlying physical transition process to determine appropriate equations of motion, then the dynamical model can replace the kinetic model. What we can do is use these models in conjunction with one another to retain the advantages of a low spatial resolution dynamical simulation (FFEA, for example) whilst applying kinetic theory to model discrete, atomistic events in a statistical manner, without the need for the inclusion of high resolution detail.

4.3 Modelling Kinetics with Newtonian Dynamics

4.3.1 Reinterpreting FFEA Dynamics

Let us assume that we have a dynamic system which can exist within N different energy minima. Although there must exist some continuous process by which the system can transition from one of these minima to the other, because of the high spatio-temporal resolution required to understand the energy landscape across these transitions, we will consider them to be *mesostates*, states which are fully separated in a dynamic sense but may undergo kinetic transitions from one mesostate to another as described in Section 4.2.4. This type of separability is valid for the formation of table salt in Section 4.2.1, where the reactants are constantly undergoing Brownian motion and the reaction itself can easily be modelled kinetically, but could, in principle, be described by the continuous change in electronic probability described in Section 1.2.1.

Within each of these individual mesostates, i, there exists a continuum of possible internal energies E_i that the population of the mesostate may be in. This energy landscape can be decomposed into a set of unique *microstates*, $(\vec{x}, \vec{p})_i$, characterised

by the set of position vectors $\vec{x}$ and the set of momentum vectors $\vec{p}$. The set of pairs of vectors $\Omega = \{(\vec{x}, \vec{p})\}$ constitutes the phase space of the system such that any point $(\vec{x}, \vec{p})$ in Ω, together with a mesostate i, specifies a unique microstate $(\vec{x}, \vec{p})_i$ within the total free energy landscape. The microstates within any specific mesostate are the familiar, continuous set of states accessible via the thermal fluctuations of Brownian motion. This means that the continuous distribution of internal energy within each mesostate i can be modelled as a Boltzmann distribution, as we assume there to be no interaction between any two mesostates i and j. This gives a probability density function $p(\vec{x}, \vec{p}|i)$, which is the probability of being in configuration $\vec{x}$ with momentum $\vec{p}$ given that we are already in mesostate i,

$$p(\vec{x}, \vec{p}|i) = \frac{1}{Z_i} \exp\left(\frac{-E(\vec{x}, \vec{p})_i}{k_B T}\right), \tag{4.7}$$

where Z_i is the partition function for the mesostate i,

$$Z_i = \int_\Omega \exp\left(\frac{-E(\vec{x}, \vec{p})_i}{k_B T}\right). \tag{4.8}$$

It is this phase space that FFEA was originally designed to explore, with each molecule being confined to a single mesostate i and exploring the subset of microstates within Ω. However, given that the transitions between these mesostates are often vital to the function of mesoscopic proteins, we must build into FFEA the ability to sample the entirety of the global energy landscape as well, including all additional mesostates j.

4.3.2 Microstates and Mesostates

As each mesostate is fully separable in our description, we can define a total partition function Z_T for the entire global energy landscape as a discrete summation of each individual mesostate,

$$Z_T = \sum_{i=1}^{N} Z_i. \tag{4.9}$$

This allows us to define a more general probability density function for the entire system, $p(\vec{x}, \vec{p})_i$, as the probability of being in any microstate $(\vec{x}, \vec{p})_i$,

$$p(\vec{x}, \vec{p})_i = \frac{1}{Z_T} \exp\left(\frac{-E(\vec{x}, \vec{p})_i}{k_B T}\right). \tag{4.10}$$

Integrating over the total phase space Ω will give us the occupation probability for each mesostate i,

$$\begin{aligned} P_i &= \int_{\Omega} p(\vec{x}, \vec{p})_i d\Omega \\ &= \frac{Z_i}{Z_T}, \end{aligned} \tag{4.11}$$

which we can finally define in terms of a free energy, F_i, for the mesostate i,

$$P_i = \frac{1}{Z_T} \exp\left(\frac{-F_i}{k_B T}\right), \tag{4.12}$$

where $F_i = -k_B T \ln(Z_i)$. We have now defined two types of kinetic state within the total system; a mesostate i which may be described in comparison to other mesostates j by their free energies, F_i and F_j, and a microstate $(\vec{x}, \vec{p})_i$ which can be described dynamically within each mesostate i by their microscopic energies $E(\vec{x}, \vec{p})_i$. We now consider how we can use this formulation to build the possibility of kinetic transitions between mesostates into a dynamic simulation.

It is perhaps useful to note at this point that because FFEA is a coarse-grained model, all of the energy contained within the underlying atomic degrees of freedom has been lost. As such, our microstate energies $E_i(\vec{x}, \vec{p})$ are actually free energies themselves, albeit of a higher order than the discrete mesostate values F_i we have defined, with a dynamic FFEA simulation more of a continuous exploration through a free energy space, not an energy space. However, it is intuitively clear to refer to the dynamic part of the model as having a real energy, and the kinetic mesoscale part as having free energy, so this will be our convention for the remainder of this work.

4.3.3 Microscale and Mesoscale Kinetics

We have already seen that detailed balance tells us how reaction rates are intrinsically linked to occupation probabilities in Equation 4.6. Let us rewrite this by substituting in Equation 4.12,

$$\frac{R_{ij}}{R_{ji}} = \exp\left(\frac{-\Delta F_{ij}}{k_B T}\right), \tag{4.13}$$

where $\Delta F_{ij} = F_j - F_i$. Equation 4.13 shows that the reaction rate ratio is dependent only upon the free energy difference between the two mesostates i and j. The assumption is often that because these free energies are constant (at constant temperature), so too are the kinetic transition rates R_{ij}. At higher resolution however, we have seen that these free energies are formed through a combination of the microstates energies

within each mesostate. As Equation 4.10 has a similar form to Equation 4.12, we may postulate a further detailed balance condition between pairs of microstates within the system,

$$p(\vec{x},\vec{p})_i r_{ij} = p(\vec{x},\vec{p})_j r_{ji}, \tag{4.14}$$

$$\frac{r_{ij}}{r_{ji}} = \exp\left(\frac{-\Delta E(\vec{x},\vec{p})_{ij}}{k_B T}\right), \tag{4.15}$$

where $\Delta E_{ij} = E_j - E_i$ and $r_{ij} = r_{ij}(\Delta E_{ij})$ is the transition rate from one microstate to another, or, from mesostate i to mesostate j *whilst also in configuration* $\vec{x}$ *and at momentum* $\vec{p}$. We find that these microscopic rates must be a function of the current energy of the system, as well as the energy of the state it is transitioning into and hence, a function of the position in phase space. These energies are constantly varying as the dynamic system undergoes thermal fluctuations and hence, the reaction rates also vary as a function of the underlying dynamics. This concept of separable mesoscale / microscale kinetics is visualised in Figure 4.2.

We will now begin to explore the relationship between the microscopic rates r_{ij}, and mesoscopic rates R_{ij}, by considering the underlying energetic contributions.

4.3.4 Energy within Microscale Kinetics

As a continuous function of both the configuration, $\vec{x}$, and the momentum, $\vec{p}$, the internal energy $E_i(\vec{x},\vec{p})$ can be divided into a number of constituent parts,

$$\frac{r_{ij}}{r_{ji}} = \exp\left(\frac{-\Delta E^c_{ij} - \Delta E(\vec{x},\vec{p})^{nc}_{ij}}{k_B T}\right), \tag{4.16}$$

where $\Delta E(\vec{x},\vec{p})^{nc}_{ij}$ is the 'non-chemical' component of energy that is fully dependent upon the location in phase space and which determines the dynamics within a mesoscale simulation such as FFEA. ΔE^c_{ij} is an additional, as yet unknown, energy change which we will call the chemical energy, and represents additional energy differences between the mesostates. ΔE^{nc}_{ij} can be further subdivided into the individual energy 'types' that contribute to it,

$$\Delta E(\vec{x},\vec{p})^{nc}_{ij} = \Delta K(\vec{p})_{ij} + \Delta U(\vec{x})_{ij}, \tag{4.17}$$

where ΔK_{ij} is the change in kinetic energy, and ΔU_{ij} the change in potential energy following a kinetic transition. It follows that ΔE^c_{ij} must exist to account for the inherent difference in forward and backward transition rates even at $\Delta K_{ij} = \Delta U_{ij} = 0$. If we assume this to be the chemical energy change due to the making and breaking

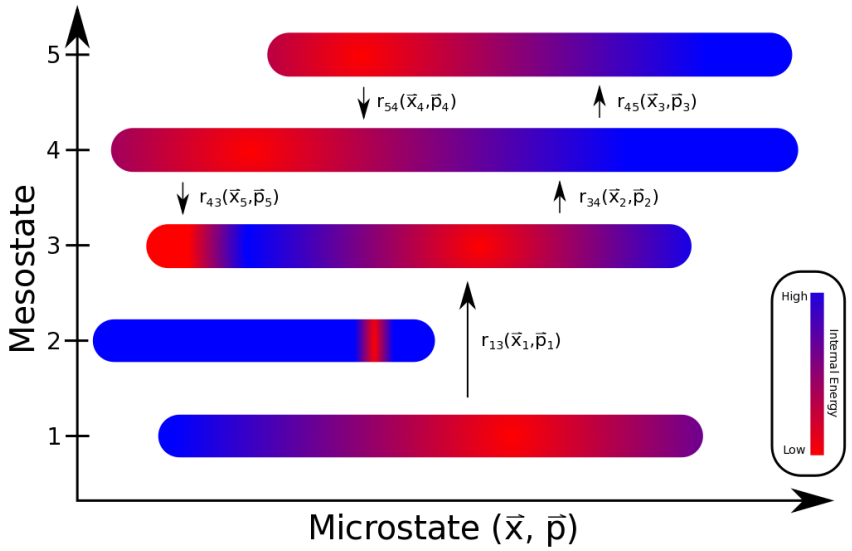


FIGURE 4.2: A representation of the nature of discrete mesostates and continuous microstates. Within each mesostate we can dynamically model the microstate as we have knowledge of the underlying continuous energy landscape from high energy (blue) to low energy(red). However, between the mesostates themselves we have neglected to model, or have no knowledge, of the underlying energy landscape (white) and so we must model them as discrete states with associated kinetic transition rates. We have included an example pathway through this joint internal and free energy landscape: 1. A kinetic transition from a low energy state to a low energy state, $r_{13}(\vec{x}_1, \vec{p}_1)$. 2. A transition from low energy to high energy. We can infer that $r_{34}(\vec{x}_2, \vec{p}_2)$ is relatively small at this point in phase space. 3. A high energy to high energy transition with rate $r_{45}(\vec{x}_3, \vec{p}_3)$. Perhaps counter-intuitively, such a transition is not kinetically suppressed, as the energy *change* is quite small at this point in phase space even though the absolute energies are high. 4. Within mesostate 5, the system has mechanically relaxed to a lower energy (a likely scenario for a Boltzmann energy distribution) before transitioning to the higher mechanical energy mesostate 4 with rate $r_{54}(\vec{x}_4, \vec{p}_4)$. 5. The system passes through the low energy region of phase space with mesostate 4 and then transitions into the (very) low energy mesostate 3 with rate $r_{43}(\vec{x}_5, \vec{p}_5)$. This final state represents a mechanical energy minimum, which, although inhibited within mesostate 3 itself, can be realised through a pathway of intermediate mesostates as we have seen in this example.

of bonds we may also approximate it as constant and can rewrite in terms of specific kinetic rates,

$$\exp\left(\frac{-\Delta E_{ij}^{c}}{k_B T}\right) = \frac{r_{ij}^{0}}{r_{ji}^{0}}, \tag{4.18}$$

and hence, Equation 4.16 simplifies to,

$$\frac{r_{ij}}{r_{ji}} = \frac{r_{ij}^{0}}{r_{ji}^{0}} \exp\left(\frac{-\Delta E(\vec{x},\vec{p})_{ij}^{nc}}{k_B T}\right), \tag{4.19}$$

where r_{ij}^0 is a constant 'base' rate which is modified by the other energy contributions to give overall rates r_{ij}.

4.3.5 Decoupling Detailed Balance

The condition of detailed balance at equilibrium has given us information about the relative sizes of the kinetic rates, but we need a further constraint in order to separate the rates and determine individual expressions for them. We introduce the following generalised form for the individual rates r_{ij},

$$\begin{aligned} r_{ij} &= r_{ij}^0 \exp\left(\frac{-f_{ij}\Delta E_{ij}^{nc}}{k_B T}\right), \\ r_{ji} &= r_{ji}^0 \exp\left(\frac{-f_{ji}\Delta E_{ji}^{nc}}{k_B T}\right), \end{aligned} \tag{4.20}$$

where the summation convention is not in effect, and f_{ij} and f_{ji} are constants specific to each transition between mesostates i and j. We can make Equation 4.20 consistent with Equation 4.19 by meeting the following condition,

$$f_{ij} + f_{ji} = 1. \tag{4.21}$$

The ratio of the two rates will be independent of these values for all energy types and will therefore have the correct dependence on the overall energy change ΔE_{ij} associated with that kinetic process. Although the values of f_{ij} and f_{ji} are no longer independent, we will continue to refer to them both in later equations for clarity. The physical meanings of both f_{ij} and f_{ji} are related to the activation energy, which is analysed in detail in Appendix A.

4.3.6 Coupling the Microscale to the Mesoscale

We are now in a position to determine how the continuously changing microscopic rates r_{ij} result in the experimentally measured mesoscopic rates R_{ij}. The lack of temporal resolution means that experiments measure the average effect of the varying rates r_{ij} as the system quickly moves through phase space. It therefore follows that,

$$R_{ij} = \langle r_{ij} \rangle, \tag{4.22}$$

where $\langle r_{ij} \rangle$ is an average taken over all possible kinetic transitions, weighted by the probability starting in a given microstate, and constitutes the intersection of phase

spaces of mesostates i and j. This average can therefore be written in terms of an integral over the Boltzmann probability distribution from Equation 4.10,

$$P_i R_{ij} = \int_\Omega p_i r_{ij} d\Omega. \tag{4.23}$$

Substitution of Equation 4.10 and Equation 4.20, and making R_{ij} the subject gives,

$$R_{ij} = \frac{r_{ij}^0 \int_\Omega \exp\left(\frac{-E_i^{nc}}{k_B T}\right) \exp\left(\frac{-f_{ij}\Delta E_{ij}^{nc}}{k_B T}\right) d\Omega}{\int_\Omega \exp\left(\frac{-E_i^{nc}}{k_B T}\right) d\Omega}, \tag{4.24}$$

where we have assumed r_{ij}^0 to be truly constant, and independent of both $\vec{x}$ and $\vec{p}$. Equation 4.24 can be simplified and rearranged for the base rate r_{ij}^0,

$$r_{ij}^0 = R_{ij} \frac{\int_\Omega \exp\left(\frac{-E_i^{nc}}{k_B T}\right) d\Omega}{\int_\Omega \exp\left(\frac{-\left(f_{ji}E_i^{nc} + f_{ij}E_j^{nc}\right)}{k_B T}\right) d\Omega}. \tag{4.25}$$

We can now, in principle, calculate the underlying rate r_{ij}^0 that gives rise to the measured rate R_{ij} as a result of thermal exploration of conformational space. If we follow the same derivation procedure for the reverse rates r_{ji}^0, we find that,

$$r_{ji}^0 = R_{ji} \frac{\int_\Omega \exp\left(\frac{-E_j^{nc}}{k_B T}\right) d\Omega}{\int_\Omega \exp\left(\frac{-\left(f_{ji}E_i^{nc} + f_{ij}E_j^{nc}\right)}{k_B T}\right) d\Omega}. \tag{4.26}$$

Notice that the denominator integrals in Equation 4.25 and Equation 4.26 are identical. As such, we need only calculate it once for any given transition, and with knowledge of the mesostate transition rates we can write a simpler relation for the reverse base rate, r_{ji}^0,

$$r_{ji}^0 = r_{ij}^0 \frac{R_{ji}}{R_{ij}} \frac{\int_\Omega \exp\left(\frac{-E_j^{nc}}{k_B T}\right) d\Omega}{\int_\Omega \exp\left(\frac{-E_i^{nc}}{k_B T}\right) d\Omega}, \tag{4.27}$$

which completes the coupling between the mesoscale and microscale.

From a set of measured kinetic rates R_{ij} together with appropriate values of f_{ij}, we can now calculate the underlying rates r_{ij}^0 that are unmodified by thermal exploration of conformational space. The values of f_{ij} used within this work are assumptions based upon the general arguments presented in Appendix A. The justification of each value will be briefly explained prior to its use.

With the full set of base rates, at any time during a dynamical simulation we can

calculate the modified rate $r_{ij}(\Delta E_{ij}^{nc})$ as the system moves through phase space. We can therefore couple this kinetic scheme together with any underlying dynamical model, explicitly reintroducing the thermal fluctuations, and use the resulting mechanical energies to calculate the effect of conformational exploration on the kinetic processes. Most importantly, we can switch between the two mesostates at any time within the dynamical framework and perform kinetic transitions in real time within a simulation environment. This provides us with a very high level of temporal resolution when studying kinetic processes in biological systems. Before we look at integrating this kinetic scheme into the FFEA framework, which has its own complications, we will first apply it to a simpler one-dimensional model for which we can calculate the exact solution.

4.4 A One-Dimensional Example - The Dumbbell Model

We validate the coupled dynamic-kinetic model using a one-dimensional bead spring model, otherwise known as a dumbbell model. This model comprises a single Hookean spring connecting two spherical objects together (see Figure 4.3).

4.4.1 Bead - Spring System: Dynamics

Each of our point particles i have mass m_i and position x_i. Immersed in a background fluid, they experience a drag force λ_i and due to the fluctuation-dissipation theorem, they must both experience a stochastic thermal force N_i in order to satisfy the equipartition theorem and give each particle the correct energy distribution [21].

Assuming our system is over-damped at the biological mesoscale, we may solve for an inertialess system. The equations of motion for these particles are,

$$\lambda_1 \frac{dx_1}{dt} - k(x_2 - x_1 - l) = N_1, \tag{4.28}$$

$$\lambda_2 \frac{dx_2}{dt} + k(x_2 - x_1 - l) = N_2. \tag{4.29}$$

These two coupled equations can be numerically integrated in parallel using a one step Eulerian integration scheme, such that the numerical update following a time step Δt

is,

$$x_1(t+\Delta t) = x_1(t)\left(1-\frac{\Delta t}{\tau_{k,1}}\right)+\frac{\Delta t}{\tau_{k,1}}(x_2-l)+\frac{\Delta t}{\lambda_1}N_1,$$
$$x_2(t+\Delta t) = x_2(t)\left(1-\frac{\Delta t}{\tau_{k,2}}\right)+\frac{\Delta t}{\tau_{k,2}}(x_1+l)+\frac{\Delta t}{\lambda_2}N_2, \tag{4.30}$$

where $\tau_{k,1} = \lambda_1/k$ and $\tau_{k,2} = \lambda_2/k$ are the time constants associated with the viscous motion of each particle. Keeping in mind that for numerical stability and accuracy, $\Delta t \ll \tau_{k,\alpha}\ \forall\ \alpha$, the above equations of motion can easily be implemented and iteratively solved within a computer simulation and, if necessary, visualised to produced dynamical trajectories of the system.

In regard to the motion of the overall dumbbell, we sum Equation 4.28 and Equation 4.29 to obtain,

$$\lambda_T\frac{dx_c}{dt} = N_T, \tag{4.31}$$

where $\lambda_T = \lambda_1+\lambda_2$, $N_T = N_1+N_2$ and we define x_c, the centre of drag, as,

$$x_c = \frac{x_1\lambda_1+x_2\lambda_2}{\lambda_T}. \tag{4.32}$$

Equation 4.31 represents a purely diffusive object which feels the background drag and associated noise from both of the constituent beads. We therefore expect the diffusion to be measured as,

$$\langle(x_c-x_{c,0})^2\rangle = 2Dt, \tag{4.33}$$

where the diffusion constant $D = k_BT/\lambda_T$.

Subtraction of Equation 4.28 from Equation 4.29 instead gives,

$$\lambda_r\frac{dx'}{dt}+k(x'-l) = \frac{\lambda_r}{\lambda_2}N_2-\frac{\lambda_r}{\lambda_1}N_1, \tag{4.34}$$

where $x' = x_2-x_1$ and λ_r, the reduced drag, is defined as follows,

$$\lambda_r = \frac{\lambda_1\lambda_2}{\lambda_1+\lambda_2}. \tag{4.35}$$

Equation 4.34 is a standard Brownian equation of motion for the single extensional degree of freedom x', and as such we expect the variance in length to be,

$$\langle(x'-l)^2\rangle = \frac{k_BT}{k}, \tag{4.36}$$

in the limit of a long time simulation, $t \gg \lambda_r/k$ as per the equipartition theorem.

4.4.2 Bead - Spring System: Kinetics

The dynamical simulation method we have developed for the bead-spring system corresponds to the microscale dynamics we defined in Section 4.3.1. This continuous exploration of conformational space could be seen to represent a single kinetic mesostate, a single energy well which is separated from other accessible wells by a free energy (see Figure 4.2). Let us define a set of kinetic states α which correspond to different instances of the material constants k and l, such that, $k \to k_\alpha$ and $l \to l_\alpha$. It may help to envisage a transition from state α to another state β as a biological molecule undertaking an ATP driven internal rearrangement of structure which in turn gives rise to different material properties due to differing bond structures. We could also 'upgrade' the parameters λ_1 and λ_2 to be varying kinetic parameters as well, but for simplicity we will leave them as constant across the kinetic landscape.

As we are modelling the system as over-damped, the total energy within any state α, $E_\alpha = E_\alpha(x)$, is a function of position only, and not momentum. With no work done on or by the system either, the energy landscape can be described using only potential energies. From Equation 4.17, it follows that,

$$E(\vec{x}, \vec{p})^{nc}_\alpha = U(x')_\alpha, \tag{4.37}$$

where,

$$U(x')_\alpha = \frac{1}{2} k_\alpha (x' - l_\alpha)^2. \tag{4.38}$$

We define a discrete kinetic transition to be an instantaneous change from $k_\alpha \to k_\beta$ and $l_\alpha \to l_\beta$, whilst keeping the spring extension constant, which gives a change in potential energy,

$$\Delta U(x')_{\alpha\beta} = \frac{1}{2} \left(k_\beta (x' - l_\beta)^2 - k_\alpha (x' - l_\alpha)^2 \right). \tag{4.39}$$

Assuming we have a set of global rates $R_{\alpha\beta}$, we can analytically solve for the corresponding base rates $r^0_{\alpha\beta}$. An integral over all microstates with a given mesostate is equivalent to an integral over the single spatial degree of freedom x'. As such, the numerator integral of Equation 4.25 is a simple Gaussian integral,

$$\begin{aligned} \int_\Omega \exp\left(\frac{-U_\alpha}{k_B T}\right) d\Omega &= \int_{-\infty}^{\infty} \exp\left(-\frac{k_\alpha (x' - l_\alpha)^2}{2 k_B T}\right) dx', \\ &= \sqrt{\frac{2\pi k_B T}{k_\alpha}}. \end{aligned} \tag{4.40}$$

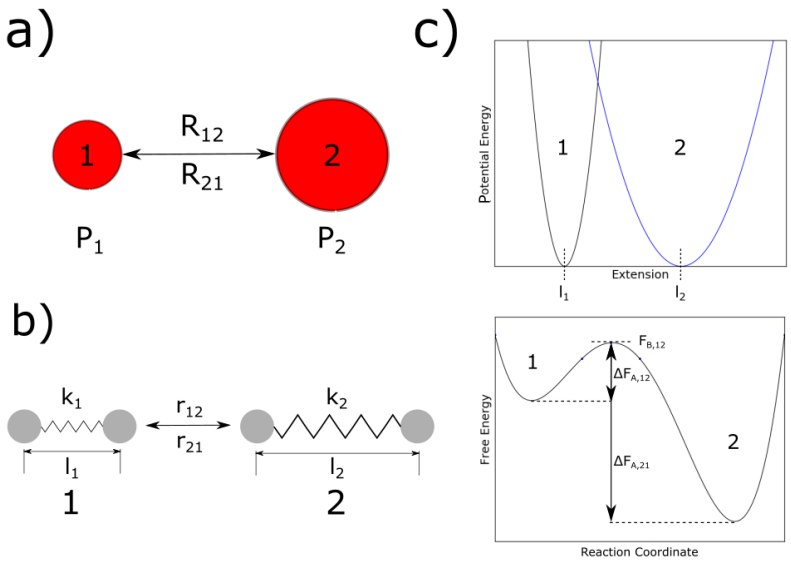


FIGURE 4.3: A visualisation of a 1D kinetic system with internal dynamics. **a)** shows two different kinetic states, with no additional knowledge of the underlying dynamics. $R_{\alpha\beta}$ and $R_{\beta\alpha}$ appear to be constant rates at this scale. **b)** shows the addition of underlying dynamic information. Each kinetic state is actually a spring connecting two point masses, but each spring has a different spring constant, k_α, and equilibrium length, l_α. At a certain rate $r_{\alpha\beta}$, the system may switch from state α to state β which means a change in the parameters: $k_\alpha \to k_\beta$ and $l_\alpha \to l_\beta$. The potential energy change due to this change affects these microscopic rates, showing that the mesoscale rates are in fact averages. **c)** shows one possible set of potential and free energy landscapes associated with the system, with activation free energies emerging as ensemble averages of the energy modifications to the base rates $r^0_{\alpha\beta}$.

The denominator integral is slightly more complex but still forms a Gaussian integral,

$$\begin{aligned}
&\int_\Omega \exp\left(\frac{-f_{\beta\alpha}U_\alpha - f_{\alpha\beta}U_\beta}{k_BT}\right) d\Omega, \\
&= \int_{-\infty}^{\infty} \exp\left(-f_{\beta\alpha}\frac{k_\alpha(x'-l_\alpha)^2}{2k_BT}\right) \exp\left(-f_{\alpha\beta}\frac{k_\beta(x'-l_\beta)^2}{2k_BT}\right) dx', \\
&= \sqrt{\frac{2\pi k_BT}{f_{\alpha\beta}k_\beta + f_{\beta\alpha}k_\alpha}} \exp\left(\frac{f_{\alpha\beta}f_{\beta\alpha}k_\alpha k_\beta(l_\alpha - l_\beta)^2}{2k_BT\left(f_{\alpha\beta}k_\beta + f_{\beta\alpha}k_\alpha\right)}\right).
\end{aligned} \tag{4.41}$$

Substituting Equation 4.40 and Equation 4.41 into Equation 4.25, and replacing $f_{\beta\alpha}$ with $f_{\alpha\beta}$ from Equation 4.21, gives us the base transition rate $r^0_{\alpha\beta}$ in terms of the measured average rate $R_{\alpha\beta}$ and the material properties of the two kinetic states. For clarity, we make $R_{\alpha\beta}$ the subject,

$$R_{\alpha\beta} = r^0_{\alpha\beta}\sqrt{\frac{1}{f_{\alpha\beta}\left(\frac{k_\beta}{k_\alpha}-1\right)+1}} \exp\left(-f_{\alpha\beta}(1-f_{\alpha\beta})\frac{k_\alpha k_\beta(l_\alpha - l_\beta)^2}{2k_BT(k_\alpha + f_{\alpha\beta}\left(k_\beta - k_\alpha\right))}\right). \tag{4.42}$$

Recall that $R_{\alpha\beta}$, the measured, average rate, is being mechanically modified from an underlying rate $r^0_{\alpha\beta}$ due to the different mechanical energies accessed via thermal fluctuations. If the change from mesostate α to mesostate β more often involves an increase in mechanical energy, then we would expect $R_{\alpha\beta} < r^0_{\alpha\beta}$. The opposite is true if the typical case is a decrease in energy. We can check that Equation 4.42 has all of these expected properties by considering the behaviour for certain types of transition.

First of all, we will consider the behaviour of Equation 4.42 when the domain of $f_{\alpha\beta}$ is restricted to $0 < f_{\alpha\beta} < 1$. In this case, the value of $f_{\alpha\beta}$ does not affect the sign of the exponent in the exponential term. For this range of $f_{\alpha\beta}$, there exist no combinations of $k_\alpha, k_\beta, l_\alpha$ and l_β that would generate a change in sign for the exponent term. Hence, this term always acts to reduce the measured rate $R_{\alpha\beta}$ with respect to the base rate $r^0_{\alpha\beta}$.

If we set $k_\alpha = k_\beta = k$, then,

$$R_{\alpha\beta} = r^0_{\alpha\beta} \exp\left(-f_{\alpha\beta}(1 - f_{\alpha\beta})\frac{k(l_\alpha - l_\beta)^2}{2k_B T}\right), \tag{4.43}$$

which shows that the base transition rate is modified only by a Boltzmann factor, with a potential energy change due to the difference in equilibrium lengths only. The symmetry of the potential energy about the equilibrium length for any pair of values l_α and l_β means that the change from l_α to l_β following a kinetic transition will cause an increase in energy for the majority of locations in phase space. Therefore, for $k_\alpha = k_\beta$, we find $R_{\alpha\beta} \leq r^0_{\alpha\beta}$. This energy difference corresponds to the relative position on the x-axis of the two potential energy graphs seen in Figure 4.3c.

If instead we set $l_\alpha = l_\beta = l$, then,

$$R_{\alpha\beta} = r^0_{\alpha\beta}\sqrt{\frac{1}{f_{\alpha\beta}\left(\frac{k_\beta}{k_\alpha} - 1\right) + 1}}. \tag{4.44}$$

In this case, we lose the Boltzmann factor and the rate is modified only by the fractional difference in spring constants, for which the energy difference corresponds to the difference in curvature between the two potential energy graphs seen in Figure 4.3c. If $k_\beta > k_\alpha$ then our target state β would always represent a higher mechanical energy state and so $R_{\alpha\beta} < r^0_{\alpha\beta}$. The opposite is true for $k_\beta < k_\alpha$. Here we also note that the restriction of the domain of $f_{\alpha\beta}$ prevents the square-root argument from becoming negative (see Appendix A). For completeness, if we implement both of the above

conditions from Equations 4.43 and 4.44, then,

$$R_{\alpha\beta} = r^0_{\alpha\beta}, \tag{4.45}$$

showing us that if our states are completely identical, then the average rate and the base rate are equal due to there being no potential energy modification between the two states. The two potential energy graphs seen in Figure 4.3c would be perfectly overlapped in this case.

We can now use the calculated base rate from Equation 4.42 together with Equation 4.20 to determine an instantaneous kinetic rate as a function of the potential energy at any point in the dynamical trajectory which will give the correct statistical average $R_{\alpha\beta}$ over a long enough time.

4.4.3 Bead - Spring System: Coupled Kinetics

At any point during the dynamical simulation described in Section 4.4.1 we can calculate the potential energy of the system, $U_\alpha(x)$, and the potential energy which the system would have if it were in the other state, $U_\beta(x)$. Using these energy values and our pre-calculated base rates $r^0_{\alpha\beta}$ we can calculate an instantaneous value for the transition rate as the simulation progresses using Equation 4.20. Transforming these transition rates to transition probabilities per time step, $\pi_{\alpha\beta}$, is as follows,

$$\pi_{\alpha\beta}(x) = r_{\alpha\beta}(x)\Delta t, \tag{4.46}$$

where Δt is the simulation time step. By testing a number, n, drawn from a uniform random number generator on the range $\{n \mid 0 \leq n \leq 1\}$ that has been partitioned into bins based on these probabilities, we can change the system parameters to those of a different state at appropriate times during the dynamical simulation and continue with no interruption. The way in which we calculated our base rates ensures that detailed balance holds at both the mesoscopic level as well as the microscopic whilst at all times, a dynamical simulation is happening in parallel.

4.4.4 Bead - Spring System: Results

Before the inclusion of the new coupled kinetics, we verified that the dynamical simulation works as a standalone simulation by performing a number of standard tests on the bead-spring system. Our two beads, each with an associated drag $\lambda = 5.65\text{kg.s}^{-1}$

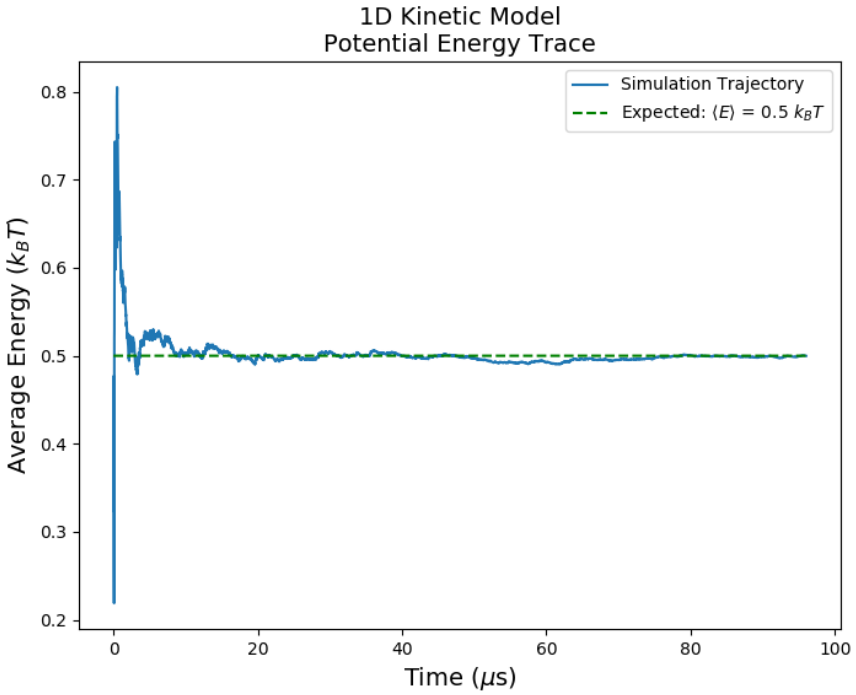


FIGURE 4.4: The average potential energy trace of the 1D dumbbell model with no coupled kinetics.

are connected by a spring with spring constant $k = 1.8849 \times 10^{-3}\text{Nm}^{-1}$ and equilibrium length $l = 3\text{nm}$. The choice of spring constant was designed to give a fluctuation time constant of $\tau = 30\text{ns}$ for testing purposes. We performed an ensemble of 1000 simulations for $100\mu s$ to fully capture the dynamics of all available types of motion for this system.

For this system, we firstly checked that each individual simulation equilibrated by measuring the average potential energy of each simulation as an average over the full $100\mu\text{s}$, corresponding to 10000 simulation frames. With an expected value $\langle U \rangle = 0.5k_BT$, the average error for each simulation was $(1.37 \pm 1.03)\%$. A single simulation can be seen equilibrating in Figure 4.4, which converged to $\langle U \rangle = 0.5013k_BT$, a 0.03% error compared to the expected value.

We also calculated the diffusion of the entire dumbbell with respect to its starting position by calculating $\langle x_c - x_{c,0} \rangle$ at each time t averaging over the entire ensemble of 1000 simulations. We obtain a diffusion constant $D = 74.08\text{nm}^2/\text{ns}$, an error of 1.93% compared to the expected value calculated from Equation 4.33. Finally, the variance in the length of the spring was calculated at each time step from the ensemble of simulations. This equilibration of variance is shown in Figure 4.6, where we see that the convergence is almost instantaneous over this time-scale, given the size of our

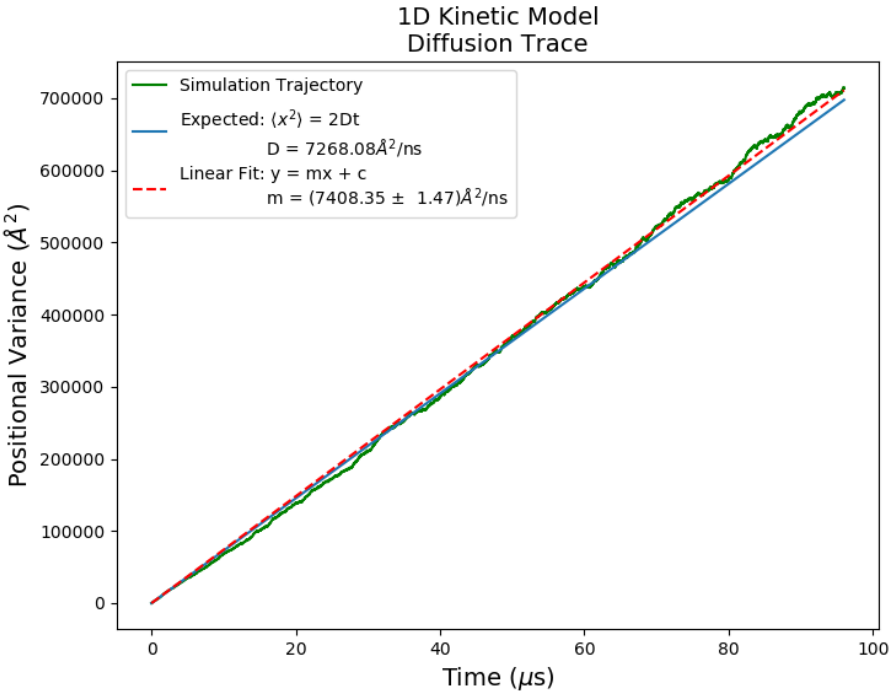


FIGURE 4.5: The diffusion trace of the 1D dumbbell model with no coupled kinetics.

simulation time step. We obtain the average extensional variance using a least-squares fit to a straight line, giving us $\langle (x' - l)^2 \rangle = 218.89 \text{Å}^2$, a 0.39% error compared with the theoretical value.

Now that we have shown that the underlying dynamical simulation functions correctly and within a reasonable error tolerance, we can define multiple kinetic states with associated transition rates and couple them to the dynamical model. We chose, somewhat arbitrarily, to define four states with the parameter sets k_α and l_α as shown in Table 4.1. These parameters are representative of the biological mesoscale [101] but not specific to any real system. In addition to the states, we must define the mesoscale

State (α)	k(pN/nm)	l(nm)
1	1	5
2	2	3
3	3	2
4	4	1

TABLE 4.1: The parameters defining each one of the kinetics states within the bead-spring system.

average rates, $R_{\alpha\beta}$, at which the states transition between one another. As we saw from Equation 4.6, the detailed balance condition means that occupation properties are related to the relative rate proportions only, and not their individual magnitudes.

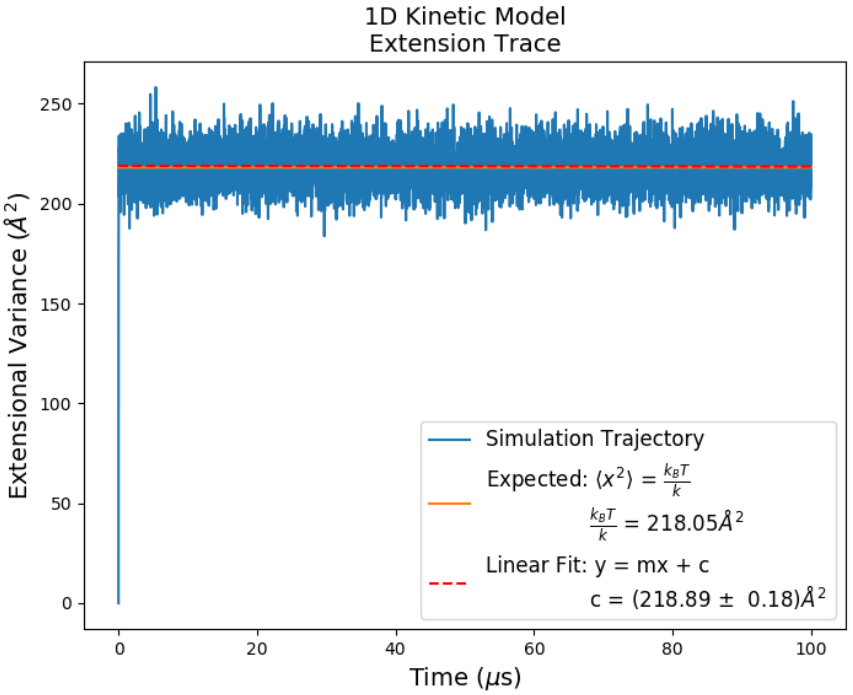


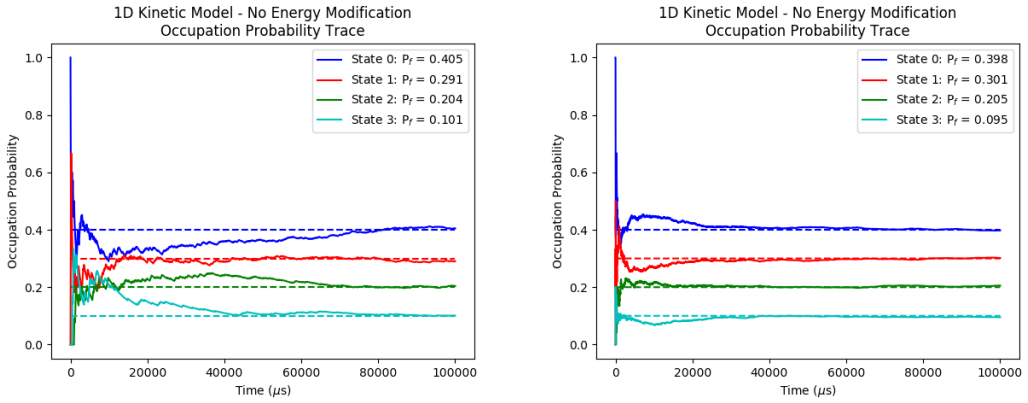
FIGURE 4.6: The variance in length between beads for the 1D dumbbell model.

We chose four occupation probabilities in advance, $0.1, 0.2, 0.3$ and 0.4, and used the detailed balance conditions between each pair of states to define a set of rates that should correspond to the pre-defined distribution of mesostate occupation. The two sets of values shown in Table 4.2 have this property, with the faster set simply being a multiple of the smaller set, thus preserving the relative rate conditions required for detailed balance whilst at the same time having a faster equilibration time-scale. These rates have been largely accelerated when compared with experimental rates (chemical reaction rates, for example) in order to show the convergence of the occupation probabilities within a reasonable amount of simulation runtime.

		Set 1					Set		
From / To	1	2	3	4	From / To	1	2	3	4
1	N/A	30	20	10	1	N/A	300	200	100
2	40	N/A	20	10	2	400	N/A	200	100
3	40	30	N/A	10	3	400	300	N/A	100
4	40	30	20	N/A	4	400	300	200	N/A

TABLE 4.2: The two sets of average transition rates $R_{\alpha\beta}$ between kinetics states within the bead-spring system, with units of MHz.

Figure 4.7 shows the convergence of occupation probabilities for the two simulations resulting from each of the sets of kinetics rates. These initial simulations ran without the mechanical energy modifications to the transition rates shown in Equation 4.42



(A) Slow set of kinetic rates (B) Fast set of kinetic rates

FIGURE 4.7: Running averages of the kinetic state occupation probabilities emerging from the two different sets of transition rates. These rates were constant throughout the simulation, i.e. unmodified by the dynamic mechanical energies.

i.e. $r(\vec{x})_{\alpha\beta} = r^0_{\alpha\beta} = R_{\alpha\beta}$. In each, we see that the same occupation probabilities are reached but at different rates, consistent with the differing magnitudes of the two sets of kinetic rates. Additionally, these converged probabilities are consistent with the analytical solution of each of the detailed balance conditions between states.

Figure 4.8 shows similar convergence properties for a simulation using the slow set of kinetic rates, but this time including the energy modification term in Equation 4.42, therefore utilising the full kinetic framework we have developed. We see that these continuously varying kinetic rates seem to have converged to the global averages faster compared to Figure 4.7a. However, this is simply an artefact of the single simulation and not a general property of including energy modifications to the kinetic rates. The important outcome is that the set of base rates $r^0_{\alpha\beta}$ were not equal to the average rates throughout the simulation, yet the exploration of phase space and associated energy modifications that form the varying rates $r(\vec{x}, \vec{p})_{\alpha\beta}$ gave us exactly the occupation probabilities we expected, validating the technique. The set of $r^0_{\alpha\beta}$ values calculated using Equation 4.42 are shown in Table 4.3.

		Set 1		
From / To	1	2	3	4
1	N/A	43.00	42.44	34.28
2	57.34	N/A	23.94	16.86
3	84.88	35.91	N/A	11.93
4	137.12	50.58	23.86	N/A

TABLE 4.3: The set of (slow) base transition rates $r^0_{\alpha\beta}$ between kinetic states within the bead-spring system, with units of MHz.

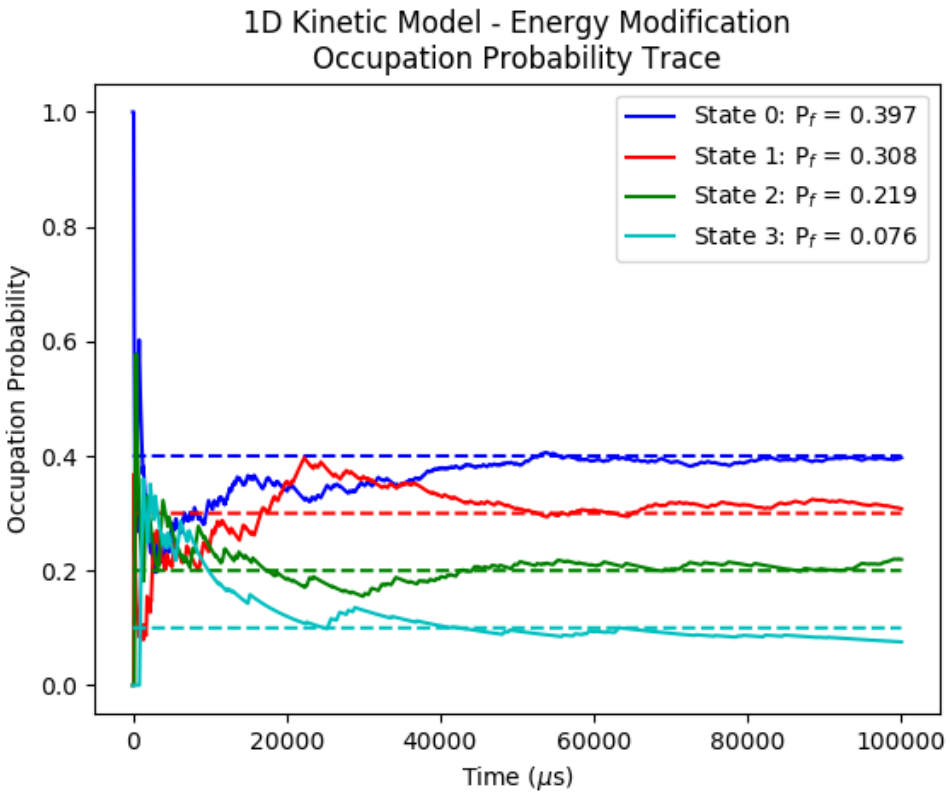


FIGURE 4.8: Running averages of the kinetic state occupation probabilities emerging from the slow set of transition rates. These rates were modified by the dynamic energy changes throughout the simulation.

4.5 Implementing Kinetics Within the FFEA Framework

In the previous section we examined the coupling between mesoscopic and microscopic kinetics in mesoscale simulations with multiple kinetic states. This model is completely general, and can be implemented within any currently existing dynamical simulation framework. We will continue the development of FFEA with our implementation of the kinetic model within the FFEA dynamical framework.

4.5.1 A Kinetic State within FFEA

Within an FFEA simulation we can define a state in much the same way as Section 4.4 for a one-dimensional system. The structural components of a simulated object within FFEA are separated into their own files and are as follows:

- The equilibrium positions of all of the nodes (Filetype: $\rightarrow$ *.node*)

- The topology (connectivity) of these nodes into a finite element mesh structure (Filetype: → *.top*)
- The material parameters of each element (Filetype: → *.mat*), which are themselves as follows:
  - Density
  - Shear + Bulk Internal Viscosity
  - Shear + Bulk Elastic Moduli
- Van der Waals interaction sites (Filetype: → *.vdw*)
- Immobile nodes (Filetype: → *.pin*)

A change in any of these components would cause the simulated object to behave in a physically different way, with different equilibrium and non-equilibrium dynamics emerging from a simulation, and so together we define them to be a kinetic state of the FFEA object. For example, if we were to change the equilibrium configuration of the nodes, this would change the structure about which the object fluctuates (under the current constitutive stress model). This is equivalent to a change in the equilibrium length of the spring in Section 4.4. Changing the connectivity of the tetrahedral mesh could change the local deformability of the mesh, due to the change in how material parameters are discretely distributed around the object. A mesh alteration could also introduce or remove degrees of freedom from the object. Changing the Van der Waals interaction sites, strengths or ranges would fundamentally change how molecules interact with one another. To visualise this idea, two different states have been defined for a hexahedral test object in Figure 4.9a. One state has a cubic equilibrium structure, the other a parallelohedron, each with different mesh connectivities and material properties. These objects represent the *same object* within the simulation with two different equilibrium structures. Just like in 1D, there are transition rates between these two kinetic states, but the method by which we perform the transition from one state to another is more complex in 3D.

4.5.2 Transitioning Between Kinetic States in FFEA

In our one-dimensional example in Section 4.4, as we transitioned into a new kinetic state we kept the positions of the nodes constant whilst only the material parameters were changed, the values of k_α and l_α. We were able to do this because each state had only one degree of freedom, the spring extension x', which implied a one-to-one

correspondence between the beads. As we can see from Figure 4.9a, the general case for a three-dimensional structure is that the different kinetic states of the same object can have different mesh connectivities and even different numbers of nodes and degrees of freedom. As the two equilibrium structures may be very different from one another, simply defining a new equilibrium configuration for the same mesh is not the optimum solution. We saw in Chapter 3 that simulations are numerically inhibited by the minimum length-scale within the system. Deforming a mesh too far away from its equilibrium structure tends to result in very long, thin elements, which are exactly the types of elements we want to avoid. A kinetic transition corresponds to this type of non-equilibrium deformation, and so the resultant structure would be very likely to contain elements that would immediately invert during an FFEA simulation.

A better solution is to have each equilibrium structure associated with its own unique mesh. Each mesh can then be independently built using meshing software such as NETGEN that has been specifically optimised to avoid numerically unstable elements [5]. However, meshing algorithms do not generally concentrate on finding a user defined number of nodes, as a mesh does not correspond to the underlying structure in any physically relevant way. Therefore, two structural meshes have no local or global node correspondence, and so the number of nodes in two different meshes representing two kinetic states of the same object can have, in general, different numbers of degrees of freedom.

To transition between states in an FFEA simulation, then, we need a method by which the kinematic properties of one equilibrium structure, the positions and velocities of all of the nodes, can be used to generate an equivalent set for the other structure. In short, we must generalise a co-ordinate mapping procedure from the co-ordinate basis of the initial structure to that of the target structure.

4.5.3 Kinematic Mapping

To transition between kinetic states, we derive a mapping procedure similar to that used by Noid *et al.* in their development of a generalised CG procedure for atomic systems [102]. Given two equilibrium structures X and Y with P and Q nodes respectively, we need to be able to construct the structural topology of Y using the details of structure X. In other words, we need to generate a Q node target FFEA structure (Y) from a P node base structure (X). We may write the positions of the target nodes, $\vec{y}$, as a function of the positions of the base nodes, $\vec{x}$,

$$y_{\alpha k} = M_{\alpha\beta} x_{\beta k}, \tag{4.47}$$

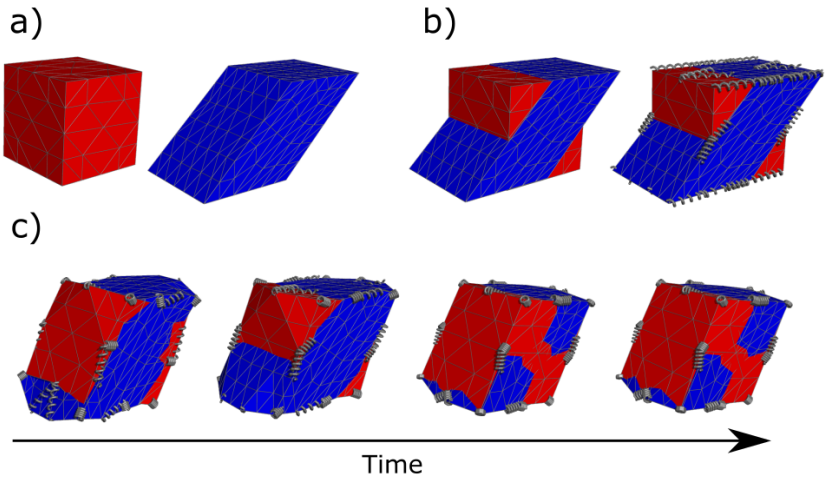


FIGURE 4.9: The process by which we generate a coordinate mapping between two kinetic states. a) The two different kinetic states we want to simulate. Both have different equilibrium structures as shown, but also differing material parameters. b) The set up for the simulation required to generate the map. We firstly overlay the structures, and then attach linear restraints between user defined 'equivalent positions'. All of the corners and the center points of each edge are attached here. c) The simulation with linear restraints included. We can see that the restraints are strong enough to overcome the elasticity of the continuum, forcing them to overlap to the global energy minimum subject to these configurational constraints.

where $M_{\alpha\beta}$ is a $Q \times P$ linear mapping operator which we must construct. Notice that the matrix operates on the nodes indices, indexed by α and β, and not to each individual directional component, indexed by k i.e. the same mapping is performed on each of the Cartesian directions. Figure 4.9a provides an example of two possible structural states, the cube and the parallelohedron.

For a stable FFEA simulation, the only constraint on $\mathbf{M}$ is that following its operation on X, structure Y must contain no inverted elements. Any form of $\mathbf{M}$ which exhibits this property will allow a kinetic state change to occur within an FFEA simulation with no disruption. However, the total set of possible mappings include center of mass translations, rotations and volumetric and shear structural deformations. The FFEA kinetic framework was designed to include the effect of how the dynamics change due to their energetic landscape, with kinetic transitions acting as instantaneous changes in the energy environment. The application of these types of structural maps constitute instantaneous changes in both energy and structure, something the kinetic framework was not specifically designed for. In the dumbbell model (Section 4.4.2), a kinetic transition involved the material parameters changing instantaneously whilst the spring extension was kept constant.

As kinetic transitions occur instantaneously within the FFEA framework as well, we must consider some form of structural alignment between the base and target structure

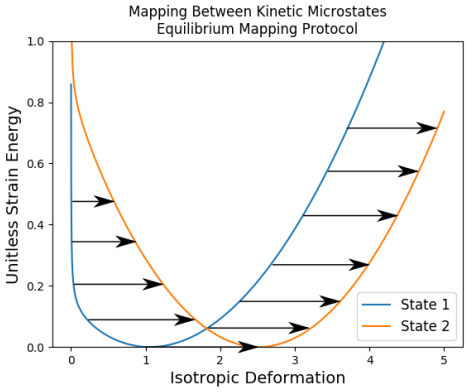
before constructing the mapping matrix.

4.5.3.1 Deformable Structure Alignment

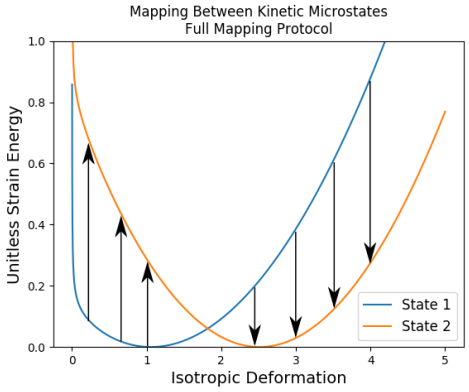
Making the mapping free from translations and rotations is relatively straightforward. A number of algorithms are available for point cloud alignment; alignment of rigid sets of nodes with no topological information. This type of alignment is shown in Figure 4.9b. One choice of mapping would be to define a structural map directly between these two equilibrium structures. This would allow us to transition directly between the initial and final equilibrium states of the process, but misses out the mechanical process of the transition itself. We call this type of structural transition 'equilibrium mapping'. However, as is clear for the hexahedral structures in Figure 4.9, FFEA objects are inherently deformable, and so we can further align the two structures by volumetric and shear deformations.

For two mechanically deformable structures, with different equilibrium structures but representing the same object, we would expect there to be a continuous mechanical transformation that would change one structure into the other composed of translations, rotations and shear and volumetric deformation. We can use the dynamic part of FFEA to take advantage of this and align the two structures using an energy minimisation procedure. The ideal case would be to define an energy penalty for the two structures being out of alignment, and run a dynamic FFEA simulation in the absence of noise and simply allow the two structures to move into alignment. However, how two structures 'should' align is a non-trivial problem. Volume overlap is an insufficient metric to define overlap as can be seen with our hexahedral test objects in Figure 4.9b, where we notice that there are a number of rotational symmetries which would have the exact same volume overlap. For more complex structures, a volume overlap algorithm may instead get caught in a local overlap minimum and converge to a completely incorrect alignment.

Instead, we consider that it is more often the case that a user will have some idea of certain points in the structure in one state corresponding to given points of the structure in the second state (e.g. the location of a binding domain might be obvious in both structures, and so the mapping we use should make these domains overlap). We have designed a semi-automatic method of alignment and mapping for the FFEA toolkit which introduces linear restraints between user-defined 'equivalent points'. This gives us an energy penalty function and enables an FFEA simulation, in the absence of steric interactions and thermal noise, to relax and overlap to a joint minimum energy configuration based on their shape. In Figure 4.9b, we have added a number of linear



(A) Example of the equilibrium mapping procedure



(B) Example of the full mapping procedure

FIGURE 4.10: The idealised case of the equilibrium and full mapping procedures for a single degree of freedom undergoing a kinetic transition from mesostate 1 to mesostate 2. For the equilibrium mapping protocol, we see that there should be no mechanical energy change affecting the mapping. In the full mapping case however, the size of the mechanical energy change depends upon the position in phase space.

restraints between the equivalent corners and sides of the cube and parallelohedron, which are visualised as springs. In Figure 4.9c, we then perform consecutive FFEA simulations with further restraints introduced between each run, giving us an adaptive procedure to generate a user-defined overlap. In principle one could continue to add springs to this to increase the degree of overlap. Alternatively, once the two states have been adequately aligned, an additional overlap energy function could be added, though this has not been implemented within the framework.

We can then define a structural map between the two deformed structures following the energy minimisation procedure. This type of map would usually introduce a significant amount of energy to the system as the state transition occurs, following which the new state would relax into its equilibrium state through the dynamic part of the simulation. This is the method we used in Section 4.4, which we will refer to as 'full mapping'. We show the effect of both types of map in Figure 4.10.

4.5.3.2 Building the Mapping Matrix

Now we have two suitably overlapping structures (either for equilibrium or full mapping), we can begin to build the mapping matrix. However, with P possible base nodes to choose from to determine the position of any target node, there is no unique structure for $M_{\alpha\beta}$. In the case that $Q > P$, Equation 4.47 represents an overdetermined system of equations with the likely scenario of no solutions for a specific matrix $\mathbf{M}$. For $Q < P$, Equation 4.47 is instead under-determined with the likely case being

infinite possible solutions. However, we can exploit the constant topology of the target FFEA structure to build $\mathbf{M}$ with a unique mapping between any two structures with the same underlying topologies. We accomplish this by choosing to make each target node position a function of only the nodes making up the base element that contains it,

$$\vec{y}_\alpha = m_{\alpha\gamma}\vec{x}_\gamma, \tag{4.48}$$

where $m_{\alpha\gamma}$ is an $Q \times 4$ sub-matrix of $M_{\alpha\beta}$ and $\vec{x}_\gamma$ are the base nodes forming the containing element. Hence, $0 < \gamma < 4$. If the two structures are such that there are some nodes with no containing element, Equation 4.48 still applies but becomes an extrapolation from the coordinates of the nearest element rather than an interpolation from the containing element. The initial alignment procedure minimises the errors introduced by extrapolation by aligning the structures based upon their shape and local deformability.

We now wish to specify the position y_α of node α that is within (or close to) an element of structure X using the coordinates of the containing element, $\{\vec{x}_0, \vec{x}_1, \vec{x}_2, \vec{x}_3\}$. As we require the mapping to be translationally invariant, we can disregard one additional degree of freedom and use the local coordinates of the element to determine a unique value of y_α with respect to a local origin, the node $\vec{x}_0$,

$$\vec{y}_\alpha = \vec{x}_0 + \sum_{\delta=1}^{3} c_{\alpha\delta}\left(\vec{x}_\delta - \vec{x}_0\right). \tag{4.49}$$

The vectors $\vec{x}_\delta - \vec{x}_0$ are the three edge vectors joined to the origin node. Three linearly independent vectors are sufficient to uniquely determine the position of any point in 3D space, and so expanding the vectors into their directional components, k, gives,

$$y_{\alpha k} - x_{0k} = \sum_{\delta=1}^{3} c_{\alpha\delta}\left(x_{\delta k} - x_{0k}\right), \tag{4.50}$$

$$= \sum_{\delta=1}^{3} c_{\alpha\delta} X_{\delta k}. \tag{4.51}$$

$X_{\delta k}$ is a 3×3 matrix, which can be quickly inverted to find the mapping pre-factors, $c_{\alpha\delta}$, for this particular target node α,

$$c_{\alpha\delta} = \left(y_{\alpha k} - x_{0k}\right) X^{-1}_{k\delta}, \tag{4.52}$$

where the inverse $X^{-1}_{k\delta}$ must exist due to the linear independence of the element edge vectors. Using the values for $c_{\alpha\delta}$, we can convert back into the global coordinates required for $m_{\alpha\gamma}$,

$$m_{\alpha\gamma} = \begin{cases} 1 - \sum_{\delta=0}^{3} c_{\alpha\delta} & \gamma = 0 \\ c_{\alpha(\gamma-1)} & 1 \leq \gamma \leq 3 \end{cases}$$

$m_{\alpha\gamma}$ contains the non-zero components of a single row of the overall mapping matrix $M_{\alpha\beta}$, and so a computational loop over all target nodes will fully populate the global mapping matrix and give us an interpolative, and therefore stable, mapping matrix.

As the global matrix is not square, we cannot simply calculate the inverse to find the equivalent mapping from structure Y to structure X. Therefore, we begin from the aligned state of the two structures and perform the same procedure a second time to build the reverse map. The non-square nature of the mapping matrix presents certain problems in regards to energetics and equipartition which we will discuss in the following section. As a final point, if we take the derivative of Equation 4.47 with respect to time, where $\mathbf{M}$ is constant, we find that the mapping matrix is suitable for the velocity components as well.

4.5.4 Kinetic Transitions and Energy in FFEA

When we attempt a transition between kinetic states using rates similar to those in Equation 4.20, it is the change in energy that affects the transition rates. In FFEA, the total amount of energy that the object will typically attain through thermal fluctuations is dependent upon the number of degrees of freedom used to define the structure. Therefore, a direct comparison of energies following a kinetic state change could easily be affected by the difference in the number of nodes between the base and target structures. This is clearly physically unrealistic, as the number of nodes used to define a structure in FFEA is completely arbitrary.

Consider the two 1D structures shown in Figure 4.11, where structure X has one more node than structure Y and hence an additional degree of freedom. Let us assume that these two structures are equivalent objects and that they have already undergone the alignment procedure for equilibrium mapping described in Section 4.5.3.1. We define the two mapping matrices between these structures as $\mathbf{M}_{YX}$, the map from structure X to structure Y, and $\mathbf{M}_{XY}$, the map from structure Y to structure X. Simply by

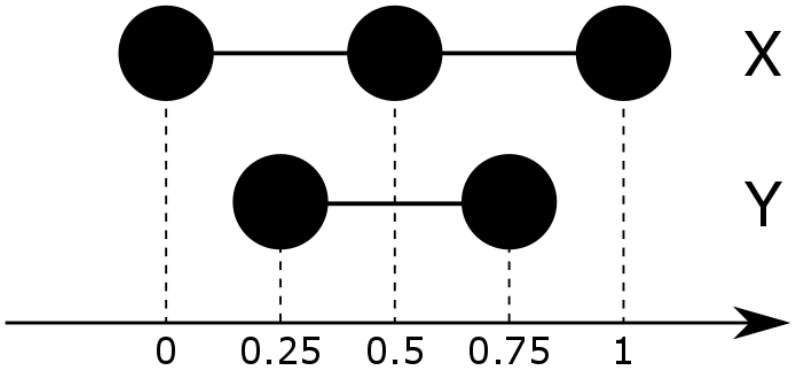


FIGURE 4.11: Two simple 1D finite element structure to which he kinetic mapping procedure can be applied.

inspection, it is relatively straightforward to show that the two maps are,

$$\mathbf{M}_{YX} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$
$$\mathbf{M}_{XY} = \frac{1}{2}\begin{pmatrix} 3 & -1 \\ 1 & 1 \\ -1 & 3 \end{pmatrix} \tag{4.53}$$

where $\mathbf{M}_{YX}$ would be applied to a three component position vector $\vec{x}$ and $\mathbf{M}_{XY}$ to a two component position vector $\vec{y}$.

If we apply a map to its corresponding structure, and then the reverse map to the result, we may expect to obtain the original structural state exactly. In other words, the product of the two mapping matrices may intuitively be expected to be the identity matrix. If we begin with structure Y, the structure with fewer degrees of freedom, that is indeed the case,

$$\mathbf{M}_{YX}\mathbf{M}_{XY} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4.54}$$

However, if we begin with structure X,

$$\mathbf{M}_{XY}\mathbf{M}_{YX} = \frac{1}{4}\begin{pmatrix} 3 & 2 & -1 \\ 1 & 2 & 1 \\ -1 & 2 & 3 \end{pmatrix} \tag{4.55}$$

The two maps operating in the order in Equation 4.55 clearly do not form the identity matrix as we might expect. Eigen-decomposition shows that instead, the matrix has two eigenvalues equal to unity, with associated eigenvectors $\vec{v}_1 = (-1, 0, 1)$ and $\vec{v}_2 =$

$(0, 1, 2)$, and a third eigenvalue equal to zero with associated eigenvector $\vec{v}_3 = (1, -1, 1)$. If we imagine the line segments between nodes as 1D finite elements, then FFEA does not permit position vectors with the topology of $\vec{v}_3$ to exist, as it constitutes element inversion. Only the position vectors that are multiples of $\vec{v}_1$ or $\vec{v}_2$ return to their original structure under the sequential action of the two maps. Any other vector, say, $\vec{x} = (x_1, x_2, x_3)$, becomes,

$$\begin{aligned}\vec{x}' = (\mathbf{M}_{XY}\mathbf{M}_{YX})\,\vec{x} &= \frac{1}{4}\begin{pmatrix} 3 & 2 & -1 \\ 1 & 2 & 1 \\ -1 & 2 & 3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \\ &= \frac{1}{4}\begin{pmatrix} 3x_1 + 2x_2 - x_3 \\ x_1 + 2x_2 + x_3 \\ -x_1 + 2x_2 + 3x_3 \end{pmatrix}\end{aligned} \tag{4.56}$$

From here, we note that for every initial vector $\vec{x}$, we find $x_2' = 0.5(x_1' + x_3')$, which shows that the central node will always be placed exactly at the center of the two outer nodes, and $x_3' - x_1' = x_3 - x_1$, or, the separation between the outer nodes is conserved over the mapping procedure and is independent of the position of the central node. This shows that when our end result requires the inclusion of extra degrees of freedom, our map acts to place those degrees of freedom into the relative positions that they were in when the maps were first defined. For both the equilibrium and full mapping procedures, because we used an energy minimisation method to construct the mapping operators to begin with, the relative positions of the extra degrees of freedom therefore corresponds to their energy minimised state, subject only to the enforced structural overlap.

Within a kinetic FFEA simulation, when we compare the energies of the initial and target structures to determine the energy modifications to the kinetics rates, we implement the following protocol,

- Apply $\mathbf{M}_{YX}$ to the base structure, X, to produce a target structure, Y
- Apply $\mathbf{M}_{XY}$ to the structure Y to get a base structure X' different from X
- Apply $\mathbf{M}_{YX}$ to the structure X' to get a target structure Y' different from Y
- Calculate the energy difference between X' and Y' and use this value to modify the kinetic rates
- If a kinetic transition occurs, use the structure Y' as the new target structure, and not Y

The above procedure is designed to remove all energy fluctuations associated with the "extra" eigen-modes of the structure with the greater number of degrees of freedom, so that the energy of the two structures can be compared on an even basis. This method should reduce the amount of energy lost or gained following a kinetic transition in the simulation due to the addition of arbitrary degrees of freedom and their associated thermal fluctuations to the varying mesh structures.

4.6 A Three-Dimensional Example - Cubes and Parallelohedra

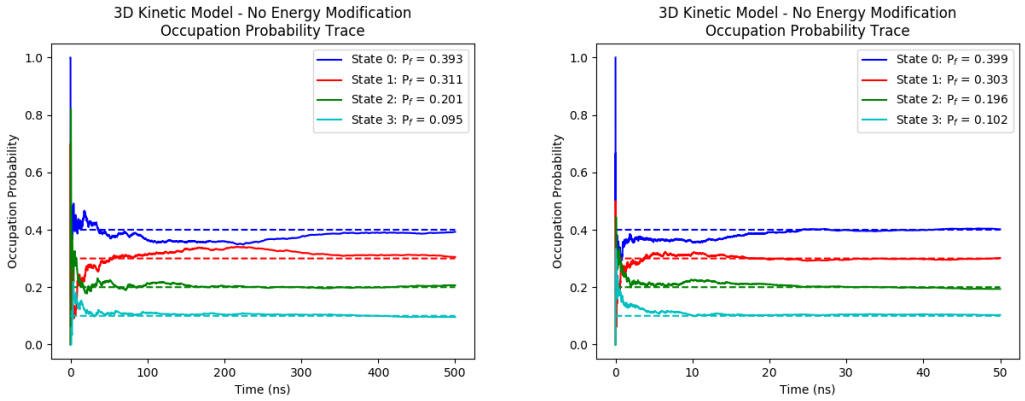
We performed a full coupled kinetic simulation on the two hexahedral example structures from Figure 4.9. As with the dumbbell model, we again chose to have four kinetic states, but this time the states have been defined much more rigorously within the FFEA framework, and are shown in Table 4.4. The cubic (equilibrium) structure has 21 nodes, 27 tetrahedral elements and 144 surface faces, and the parallelohedron has 63 nodes, 160 elements and 416 faces. Each element from both systems was parametrised with the same material parameters, corresponding to a Young's modulus, $E = 1\text{GPa}$ and a Poisson ratio, $\nu = 0.35$, and bulk and shear viscosities $\mu_S = \mu_B = 1 \times 10^{-3}\text{Pa.S}$. The base length of each object, $L = 5nm$.

Mesostate	1	2	3	4
Conformation	Cube	Parallelohedron	Parallelohedron	Cube
Occupation Probability	0.4	0.3	0.2	0.1

TABLE 4.4: The states in our example FFEA simulation defined by their conformational structures.

Conformation	Cube	Parallelohedron
Nodes	21	63
Elements	27	160
Surface Faces	144	416
Density	N/A	N/A
Young's Modulus	1GPa	1GPa
Poisson Ratio	0.35	0.35
VdW Interactions	N/A	N/A
Pinned Nodes	N/A	N/A

TABLE 4.5: The set of conformations within our example FFEA simulation defined by the relevant structural components. As our simulations are non-inertial, the density is not defined. VdW interactions and pinned nodes are also optional, and so left out for simplicity.



(A) Slow set of kinetic rates (B) Fast set of kinetic rates

FIGURE 4.12: Running averages of the kinetic state occupation probabilities emerging from the two different sets of transition rates for the FFEA cube/parallelohedron kinetic model. These rates were constant throughout the simulation, i.e. unmodified by the dynamic mechanical energies.

Previous applications of the FFEA software together with a newly developed, rigorous test suite [93] show that the underlying mechanical model for FFEA is consistent with theoretical predictions, so we need only test the new kinetic components.

4.6.1 3D Example - No Energy Modifications

Firstly we ran kinetic simulations in the absence of energy considerations, using the average transition rates shown in Table 4.6 and treating them as constant. These values are similar to Table 4.1 and thus give the same resulting occupation probabilities, but they are much faster to allow for probability convergence within a single simulation. As energy modifications to the rates are not being made, which mapping protocol we use makes no difference in regards to convergence tests. Regardless, the structural maps used in these simulations were constructed using the full mapping procedure. All simulations were run for 500ns.

		Set 1					Set 2		
From / To	1	2	3	4	From / To	1	2	3	4
1	N/A	3	2	1	1	N/A	30	20	10
2	4	N/A	2	1	2	40	N/A	20	10
3	4	3	N/A	1	3	40	30	N/A	10
4	4	3	2	N/A	4	40	30	20	N/A

TABLE 4.6: The two sets of average transition rates $R_{\alpha\beta}$ between kinetics states within the cube/parallelohedron system, with units of GHz.

Figure 4.12 shows the results of these simulations. We have shown only the first 50ns of the faster set of rates for clearer comparison. We can clearly see that the faster set

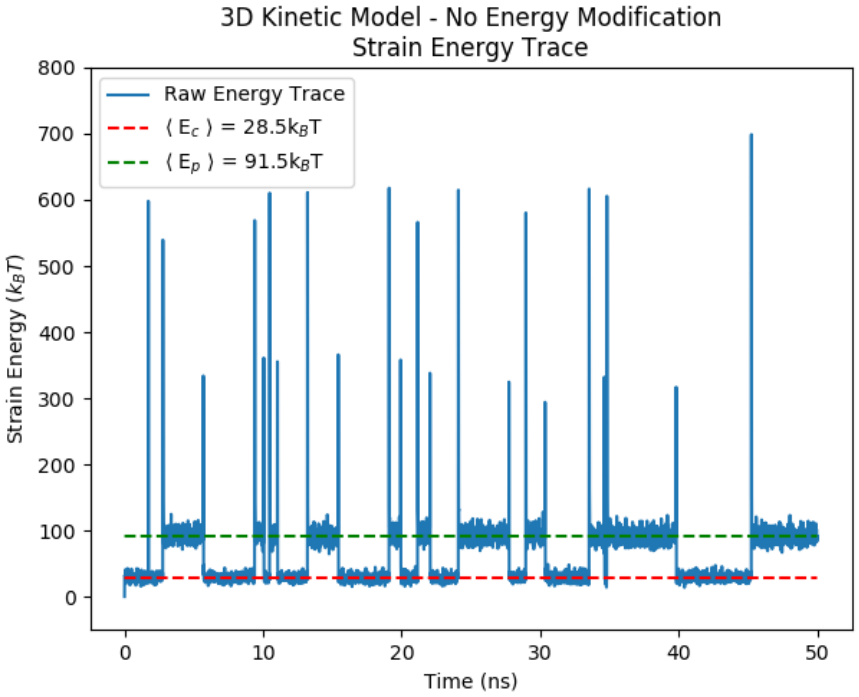


FIGURE 4.13: Raw energy trace from the first 50ns of the cube/parallelohedron FFEA kinetic simulation. We clearly see two distinct equilibrium states corresponding to the equipartition of energy for the cube and parallelohedron.

of rates converge to the predicted values approximately 10 times faster than the slow set.

Although the simulations utilising the rates from Table 4.6 sufficiently show occupation probability convergence, both sets of rates are too fast to allow the systems to relax into equipartition of energy (see Appendix B). We therefore ran one additional simulation using the slower set of rates reduced by a further factor of 10. The raw energy trace for the first 50ns of this simulation is shown in Figure 4.13, which we can see moving between the expected values for the cube and parallelohedron as per the equipartition theorem, with large spikes in energy between each transition. These spikes correspond to the energy changes $\Delta E(\vec{x})^{nc}_{\alpha\beta}$ that determine how much each transition rate between states is reduced / increased.

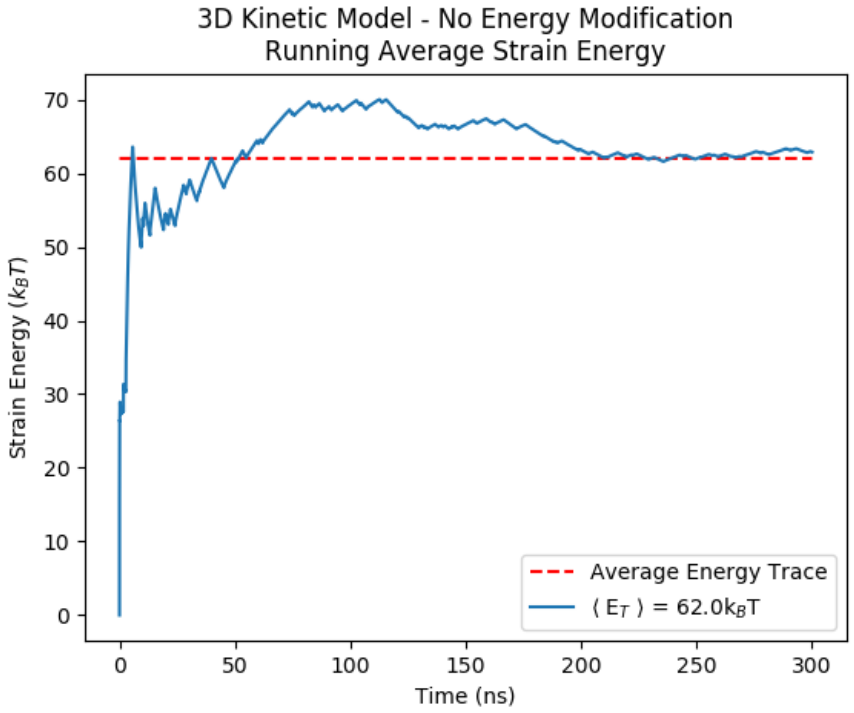


FIGURE 4.14: Running average of the strain energy from a 300ns of the cube/parallelohedron FFEA kinetic simulation.

We can use the expected occupation probabilities to calculate the expectation value of the total strain energy, E_T, within the system,

$$\begin{aligned}\langle E\rangle &= \sum_{\alpha=1}^{4} P_\alpha \langle E_\alpha\rangle, \\ &= (0.1+0.4)\langle E_c\rangle + (0.2+0.3)\langle E_p\rangle, \\ &= \left(\frac{3}{4}\left(N_c+N_p\right)-1\right)k_BT, \end{aligned} \tag{4.57}$$

where N_c and N_p are the number of nodes within the cube and parallelohedron structures respectively. A running average of energy for a 300ns simulation is shown in Figure 4.14. The underlying, continuous microstate probability densities seem to be disproportionately skewed towards the higher energy states of their respective Boltzmann distributions to begin with. This is perhaps due to the huge amounts of energy gained by the system as it transitions to a new state, or could simply be an artefact from averaging over a single simulation. In either case, total system equilibration is attained as the length of simulation increase, showing that each kinetic transition was slow enough to allow each state to relax and undergo sufficient thermal exploration to approximate thermodynamic equilibrium.

We note that in these simulations, detailed balance is only adhered to at the mesostate level, and not at the microstate level, and as such the amount of time spent in each of the microstates will be incorrect. However, even without inclusion of energy modifications we are still able to introduce large amounts of energy into the system and study the effect on the resulting dynamics.

4.6.2 3D Example - Energy Modifications

Finally, we consider simulations in which the energy differences were used to modify the transition rates. Unlike in Section 4.4, we no longer have a single degree of freedom. We are now working with biologically realistic objects with hundreds of degrees of freedom in both the initial and final mesostates of a kinetic transition. Each structure has the potential to store considerably larger amounts of energy than a 1D model. As such, we need to understand exactly what our mapping physically represents as we consider each form of energy comparison.

Remember that our cubic structure had 21 nodes, the parallelohedron 63 nodes and both had side length $L = 5\text{nm}$. With these parameters, a transition from the equilibrium state of the cubic structure to the parallelohedron using the full mapping procedure (corresponding to Figure 4.10b) constitutes a mechanical energy change of $2922k_BT$. If a kinetic process with rate $r_{\alpha\beta}$ was gated by this mechanical energy change as per Equation 4.20, then $r_{\alpha\beta} \approx 0$ and the process would simply never occur. However, we must consider what we are trying to represent by this process.

In a biological system, a conformational change like this cannot be achieved via thermal diffusion alone. Instead, ATP hydrolysis provides energy to the system. This process has a very low activation free energy and is therefore largely spontaneous, dependent only upon the presence of water in the local environment [103]. As such, there is no mechanical energy barrier to the independent process of ATP hydrolysis itself. However, for the conformational changes we are modelling, the question is whether ATP hydrolysis provides enough energy to the system to proceed with a conformational change that does have a mechanical energy barrier.

An ATP hydrolysis reaction [104] has an associated Gibbs free energy $\Delta G_{ATP} \approx 12k_BT$, considerably less than the mechanical energy required to undertake our example kinetic process. We must again consider the physical meaning of our process. Assuming a density of approximately $1 \times 10^3 \text{kg/m}^3$, an object of length $L = 5\text{nm}$ would contain approximately $\sim 10^4$ heavy atoms, each associated with approximately $1k_BT$ of energy, and our kinetic transition requires them all to be in a highly unfavourable

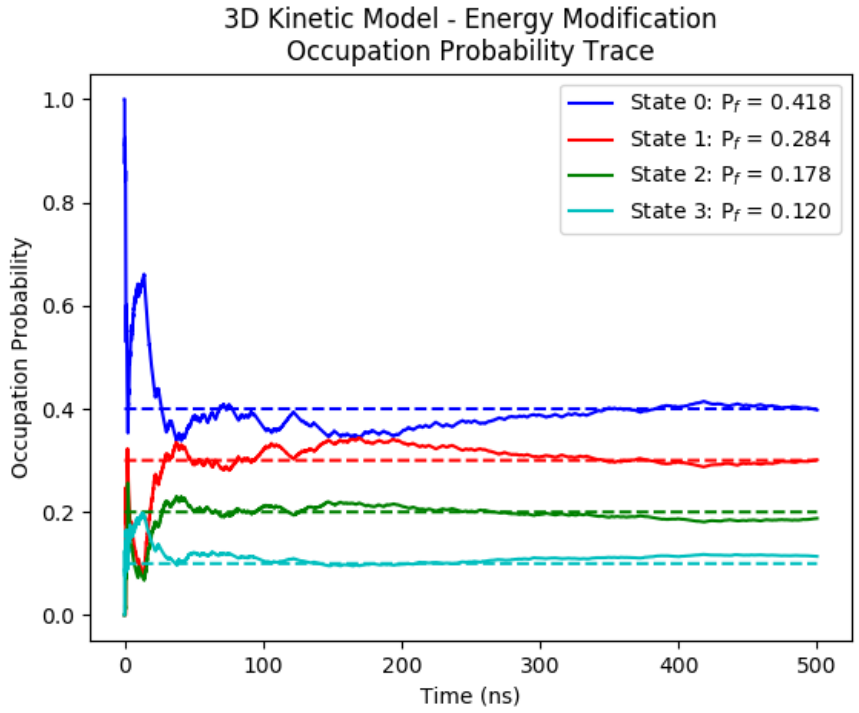


FIGURE 4.15: Running averages of the kinetic state occupation probabilities emerging from an energy modified set of transition rates for the FFEA cube/parallelohedron kinetic model.

mechanical state. Indeed, the total strain energy within a continuum mechanical object is proportional to the total volume of the object for exactly this reason. ΔG_{ATP} implies that ATP hydrolysis releases enough energy to its local environment to significantly affect only a few degrees of freedom, 5 at most, but significantly less than our example transition requires.

To compensate for this, we have reduced the size of our cube to $L = 0.6$nm. At this size, the lower end of the mesoscale, the energy changes involved in a state transition $\Delta E(\vec{x}, \vec{p})_{\alpha\beta} \sim \Delta G_{ATP}$. In these simulations, we were forced to neglect thermal effects as the size of the thermal fluctuations caused the mesh elements to invert. The system is still able to sample a reasonable amount of phase space by being brought out of equilibrium by the mapping procedure itself. As the dynamic part of the simulation is now deterministic, we can calculate the maximum energy gain from a state transition. This occurs when a parallelohedron (in its equilibrium state) transitions to a cube with $\Delta E(\vec{x}, \vec{p})_{\alpha\beta} = 7.53 k_B T$. We assume the value $f_{\alpha\beta} = 0.5$ for both the forward and backwards transitions. As shown in Appendix A, $f_{\alpha\beta}$ is related to the mechanical energy barrier, and so we are effectively assuming that the mechanical energy barrier is halfway between the initial and final mechanical energy states, and the ATP hydrolysis process provides the energy required for the remainder of the transition.

		Set 1		
From / To	1	2	3	4
1	N/A	30.16	20.12	10.00
2	40.00	N/A	20.00	9.93
3	40.00	30.00	N/A	9.94
4	40.00	30.00	20.00	N/A

TABLE 4.7: The base transition rates $r^0_{\alpha\beta}$ between kinetics states within the energy modified cube-parallelohedron system, with units of GHz.

By using a set of rates which allowed the system to relax to mechanical equilibrium between each transition, we were able to model the continuous probability density function as having only two relevant configurations, $\vec{x}_1$ and $\vec{x}_2$ with regards to kinetic transition probabilities, where $\vec{x}_1$ is the equilibrium configuration corresponding to state α and $\vec{x}_2$ is the configuration in state α corresponding to the equilibrium configuration in state β. The integrals from Equation 4.23 can then be written as summations,

$$\int_\Omega p_\alpha(\vec{x},\vec{p}) r_{\alpha\beta}(\vec{x},\vec{p}) d\Omega \approx p_\alpha(\vec{x}_1) r_{\alpha\beta}(\vec{x}_1) + p_\alpha(\vec{x}_2) r_{\alpha\beta}(\vec{x}_2),$$

$$\int_\Omega p_\alpha(\vec{x},\vec{p}) d\Omega \approx p_\alpha(\vec{x}_1) + p_\alpha(\vec{x}_2).$$

These two conditions allowed us to solve exactly for the base rate ratio,

$$\frac{r^0_{\alpha\beta}}{r^0_{\beta\alpha}} = \frac{1+\exp\left(\frac{\Delta E^{nc}_{\alpha\beta}(\vec{x}_2)}{k_B T}\right)}{1+\exp\left(\frac{\Delta E^{nc}_{\beta\alpha}(\vec{x}_1)}{k_B T}\right)} \frac{P_\beta}{P_\alpha} \tag{4.58}$$

where P_α and P_β are the mesoscale probabilities. If state α represents the cubic structure and state β the parallelohedron, then within the FFEA simulation we measured $\Delta E^{nc}_{\alpha\beta}(\vec{x}_1) = -5.04 k_B T$ and $\Delta E^{nc}_{\beta\alpha}(\vec{x}_1) = -7.53 k_B T$. Using the values for the total occupation probabilities we have been using for the rest of this section, we obtained the set of base rates for the simulation shown in Table 4.7,

For this set of base rates, which deviate only slightly from those used in Section 4.6.1, we obtained a system which converged to the required mesostate probabilities after $\sim$ 400ns, shown in Figure 4.15. Although not thermally exploring their respective Boltzmann probability distributions, the energy modifications were calculated using the respective Boltzmann probabilities following a state change to give the correct detailed balance conditions between the two considered microstates as well as the overall mesostates.

Chapter 5

Applications of FFEA to Cytoplasmic Dynein

We now have a software package capable of simulating a wide range of possible biological mechanics. Both stochastic dynamics (using standard FFEA simulations) and kinetics (through the newly developed kinetic coupling) can be performed in conjunction with one another, with an energetic coupling between the two. These two components encompass the entirety of the functional capabilities of the molecular motor cytoplasmic dynein as they are currently known.

This chapter will focus on application of the full FFEA framework to furthering our understanding of cytoplasmic dynein. We perform a manual parametrisation of the molecule using data from both high resolution MD simulations and state of the art experimental imaging, followed by a suite of simulations to look at how the monomer may interact with its microtubule track. Finally, we perform a kinetic simulation of the entire superstructure to see the effect of ATP hydrolysis on the independent dynamics of the molecule, and its interactions with the microtubule.

5.1 Introduction to Molecular Motors

Of the three different superfamilies of translational molecular motors, myosin, kinesin, and dynein, it is dynein that is generally considered to be the least well understood. The size and complexity of the molecule, as well as its diversity of function, has made the mechanism by which dynein generates force significantly harder to grasp than its counterparts. Over the past $5 - 10$ years, experimental advances have given us a

wealth of structural information with which to begin characterising this mechanism, and understand how dynein is able to perform its function within a eukaryotic cell.

We begin with an introduction to the more well understood translational motors, myosin and kinesin.

5.1.1 Myosin

Myosin is the molecular motor responsible for force generation through interactions with actin filaments [105]. The most recent phylogenetic classification of myosin has hinted at 31 distinct myosin molecules [106] which can be subdivided into 12 classes [107]. They perform a wide range of functions, from conventional muscle contraction (myosin II) to transport of vesicles and other cargo types, including entire organelles (myosin V) [108]. The differences between these types are highly nuanced from a mesoscopic perspective, yet slight differences in sequence generate the differences in activity regulation and dimerisation capabilities that specify each class of myosin to their own specific tasks. Each monomer of myosin V, for example, contains repeated hydrophobic registries that allows the formation of a stable coiled-coiled dimerisation when in water [109]. However, the mechanism of force generation is similar across all classes, whether that force is used to transport the motor itself together with its cargo, or act on an external structure (a muscle fibre, for example). A simplified structure of myosin V is shown in Figure 5.1**a)**. Each myosin class has three major domains, the motor, linker and tail. The motor domain acts as both the catalytic site of the molecule and the external actin binding region. The tail domain is a coiled-coil structure at the end of which is the cargo binding domain that mediates interactions with external objects, leaving the linker domain to couple the two subdomains together. Each myosin motor domain is able to bind an adenosine triphosphate (ATP) molecule, which can be hydrolysed to form adenosine diphosphate (ADP),

$$\mathrm{ATP} + \mathrm{H_2O} \longrightarrow \mathrm{ADP} + \mathrm{Pi}. \tag{5.1}$$

Equation 5.1, ATP hydrolysis, releases energy from the phosphate bond to the surrounding molecule [110] which in the case of myosin is used to generate force.

The functional ATP cycle for myosin proceeds as follows. When a myosin motor is not bound to the actin filament, it is able to bind an ATP molecule from the surrounding cytoplasm. This ATP is rapidly hydrolysed via Equation 5.1, and the energy released triggers a cascading series of microscopic conformational changes that result in a mesoscopic 'priming' of the neck domain, storing the released energy in a similar manner

to an elastic spring. This structural remodelling also gives the motor domain high affinity for actin. Subsequent binding of the head domain to an actin filament triggers the release of the now independent single phosphate, which in turn triggers the release of the stored energy in a process called the 'powerstroke'. Unlike during the priming process, both tail and head domains are now attached to external structures, which means that tension within myosin must apply force to these external objects. Following the powerstroke, the myosin is able to release the remaining ADP molecule, again weakening the affinity to the actin and leading to dissociation of the motor domain. The whole process can then repeat [111].

The type of myosin involved changes how the force is applied to external objects. In muscle, multiple myosin II molecules associate to the same cargo, a long, static central filament between two separate actin filaments that are arranged axially along muscle fibres. Therefore, the powerstoke causes the actin tracks themselves to move towards each other, directly causing muscle contraction [112]. If instead it is the actin filament that is rigid, as in the case of myosin V on the cytoskeletal actin network, then the powerstroke acts to pull the tail domain and associated cargo forward [113]. However, the dimerisation of two monomers in myosin V generates allosteric communication so that following the powerstroke, the leading motor is severly inhibited from releasing from the actin track [114]. Instead, the rear head is much more likely to release, which inhibits backward steps and likely forms a more efficient walking mechanism than if both motors were equally likely to release.

Although this is the core ATP hydrolysis cycle for myosin, each stage within it has specific regulation mediated by additional proteins that bind to the motor. A thorough review of the mechanism of force generation for myosins is available [115], as well as an in depth review of each individual subclass [108]. For the purposes of this work, it is sufficient to note that the same molecule is able to both translate itself together with external cargo along its filament track, or remain static and translate the track.

5.1.2 Kinesin

The 14 classes of kinesin [116] are mainly transport motors but are involved in a similarly diverse range of functions as myosin. These range from standard cargo transport around the cytoskeleton to the more specific intraflagellar transport (IFT) of the structural components of cilia as they are being built [117]. Kinesin is even involved in the construction of the mitotic spindle, a vital component of cell division [118].

Unlike myosin, kinesin associates to microtubules instead of actin filaments, specifically performing anterograde transport toward the plus-end end of the microtubule protofilaments, which generally results in motion towards the outside of the cell [119].

Disregarding the size differences and various regulatory proteins which bind to specific types of kinesin, this motor has approximately the same architecture as dimerised classes of myosin (see Figure 5.1). Kinesin dimers contain two motor domains that are again the catalytic sites of the molecule and which also associate with the microtubule track, and a stalk domain which is a long coiled-coil region terminating in a subdomain responsible for binding to the cargo. Connecting the two motors to the stalk is a smaller neck region that transduces the chemical energy generated by Equation 5.1 into force and motion.

The diversity of function between classes of kinesin result from changes in structure at the atomic level, but as with myosin, the core force-generation process is approximately conserved across the entire superfamily. However, the functional mechanism is slightly different than for myosin, possibly due to it existing as a dimer in most of its classes. Once a motor domain is bound to the microtubule track, it binds an ATP molecule. This binding triggers the powerstroke of the molecule, which causes the neck region to contract towards the bound motor. This contraction not only moves the cargo, but generates enough momentum to throw the additional, unbound motor domain forward, past the bound motor domain and toward the next binding site along the microtubule. ATP is then hydrolysed in the bound motor domain, and the resulting energy releases the contracted neck region, storing the energy for the next cycle. ADP from the previous cycle is then released from the unbound head, enabling it to bind to the microtubule. Finally, the remaining phosphate present in the original bound head is then released, which subsequently releases the head from the microtubule. The process can then be mirrored in the other motor which is now bound to the microtubule in the leading position [120].

We see here that the two heads of kinesin are highly coupled in its dimeric form, such that the powerstroke in one head directly affects the other, biasing the next step to be a step in the forward direction. Yet monomeric kinesin does exist (kinesin III), and can be processive in a so-called ‘inch-worm’ manner.

Kinesin remains loosely associated to the microtubule through non-specific electrostatic interactions with the microtubule filaments [121], which allows sliding of the motor domain along the mirotubule track without full dissociation. With no partner monomer to tether it to the microtubule, kinesin III is dependent upon this electrostatic interaction. However, the processivity of kinesin III has been found to be a factor three less

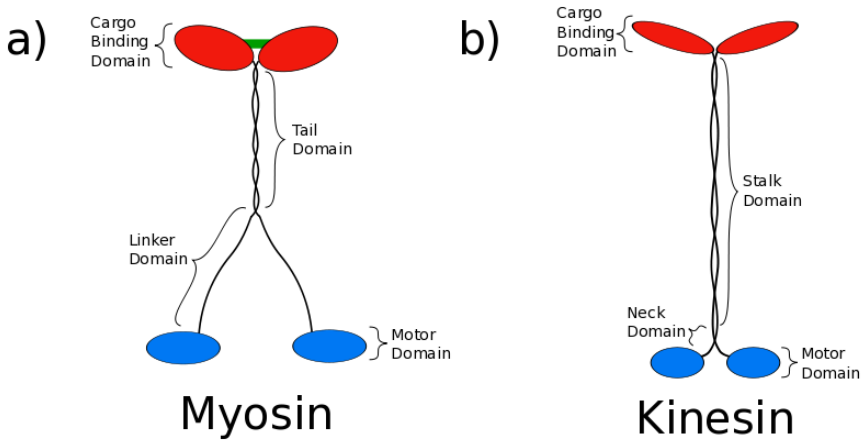


FIGURE 5.1: Minimalist depictions of the structure of two translational molecular motors. **a)** Myosin V, which exists as a dimer, and **b)** Kinesin I. *Images reproduced with reference to work by R.D. Vale* [6]

than when dimerised [122]. The attraction to the microtubule for kinesin monomers allows 1D diffusional searching for the next binding site, whereas the close coupling of the dimer motors means that the powerstroke directly places the unbound head much closer to the next binding site. This contrasts with myosin, which exists and functions both as a monomer and as a dimer. The linker region of myosin is much longer than that of kinesin, so that the powerstroke provides less of a reduction in the diffusional search time for myosin dimers. These differences in processivity mechanisms will be important to keep in mind in Section 5.2 when we look at cytoplasmic dynein.

5.2 Dynein

We now move onto the main focus of this chapter, the molecular motor dynein. As with myosin, dynein exists in both monomeric and dimeric forms. As a monomer, dynein is found within eukaryotic axonemes. Vast arrays of dynein monomers, or axonemal dyneins, apply coordinated forces to the axonemal superstructure, which results in the beating of the flagellum within eukaryotic sperm tails. As a dimer, dynein is the counterpart to kinesin, performing retrograde transport of cellular cargo to the minus-end of the microtubule network i.e. towards the cell nucleus. This form of dynein has been termed cytoplasmic dynein [7]. One may expect, then, that dynein has structural characteristics common to both myosin and kinesin, due to its dual monomeric/dimeric nature (myosin) and its binding affinity for the microtubule network (kinesin). Yet we can see in Figure 5.2 that structurally, dynein seems to have very little in common with either motor.

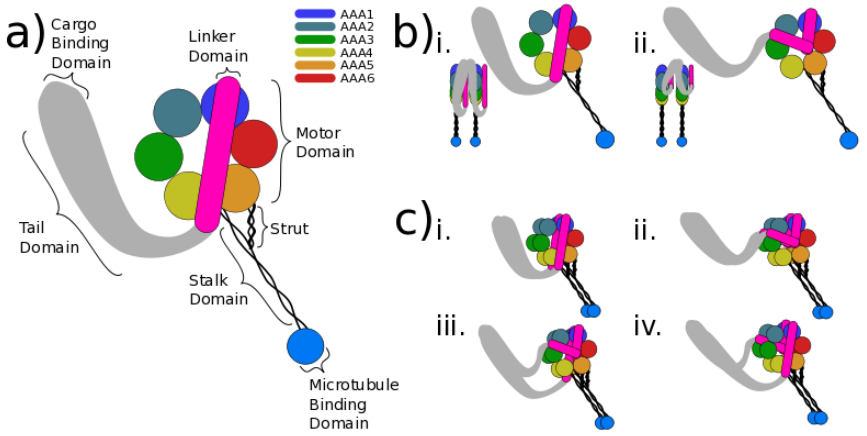


FIGURE 5.2: Minimalist depictions of the structure of the dynein molecular motor. **a)** The monomeric structure of dynein. **b)** The two main conformations of dynein throughout a kinetic cycle. **i.** The post-powerstroke state of dynein. **ii.** The pre-powerstroke state. Main images are walking to the right, and inset images walking into the page, showing the dimerisation of the tail domains. **c)** The four possible dimeric structures occuring throughout the kinetic cycle of the entire dimer. *Images reproduced with reference to work by A.J. Roberts et al.* [7]

The first thing to note is that in a dynein monomer, the motor domain is completely separate from the stalkhead that contains the microtubule binding domain (MTBD) (Figure 5.2**a)**. This motor domain is known as a AAA+ ring, a set of 6 subunits from a superfamily of ATP hydrolysing proteins which are not specific to dynein [123]. We now know that in kinesin and myosin, binding and hydrolysis of ATP at the motor domain triggers release from the microtubule, so this process must be significantly different in dynein. We also see that dimerisation of dynein occurs in the tail region [124] (Figure 5.2**b)**), leaving the two motor domains much more loosely connected to one another than in kinesin and myosin, and the MTBDs even more so due to their distance from the point of connection.

Work over the past 5-10 years has elucidated the mechanism by which a single monomer of dynein proceeds through its ATP hydrolysis cycle to generate force / motion [124]. Without a bound ATP in its primary binding pocket, dynein is able to bind strongly to the microtubule. As a dimer, both monomers will associate to approximately adjacent protofilaments. Binding of ATP to the motor domain at the AAA1 site triggers a restructuring of the entire AAA+ ring, shifting the relative positions of the AAA4 and AAA5 subunits [125]. We can see from Figure 5.2**a)** that this shift will affect both the stalk and the strut, which results in each individual coil from the stalk sliding with respect to one another. This transmits a mechanical signal down the stalk, causing the MTBD to adopt a new conformation that is only weakly attracted to the microtubule [126]. This rearrangement of the ring structure also causes the linker domain to

come into steric contact with AAA4, thus allowing it freedom of movement with respect to the motor domain. ATP binding and microtubule release also causes the linker to adopt the pre-powerstroke configuration by undergoing the 'reverse powerstroke', an approximately 90° rotation about AAA1 (clockwise in Figure 5.2**b)**). ATP hydrolysis provides energy to the molecule as it diffuses to the new binding site. Subsequent phosphate release allows the MTBD to readopt the high-affinity conformation for the microtubule, and rebinding sends a return signal up the stalk to the motor which in turn triggers the powerstroke. The linker returns to its straight position, pulling the external cargo / axoneme forward with it. ADP is then released and the cycle can begin again [127].

The full conformational cycle for a single monomer as it is currently known is shown in Figure 5.3. With the exception of the communication between the separated MTBD and motor domains, the kinetic cycle is almost identical to that of myosin. As with muscle myosin, this cycle fully explains how the dynein motor generates force in the monomeric (axonemal) form. The tail structure acts as a permanent tether to the axoneme, the motor simply detaches from the microtubule, binds again at a point further along, and undergoes the powerstroke to move to the adjacent microtubule. For cytoplasmic dynein however, the walking mechanism is very unlike dimeric myosin or kinesin. Although each step proceeds in a similar manner (with the addition of motor / MTBD allostery), in cytoplasmic dynein the two MTBDs are connected only very loosely, whereas in kinesin they are very tightly bound via the relatively short neck region. In kinesin, this tight binding combined with the 'hand-over-hand' stepping mechanism means that the binding of one motor allosterically inhibits the other motor, preventing both motors being unbound at the same time [128]. In dynein however, there is no clear mechanism for such steric inhibition due to the loose coupling between the heads. Dynein has regulatory proteins, such as Lis1 which is known to inhibit conformational changes in the molecule [129] [130]. However, our knowledge of kinesin regulation suggests that inhibition should depend not on an external molecule, but on the state of the other monomer. In the case of kinesin, this steric inhibition not only causes the rear head to be more likely to detach, and thus make for a more efficient walking mechanism, but also prevents the entire molecule from detaching from the microtubule altogether.

To further complicate matters, the motions of the two monomers of cytoplasmic dynein have been experimentally visualised and seen to be almost independent from one another unless the two monomers are relatively far from one another [8], making allosteric

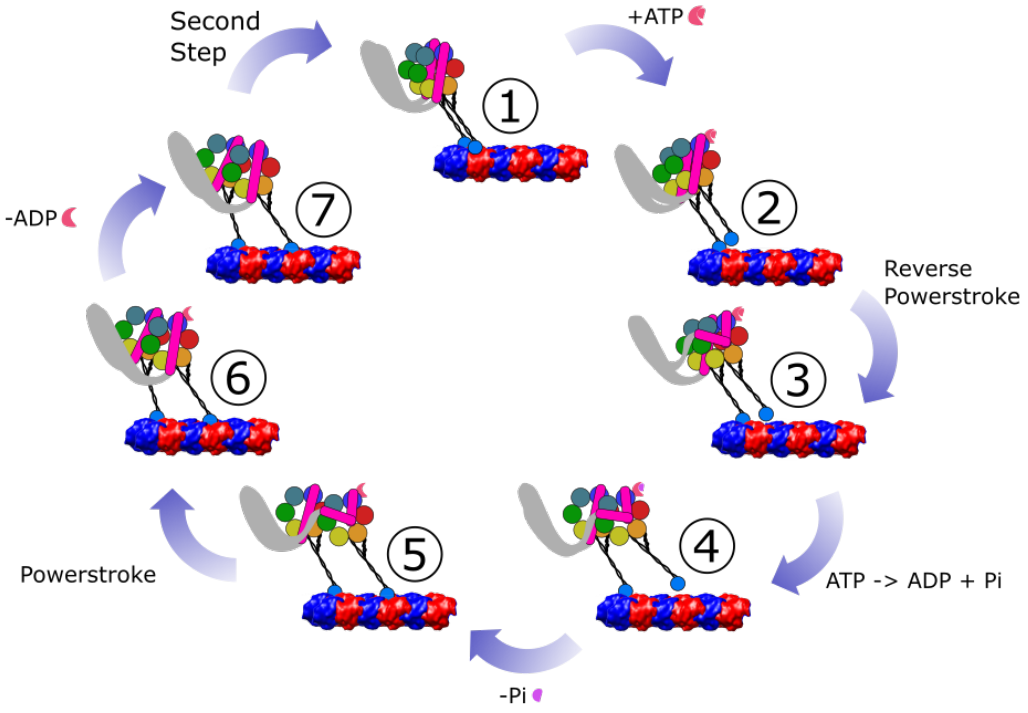


FIGURE 5.3: The ATP hydrolysis cycle of a single step of a monomer of cytoplasmic dynein. **1)** The MTBD is tightly bound to the microtubule. **2)** ATP binding to AAA1 weakens the MTBD microtubule affinity. **3)**Internal rearrangements trigger the priming of the linker domain. **4)** ATP is hydrolysed as the MTBD diffuses to the next binding site. **5)** Phosphate release causes the MTBD to readopt the high affinity conformation, and rebinding to the microtubule occurs. **6)** Rebinding triggers the powerstroke, pulling the trailing motor and associated cargo forwards. **7)** ADP is released, keeping the MTBD in its high affinity state until ATP binds again.

communication seem highly unlikely altogether. Knowing how a single monomer functions, we pose the following question: by what mechanism is the *dimer* cytoplasmic dynein able to walk without detaching from the microtubule?

5.2.1 Cytoplasmic Dynein - How Can It Walk?

For cytoplasmic dynein to be a processive motor it must have two basic properties. Firstly, each monomer must be able to detach from the microtubule track and reattach to a different binding site further along. Secondly, both monomers must be highly unlikely to detach at the same time, as this would likely cause them to diffuse away from the microtubule track. Here we discuss how this may be accomplished for cytoplasmic dynein given the relatively weak mechanical coupling between the monomers.

The most simple answer would be that there is little to no communication between the monomers, and a detached monomer has sufficient time to freely diffuse to the next

binding site. This was postulated by Qui *et al.* , who assumed that the high duty ratio of the molecule compensates for the lack of communication [131]. The rate limiting step of the ATP hydrolysis cycle for dynein, r_l, is the ADP release rate, and the most conservative estimate for this is $r_l = 160$Hz [132]. Therefore, we assume that a dynein monomer releases from the microtubule at this rate. Under what conditions, then, is the partner monomer able to diffuse to the next binding site in time?

We first consider some form of undirected diffusion. The 'weakest' form of this would be if the detached dynein monomer could freely rotate about the GST connection to the partner monomer, and also undergo translational diffusion subject only to the GST domain flexibility. Let us consider free rigid-body rotation together with translational diffusion limited to the microtubule axis. In this case, we can calculate an approximate time taken for the MTBD of dynein to coincide with a binding site using the set of dynein parameters collected by Sarlah *et al.* [132].

The total length of a dynein monomer from the MTBD to the end of the motor domain, $l_D \approx 25$nm, and its largest radial component (the motor), $r_D \approx 5$nm. Using these values with Equation 3.28, the approximate drag on this molecule when immersed in a water-like medium is $\lambda = 3.91 \times 10^{-10}$Ns/m. The distance between binding sites, $l_{bb} \approx 8$nm. Assuming l_{bb} to be the required root-mean squared displacement to find a new binding site, we can use the equation for free diffusion, $l_{bb}^2 = \langle x^2 \rangle = 2Dt_{bb}$ to determine an approximate transit time, t_{bb} between binding sites. For $D = k_B T/\lambda$, we find that $t_{bb} = 3.04\mu$s, i.e. for a molecule the size of dynein undergoing 1D translation only, we expect it to move between binding sites in approximately this amount of time. We have not taken the action of ATP hydrolysis into account, which causes the monomer to adopt the pre-powerstroke state and likely biases the diffusion in the forward direction. Even without this process, the number of times a dynein monomer will be able to diffuse between binding sites within a single ADP release, $n_{bb} \approx 2056$, which shows that it is almost certain to occur. However, rotation of the monomer must also be taken into account. If it is truly free rotation, then we cannot simply use a root-mean squared displacement argument as we did with translation, as the MTBD needs to be in a specific orientation. The long-time probability of the monomer being in the correct rotational orientation for a given binding site, P_r, is the ratio of the areas of the MTBD, $A_{MTBD} \approx \pi r_s^2$, and the total spherical surface area it sweeps out, A_S, where $r_s \approx 1$nm is the radius of the MTBD. If it rotates about its linker connection, the radius of this spherical surface is approximately the length of the stalk domain, $l_s \approx 12$nm. This gives $P_r = A_{MTBD}/A_S = 1.74 \times 10^{-3}$. The total number of times the MTBD is both in the correct position along the microtubule axis and in the correct

orientation within the transit time, $N_{bb} = n_{bb}P_r \approx 4$, with a total transit time between two adjacent binding sites including rotation, $T_{bb} = 1.75$ms.

This simple calculation shows that for a dynein monomer freely rotating about the GST connection, we would expect both monomers to detach from the microtubule every 4 ATP cycles, which would correspond to run-lengths of only $32nm$ if all of the steps were in the forward direction. However, experiments have shown that cytoplasmic dynein has run-lengths closer to 1.9μm, or 238 steps in the forward direction [133]. Of course, there is significant uncertainty in our calculation. The flexibility of the GST domain would allow for diffusion both around and away from the microtubule, both of which would increase the transit time. Additionally, the initial dependence of the expected MTBD position on its initial position (see Appendix C) may mean that the MTBD remains much closer to the microtubule over the course of a single ATP cycle, increasing the area ratio and lowering the total transit time. But our estimate, together with our knowledge of the other motors, does encourage the idea of some additional interaction, either mechanical inhibition or otherwise, that enables a dynein monomer to reach its next binding site faster. Both myosin and kinesin exhibit some form of communication, so considering these motors may give us some insight into what mechanisms are possible.

Kinesin achieves processivity by heavily biasing the diffusional search for the new binding site in the forward direction. As we saw previously, kinesin has a relatively short linker domain compared with myosin and dynein. This allows significant mechanical strain to build between the two motor domains, as well as lessening the diffusional volume the detached motor is able to search over. The powerstoke of kinesin also biases the search in the forward direction by throwing the unbound motor directly towards its next binding site. Indeed, the processivity of a kinesin motor has been shown to be heavily dependent upon the length of the linker domain [134] or more specifically, the distance between the microtubule bound domain and its unbound partner. Some monomeric forms of kinesin, which do not have this mechanical strain method of communication, exhibit a non-specific attraction to the microtubule [135]. This reduces its diffusion to a 1D search along their respective protofilaments, likely as a compensation for the lack of directed diffusion enabled by the power-stroke in the dimeric form.

Myosin V on the other hand has been seen to perform something close to 2D surface diffusion to find its next binding site [136], with one linker and the associated unbound motor domain acting somewhat like a rigid rod pivoting about its connection to the other linker and bound motor domains. As we have previously discussed, there is mechanical inhibition through the rigid linker domains that inhibits the lead motor

from detaching when both heads are in the attached state, prioritising forward steps, but once detachment has occured and the search is underway, mechanical strain can theoretically relax in the molecule. Therefore, it is only the rate limiting step of ADP realease in myosin V, $12Hz$ [109], that prevents the rear head from detaching as the already detached head searches for a new site. Myosin V exhibits intermediate mechanisms other than the powerstroke that bias the rotational search in the forward direction [137], and together with the very slow ADP release rate, allow myosin to find a new binding site through an effectively free rotational diffusion. Cytoplasmic dynein may have a similar directional bias through its reverse powerstroke, but nevertheless, it has a much faster ADP release rate than myosin V. The entire dynein monomer is approximately the same size as a myosin V linker domain [114], meaning that their diffusion is approximately equal. Although myosin V has to diffuse 36nm along an F-actin filament to find its next binding site, whereas cytoplasmic dynein only needs to diffuse 8nm along the microtubule, myosin V seems to have a much more limited diffusional range due to its specific hinge joint between the two linker domains, whereas a dynein monomer has much more freedom allowed by the flexible GST region.

From a mesoscopic structural perspective, cytoplasmic dynein seems to be closer to myosin V than kinesin in its diffusional characteristics. However, as it shares a track with kinesin, we may hypothesise that some of its diffusional dynamics are similar to the forms of kinesin which have no strong mechanical connections between them, i.e. the monomeric forms. Recents simulation studies have indeed shown that cytoplasmic dynein may feel a long-range electrostatic attraction to the microtubule track, which guides its diffusion to the next binding site [138]. Our interpretation of all of the available evidence is that cytoplasmic dynein take a single step through an almost free diffusion, as myosin V does, but with some sort of interaction with the microtubule, as kinesin does.

Recent experiments have taken advantage of the technological advances in cryo-electron microscopy [72] to generate new data that gives novel insight into the action of cytoplasmic dynein [8]. New high-resolution structural data has also been obtained using x-ray crystallography [9], allowing structural interpretation of the experimentally observed dynamics. The remainder of this chapter describes our use of this data to build a full FFEA simulation suite of the cytoplasmic dynein system. We discuss continuum mechanical parametrisation of the individual monomers from both experimental studies and higher resolution molecular dynamics simulations. We then look at parametrisation of the entire dimer, and see what we can learn about the walking mechanism of the molecule by using FFEA simulations to probe the long-time dynamics of cytoplasmic dynein.

5.3 FFEA Modelling of Cytoplasmic Dynein from MD

One of the major topics in the development of the FFEA method is determining a robust method of general parametrisation for FFEA systems. Generalised force-fields cannot necessarily be developed for FFEA models due to the differences in elastic and viscous properties for structures that seem to be identical at the mesoscale. For example, the coiled-coil structures present in myosin, kinesin and dynein, although structurally similar at the mesoscale[1], do not have the same flexibilities. If an all-atom MD structure is available however, then simulations can be performed on the entire structure, or simply segments of it, and flexibility data extracted. These can be converted to FFEA material parameters to reproduce the same flexibility seen in the higher resolution MD, and used in subsequent FFEA simulations to probe emergent properties that are often beyond the capabilities of MD due to time restrictions. This is the type of empirical modelling we will be using for the remainder of the chapter. We begin with a possible parametrisation from existing high-resolution MD simulations.

5.3.1 Building an FFEA Model of Dynein - MD Model

In collaboration with Kamiya *et al.* [41] we have had access to long-time, all-atom MD simulations of *dictyostelium discoideum* dynein from which we are able to extract flexibility data. Kamiya *et al.* built two atomistic dynein structures, one with ADP present in the AAA1 primary binding pocket (ADP model) and one with ATP present instead (ATP model). The other secondary binding locations at AAA2, AAA3 and AAA4 all had ADP present [41]. These systems were then energy minimised to form two distinct structures, and a 200ns MD simulation was performed for both models in explicit water (TIP3P water model) using the AMBER MD package, with a modified force-field optimised for swift electrostatic calculations [139]. We were able to extract the average structure for each model from the simulation and, using VMD, calculate an electrostatic surface to capture the overall shape of the molecule. From here, we used a surface collapse algorithm to coarsen the system to an acceptable resolution. Due to the resolution used to calculate the electrostatic surface, we also needed to manually detach the linker domain from the motor domain to give it the freedom to fluctuate independently from the motor. This was achieved using the Blender computational modelling software [4]. Finally, we populated the triangulated surface with tetrahedra using the Tetgen software package [65], completing the FFEA modelling procedure. The entire process is shown in Figure 5.4.

[1] Consider just the secondary structure, for example

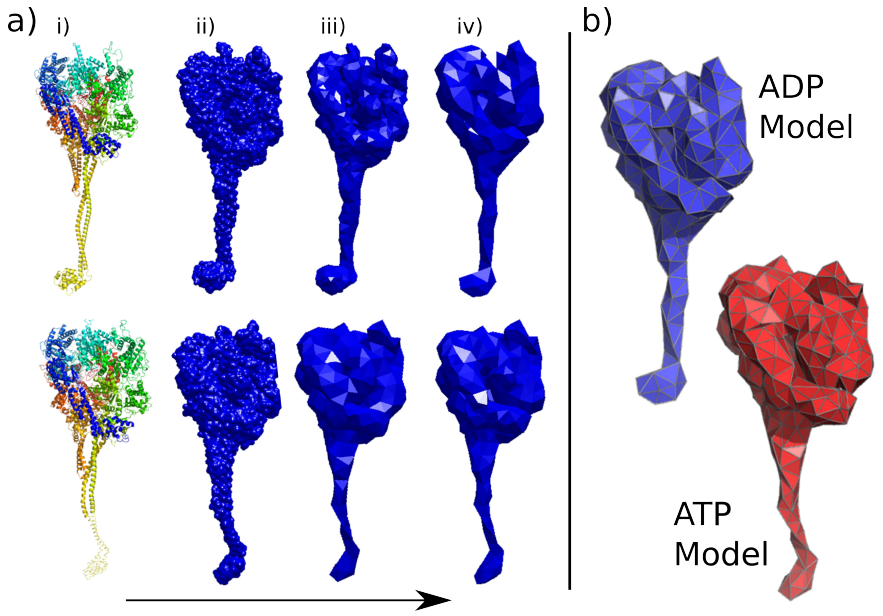


FIGURE 5.4: FFEA conversion procedure for the atomistic models of *dictyostelium discoideum* dynein. **a)** The conversion process. **i.** The average atomistic structure calculated from the 200ns MD simulations. **ii.** The electrostatic surface calculated using VMD. **iii.** First coarsened triangulated surface. **iv.** Final coarsened surface with linker domain separated from motor. **b)** The final FFEA models visualised in PyMOL for the two structures.

5.3.2 Parametrisation of Monomeric Dynein - MD Model

In order to parametrise FFEA models in the current framework, a manual analysis of the underlying atomistic trajectory is required, the dynamics of which can then be reproduced within an FFEA simulation.

5.3.2.1 Atomistic Dynein Models

Initial analysis of the MD simulations of the two atomistic dynein models provided flexibility data for the stalk region of dynein [41]. An axial stalk vector can be defined by taking residues at both ends of the stalk, and by aligning the motor domain within each frame of the simulation to its average structure (see Figure 5.5), the variance in both angle and length of the stalk can be calculated and converted into stiffnesses,

$$\begin{aligned} \langle U(x) \rangle &= \frac{1}{2} k_x \langle (x - \bar{x})^2 \rangle, \\ \Rightarrow k_x &= \frac{k_B T}{\langle (x - \bar{x})^2 \rangle}, \end{aligned} \tag{5.2}$$

where the equipartition theorem has been applied to the general coordinate x to give an associated stiffness k_x for the motion of that coordinate. Kamiya *et al.* performed this

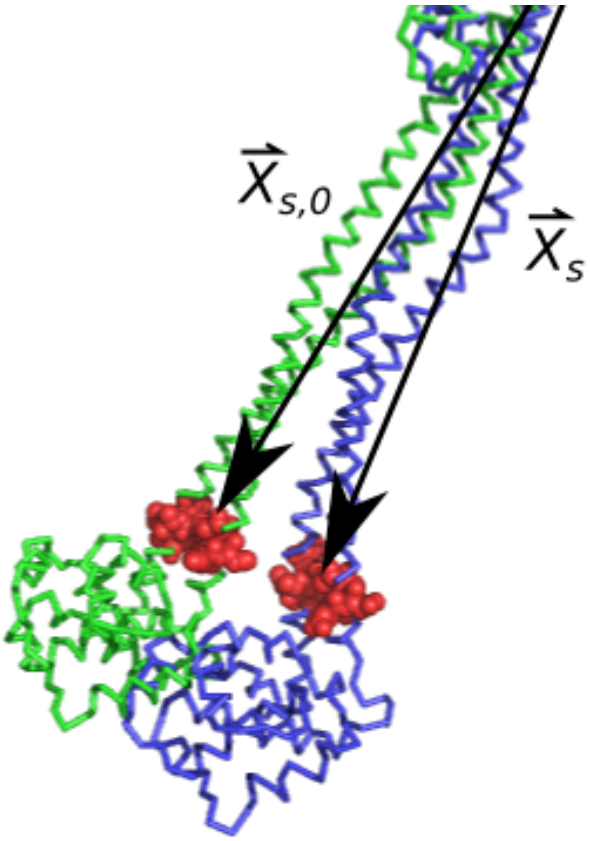


FIGURE 5.5: The definition of the stalk vectors for our analysis of the dynein molecular dynamics simulations. $\vec{X}_{s,0}$ is the vector defining the stalk of the average structure (coloured in green). The variance in stalk angle and length are calculated relative to this vector for each trajectory frame, (example coloured in blue). Similar vectors were defined for the linker domain.

calculation for the length and angular fluctuations of each helix within the coiled-coil of the stalk, and determined that the presence of ATP in the AAA1 domain increases the flexibility of the stalk. A change in flexibility is consistent with the coil-coil sliding mechanism seen by Schmidt, where ATP binding to the motor domain was seen to allosterically communicate with the MTBD [125]. However, as the coil-coil sliding was explicitly noted to be missing from the MD simulations, the increase in flexibility may be due to a different mechanism. Appendix C (specifically Equation C.14) does show that any stiffness calculated from the variance of insufficient simulation trajectory data should be proportional to the actual stiffness, which it exponentially decays towards. Hence, assuming the stalk was subjected to approximately the same viscous drag in both simulations, we expect that this increase in flexibility is representative of a real change in the free energy landscape of the monomer, and not an artefact of the lack of simulation data.

Using a single vector to define the motion of the stalk means that we can parametrise

the stalk domain within FFEA only as a single, homogeneous object. We performed similar calculations to Kamiya *et al.* , using the same residues at the base and tip of the stalk but defining only a single stalk vector, $\vec{X}_s$, as,

$$\vec{X}_s = \vec{t}_s - \vec{b}_s, \tag{5.3}$$

where $\vec{t}_s$ and $\vec{b}_s$ are calculated as the centroids of the tip residues ($\mathcal{T}_s$) and base residues ($\mathcal{B}_s$) respectively,

$$\begin{aligned} \vec{t}_s &= \frac{1}{N}\sum_{i}^{N} \vec{x}_i \quad i \in \mathcal{T}_s, \\ \vec{b}_s &= \frac{1}{M}\sum_{i}^{M} \vec{x}_i \quad i \in \mathcal{B}_s, \end{aligned} \tag{5.4}$$

We also defined a vector for the linker domain, $\vec{X}_l = \vec{t}_l - \vec{b}_l$, using a synonymous set of residues at the base and tip of the linker.

We calculated the distributions of the changes in length and angle for both the stalk and the linker with respect to their average structures over the course of the simulations. Figures 5.6 and 5.7 show examples of the probability distributions emerging from the simulation for the linker length and angles respectively, both from the ADP model. For the linker length we see a well defined Gaussian distribution, as expected for a Boltzmann distribution in energy. For the linker angle we obtain a chi-distribution, still corresponding to Boltzmann distribution for the case where we do not define negative values i.e. for angles.

Treating the stalk as a bending beam as in Section 3.2.1.2, a simple calculation for the viscous time constant for stalk length $L = 12$nm, width $w = 1.5$nm and biologically representative material parameters gives $\tau_k \approx 16$ns. Figures 5.8 and 5.9 show the evolution of the variance over the course of the simulation which results in these distributions. These traces indicate that the variance has almost converged, and therefore that 200ns seems to be sufficiently long to explore the entire conformational space available for these degrees of freedom. However, the stalk angular variance in the ADP model, shown in Figure 5.10, shows a clear discontinuity in the variance trace, as did the stalk length variance. These are likely the result of the molecule escaping from local energy minimum, something not considered in our simplified beam bending analysis, and therefore implies the existence of a larger free energy landscape that the molecules have not yet had time to explore.

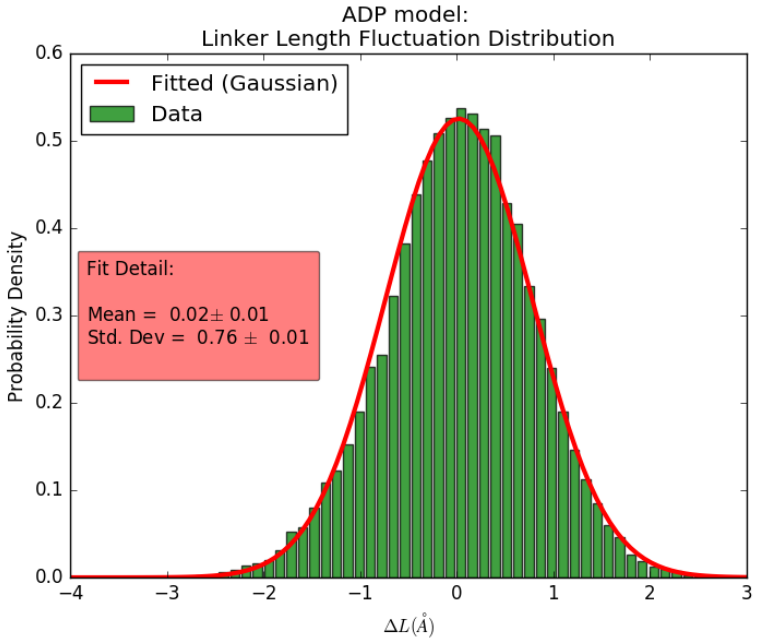


FIGURE 5.6: Probability distribution for the linker length in the ADP atomic model of dynein.

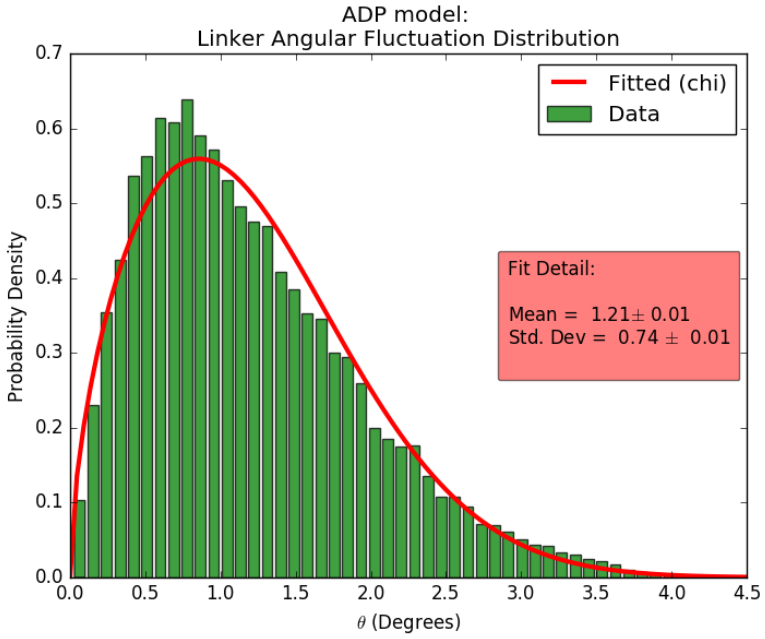


FIGURE 5.7: Probability distribution for the linker angle in the ADP atomic model of dynein.

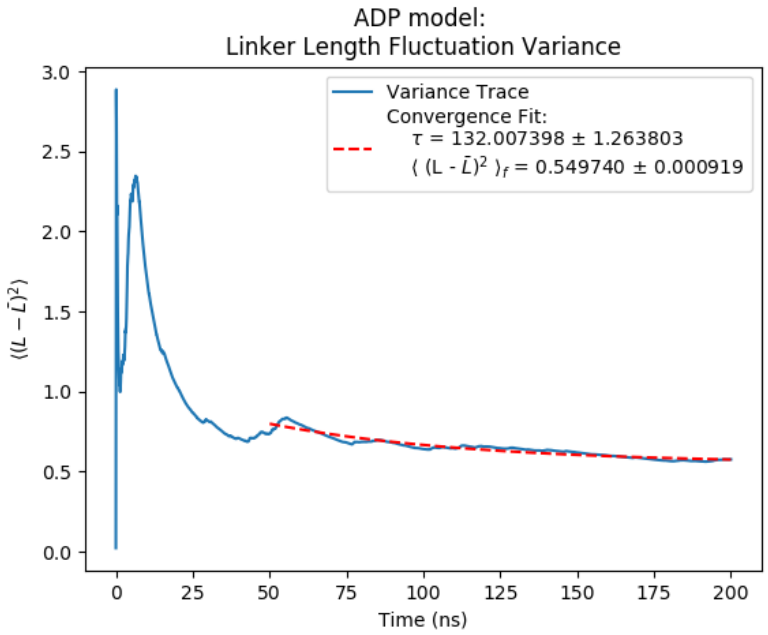


FIGURE 5.8: Evolution of the variance of the ADP model linker length over the course of the simulation.

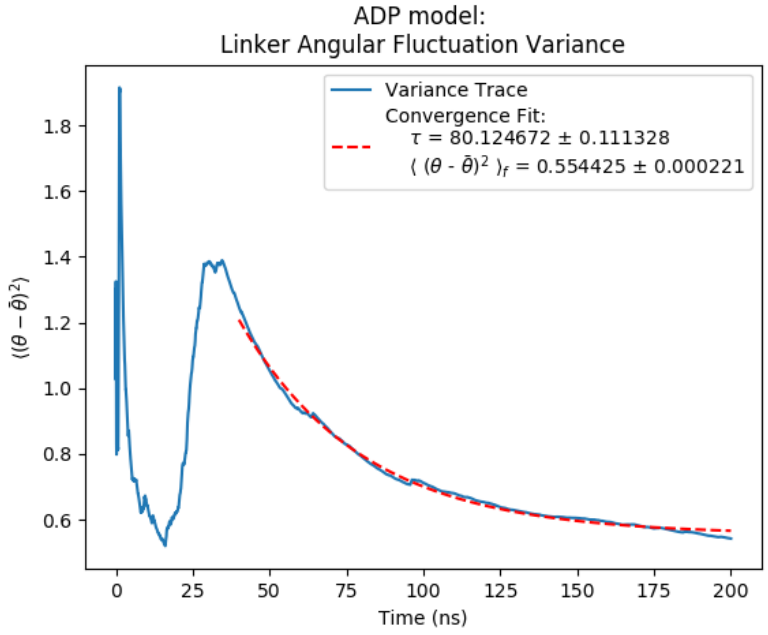


FIGURE 5.9: Evolution of the variance of the ADP model linker angle over the course of the simulation.

Variances	ADP Model	ATP Model
Stalk Length	(2.111 ± 0.001) Å^2	(11.70 ± 0.01) Å^2
Stalk Angle	$(10.030 \pm 0.008)^{\circ 2}$	$(21.55 \pm 0.02)^{\circ 2}$
Linker Length	(0.550 ± 0.001) Å^2	(0.4431 ± 0.0001) Å^2
Linker Angle	$(0.5544 \pm 0.0002)^{\circ 2}$	$(2.557 \pm 0.0007)^{\circ 2}$

TABLE 5.1: The extrapolated limit of the fitted variance traces of the atomic simulations of dynein monomers. Errors reported are the standard errors of exponential fit parameters.

To better approximate the flexibility of the model given potentially insufficient simulation runtime, we fit an exponential decay to this trace of the form Equation C.9 in order to extrapolate the data. We note that there is both excessive noise and an initial spike in variance in the early stages of the simulation, which corresponds to the equilibration phase. Hence, we assume the exponential fit to be valid only beyond $40ns$.

To fit our extrapolation curves to the data, we used the least-squares fitting algorithm implemented in the SciPy Python package and chose the segment of data after this characteristic initial spike which gives the best fit based on the normalised least-squares residual value (residual per data point). The limiting variance values for these curves are shown in Table 5.1 for each of the degrees of freedom and for each model.

The relative size of the discontinuities in some of the trajectories make our exponential fit an inappropriate extrapolation method over the total simulation time. In these cases, we have manually chosen the range over which we extrapolate the data to better approximate the final tendency of the variance. It is unlikely that this is representative of the true total variance, but it does represent the local environment of the molecule at that time in the simulation.

Although our flexibilites cannot be directly compared to those of Kamiya *et al.* due to differences between our definitions of the vectors, our calculated values for the stalk properties are qualitatively equivalent, with differences in flexibility of the same order of magnitude between the ADP and ATP models. Minor differences in our values are likely due to our extrapolation procedure. Although the relationship of these calculated values to the true variance is questionable due to the potentially insufficient amount of simulation runtime from the clear discontinuities in the data, which imply localised energy minima, we justify the use of these values to parametrise the FFEA models as the most suitable approximation to use based upon the data available to us. With reference to Figure 5.3, we specify the ADP model as corresponding approximately to the post-powerstroke state shown (states 6,7 and 1) and the ATP model as corresponding to the pre-powerstroke state (state 3). The difference in flexibilites point towards

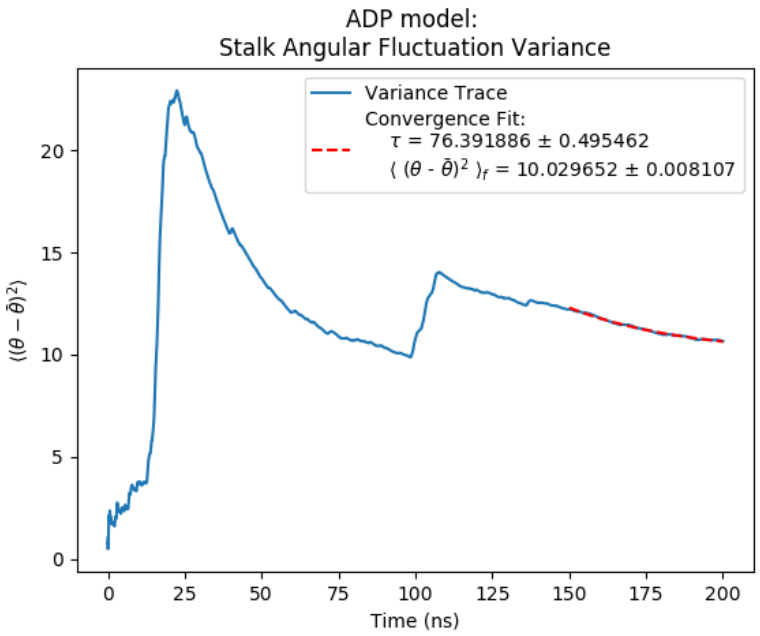


FIGURE 5.10: Evolution of the variance of the ADP model stalk angle over the course of the simulation. The clear discontinuity at 100ns $< t <$ 120ns may indicate escape from a metastable state, leading to difficulties in extrapolating the true variance from the available data.

some sort of conformational change within the stalk, consistent with this definition, although it was explicitly noted that the coiled-coil chain sliding observed by Schmidt *et al.* [9] was not observed in the MD.

5.3.2.2 FFEA Dynein Models

To determine the correct parameters for an FFEA simulation given the existing MD simulations, we performed a suite of FFEA simulations using the models shown in Figure 5.4. For both the ADP and ATP structures, we performed 7 FFEA simulations with Young's moduli, E, ranging from 100MPa to 100GPa to centre on a biologically realistic range [87], together with a constant Poisson ratio, with insights from Oliver [140], of $\nu = 0.35$. The resulting trajectories of each of these FFEA simulations were then mapped onto the atomic structures from which they were created using the structural mapping procedure shown in Section 4.5.3, which preserves the mesoscopic detail of the simulation [93]. From here, the same analysis performed in Section 5.3.2.1 was performed on the pseudo-atomic trajectories, allowing a consistent analysis procedure for both simulation types.

The simplicity of the FFEA simulations meant that convergence of the variance trajectories occurred much faster than the MD simulations. Figures 5.11 and 5.12 show

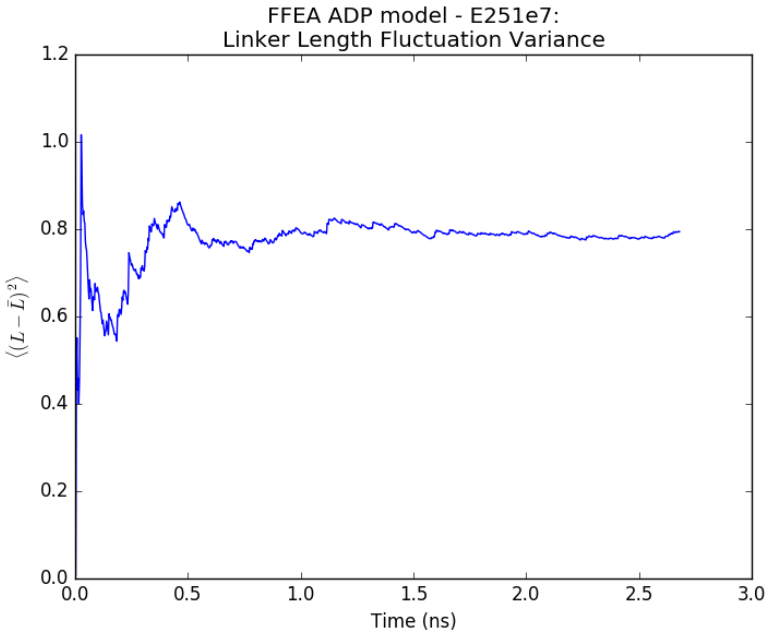


FIGURE 5.11: Evolution of the variance of the linker length over the course of the FFEA simulation corresponding to the ADP atomic model with Young's modulus, $E = 2.51\text{GPa}$.

examples of the results of the range of simulations, showing the linker length fluctuations of the ADP model and stalk angle fluctuations of the ATP model respectively.

5.3.2.3 Combining the Models

For a stiffness k_x of a general coordinate x, we expect $k_x \propto E$. From Equation 5.2, this implies,

$$\begin{aligned} E &\propto \frac{k_B T}{\langle (x-\bar{x})^2 \rangle}, \\ \Rightarrow E &= A \frac{k_B T}{\langle (x-\bar{x})^2 \rangle}, \end{aligned} \tag{5.5}$$

where A is a constant. Taking logarithms of both sides of Equation 5.5 gives,

$$\ln(E) = \ln(A k_B T) - \ln\left(\langle (x-\bar{x})^2 \rangle\right). \tag{5.6}$$

Equation 5.6 has the form of a straight line. Hence, for our set of simulations, we plot the logarithm of the extrapolated fluctuations against logarithm of the initial moduli to obtain a linear fit. Figures 5.13 and 5.14 show these fits for the linker angular fluctuations for the ADP and ATP models respectively. To each of these linear fits,

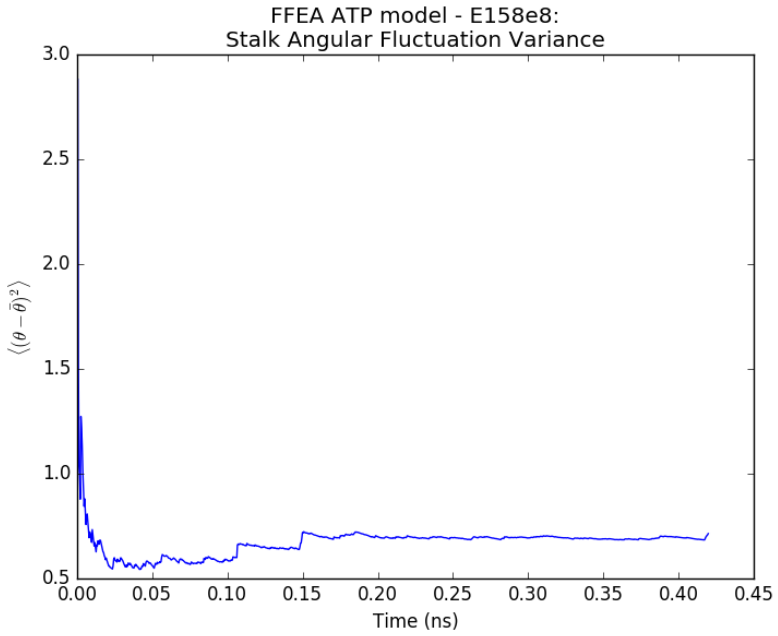


FIGURE 5.12: Evolution of the variance of the stalk angle over the course of the FFEA simulation corresponding to the ATP atomic model with a Young's modulus, $E = 15.8\text{GPa}$.

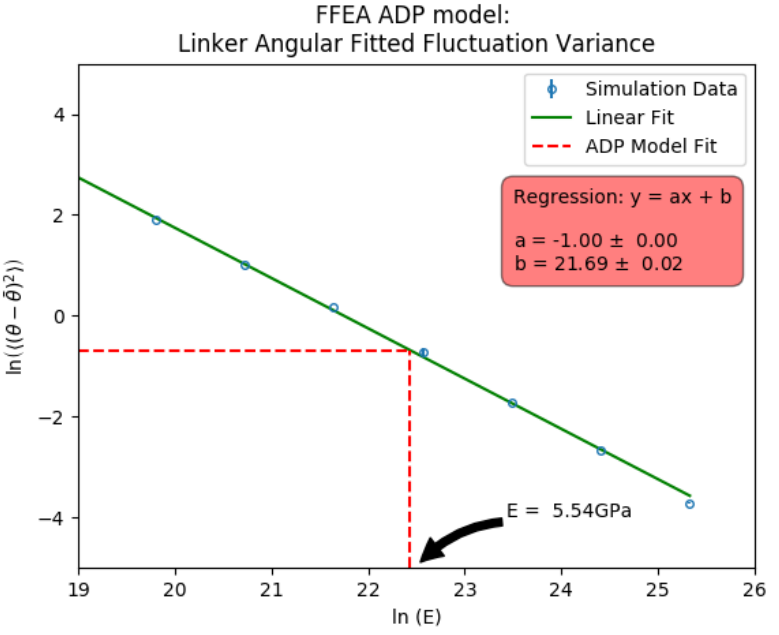


FIGURE 5.13: The linear relationship between the defined Young's modulus and measured linker angular variance (logarithms) in an FFEA simulation for the ADP model of dynein. By tracing the measured atomic fluctuations to the graph, we can extract the Young's modulus that reproduces the dynamics within an FFEA simulation.

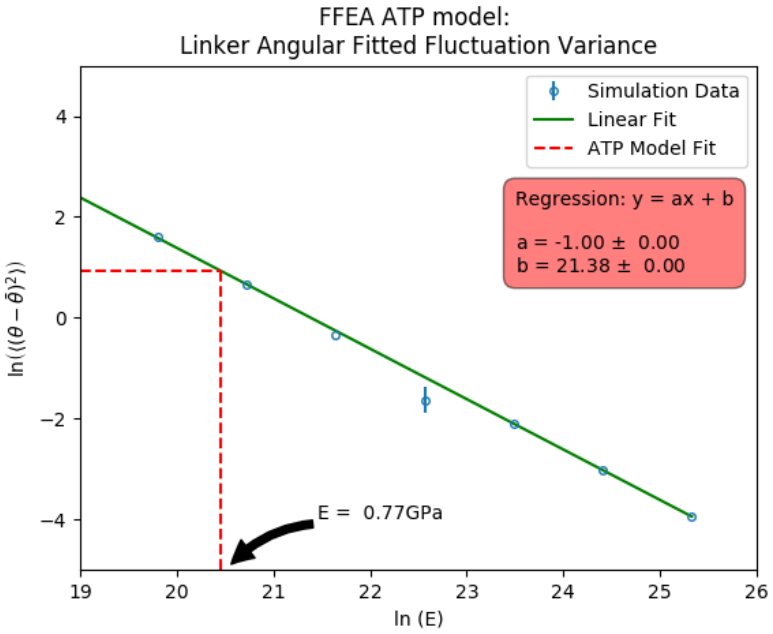


FIGURE 5.14: The linear relationship between the defined Young's modulus and measured linker angular variance (logarithms) in an FFEA simulation for the ATP model of dynein.

Young's Moduli	ADP Model	ATP Model
Stalk Length	2.22GPa	2.13GPa
Stalk Angle	1.08GPa	0.62GPa
Linker Length	4.06GPa	3.23GPa
Linker Angle	5.54GPa	0.77GPa

TABLE 5.2: The effective Young's moduli for the two dynein models required to reproduce the atomistic dynamics within an FFEA simulation. Values obtained from fitting atomic variance measurements to FFEA variance measurements.

we have determined the value of the Young's modulus which would reproduce the measured atomic fluctuation variance for each parameter of interest. The total set of measured Young's moduli extracted from the series of fits are shown in Table 5.2.

For an ideal linear beam, we would expect that the modulus corresponding to the angular and length fluctuations would both be identically equal to the Young's modulus. We see differences between the Young's modulus obtained for length and angular fluctuations in both the linker and stalk of both models, implying (somewhat unsurprisingly) that these structures are not ideal beams. Therefore, homogeneous parametrisation in this fashion means that we can reproduce either the angular or length fluctuations of each model, but not both.

The ATP model is still more flexible on the whole in terms of the magnitude of the Young's modulus, the continuum stiffness parameter. However, the difference between

Parametrisation Variables	Stalk Length	Stalk Angle
Linker Length	Stalk: (2.39 ± 0.11) $Å^2$ Linker: (0.49 ± 0.02) $Å^2$	Stalk: (12.91 ± 0.60) $^{\circ 2}$ Linker: (0.56 ± 0.02) $Å^2$
Linker Angle	Stalk: (2.08 ± 0.09)$Å^2$ Linker: (0.64 ± 0.03 $^{\circ 2}$)	Stalk: (13.51 ± 0.72)$^{\circ 2}$ Linker: (0.65 ± 0.03 $^{\circ 2}$)

TABLE 5.3: The variances resulting from FFEA simulations of the ADP dynein model when parametrised using each variable combination from the analysis of the ADP model atomic trajectories. Errors reported are standard errors of the variance.

the Young's modulus values between the ADP and ATP models is not as prominent as the raw variance values obtained in Section 5.3.2.1 (Table 5.1). This is because the measured variances are dependent upon the geometry of the object, whereas the Young's modulus is an intensive property, independent of the geometry. The surface profile we generated from the ATP model has a thinner stalk than that of the ADP model, such that the ATP model will be much more flexible than the ADP model when parametrised by the values in Table 5.2, even though they have similar Young's moduli. Nevertheless, given these two models, parametrisation using these values gives the correct amount of flexibility for the molecule we are modelling.

To show the reliability of this parametrisation, we ran a final set of FFEA simulations parametrised by the values shown in Table 5.2, with a linear interpolation of the material parameters across the motor domain. The variances calculated from these simulations of the ADP model are provided in Table 5.3.

These parametrised models give variances to within 10% of the measured values from the atomic simulations shown in Table 5.1. The slightly larger error is likely due to the linear interpolation of material parameters over the motor domain giving additional sources of flexibility when compared to the homogeneous models used for the parametrisation.

5.4 FFEA Modelling of Cytoplasmic Dynein from Experimental Studies

We have seen that although approximate stiffness values for proteins can be extrapolated from MD simulations and reproduced in FFEA simulations, these values are perhaps more representative of a local energy environment rather than the full free energy landscape, given the short time of the all-atom simulations. However, for our models of cytoplasmic dynein, experimental evidence is available that enables a much

more accurate estimation of the monomeric parameters, which allows us to perform subsequent simulations on the full cytoplasmic dynein system.

Imai *et al.* performed a thorough investigation into the mechanical properties of the cytoplasmic dynein system via a statistical analysis of flexibility data of the molecule whilst bound to microtubules [8]. By utilising unstained cryo-electron microscopy (cryo-EM), the group were able to freeze a prepared and equilibrated sample of wild type, chimeric cytoplasmic dynein molecules in the presence of 3.7μM microtubule and 3.6mM ATP concentrations. The chimeric monomers were formed using the majority of the *dictyostelium discoideum* structure [141], with the stalkhead region, which contains the MTBD, replaced with that of human axonemal dynein as it has a higher overall binding affinity for the microtubule. The dynein monomers were artificially dimerised into the cytoplasmic dynein complex by expressing the *schistosoma japonicum* GST domain within the two monomers, coupling them by their linker domains without the disordered tail region seen in natural dynein [7]. Although mutations in the tail domain of certain dynein complexes in mice have been seen to inhibit cytoplasmic dynein motor processivity [142], the GST dimerised dynein is known to be capable of well-defined processive motion [131] and so is suitable for this simulation suite.

Following the cryo-EM procedure, Imai *et al.* determined the polarity of the microtubules within the sample, and used their presence to locate 10,080 individual dynein monomers using particle picking software. This sample was further reduced to 711 dimers by considering only pairs of monomers that were within 40nm of each other, or single monomers at least 40nm away from any other monomer, where 40nm is the total length of the GST domain at maximum linear extension. This reduction ensured that all pairs of monomers considered in the sample were unique dimers or single monomers. Image analysis was then performed on the data set to align the monomers with respect to various structural components visible in the raw images. Alignment of the 'single monomer' images with respect to the visible stalkhead domain showed that the normalised pixel density values were approximately twice that of the equivalent monomeric dynein images, showing that these molecules were actually dimers with superposed monomers on adjacent microtubule protofilaments. These superposed structures formed 46.3% of the total number of dimers, a significant amount that may point towards some as yet unknown interation between the dimers that keeps them superposed. The remaining 'offset' dimers were aligned with respect to the leading head under the assumption that the stalk flexes as a rigid body with a localised hinge region near the interface between stalk and MTBD. This analysis showed a variation in stalk angle with respect to the microtubule with a visible stalk emerging from the image averages. This implies that the flexibility is indeed mostly localised to the hinge

region, thought to be caused by a conserved set of proline residues in the vicinity of the neck of the stalk [8]. We will use this flexibility data to determine structural parameters for each monomer in our model.

Using the same structure of dynein used in the cryo-EM studies, for which x-ray crystallographic data is available [143] with the stalk head replaced by that of *mus musculus* to give the higher affinity properties of the study, Imai *et al.* were finally able to build a full atomic structure for the dimer complex by fitting the atomic models to the available EM density data. We were given access to these atomic models in order to build our lower resolution FFEA structures.

5.4.1 Building an FFEA Model of Dynein - Experimental Model

As the experimental images were all bound to the microtubule in the presence of ATP, and the limiting step of the cytoplasmic dynein ATP hydrolysis cycle being the ADP release rate [132], we can assume that the majority of the experimental images correspond to the post-powerstroke, or 'unprimed' state of the monomers (states 6, 7 and 1 in Figure 5.3). We recall that the atomic ADP model also corresponds approximately to these conformational states. Figure 5.15a shows the FFEA model of this structure, which was built in an analagous manner to the models in Section 5.3.1, and which corresponds to atomic model A built by Imai *et al.* [8]. In preparation for the inclusion of a hinge region as predicted by the cryo-EM data, we manually remodelled the stalk region into a more regular topology with respect to the tetrahedral elements using the Blender modelling software [4]. This allowed us to manipulate the parameters of specific elements to generate specific types of motion. We note that if this was performed after the parametrisation procedure, the parametrisation would no longer be valid. However, as a generalised parametrisation is not yet available, we believe this to be an appropriate method at present. Figure 5.15 visualises the completed hinge, with specific detail on the parametrisation presented later in this section.

To build the pre-powerstoke / 'primed' model for a dynein monomer, we used Blender to manually detach the triangulated linker domain from the post-powerstoke model and reattach it to an FFEA structure built using the work of Schmidt *et al.* [9]. The linker was rotated by 90° with respect to the motor domain before being reattached, approximately corresponding to the linker domain structure reported by Schmidt *et al.* This is shown in Figure 5.15b. A visual inspection shows that our surface model is sufficient to capture the differences in overall shape between both motor domains that

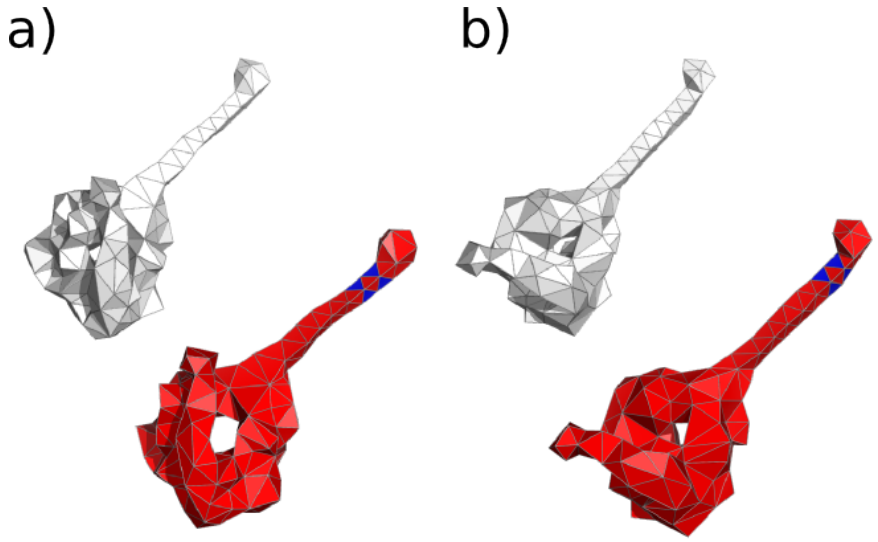


FIGURE 5.15: FFEA models constructed from the atomic models built by Imai *et al.* based on cryo-EM [8] and crystallographic [9] experimental results, and the modified models that allow a hinge region to be defined. **a)** is the post-powerstroke model, and **b)** the pre-powerstroke model, with the colour images showing a higher flexibility region (the hinge) defined at the stalk-stalkhead interface.

have been remodelled through the ATP hydrolysis process [125], as well as the clear differences in linker position corresponding to the action of the powerstroke.

Following the construction of the monomer, we built a full microtubule complex comprised of 14 protofilaments, each containing 8 tubulin monomers. This represents a 96nm length of track for the cytoplasmic dynein FFEA model to interact with. The surface profile generated for the atomic structure of each tubulin monomer was of a high enough resolution to define a binding site between the α and β domains [126]. We reconstructed a full microtubule complex, including the 0.9nm offset between protofilaments, using simple geometric translations and rotations of the single tubulin monomer. Finally, we placed a dynein monomer on the microtubule such that the stalkhead was within 5Å of the tubulin binding site, and another monomer in the equivalent position on the adjacent protofilament, with the angle between the stalk and microtubule being 41.5° as reported by Imai *et al.* for superposed dimers. The completed superstructure is shown in Figure 5.16.

5.4.2 Parametrisation of Monomeric Dynein - Experimental Model

Flexibility data reported by Imai *et al.* gives the variances of the angular distributions of the dynein monomer stalks with respect to the microtubule. For the superposed monomers this data is sufficient to immediately calculate an effective angular stiffness for each of the monomer stalks. However, the presence of the GST domain means that

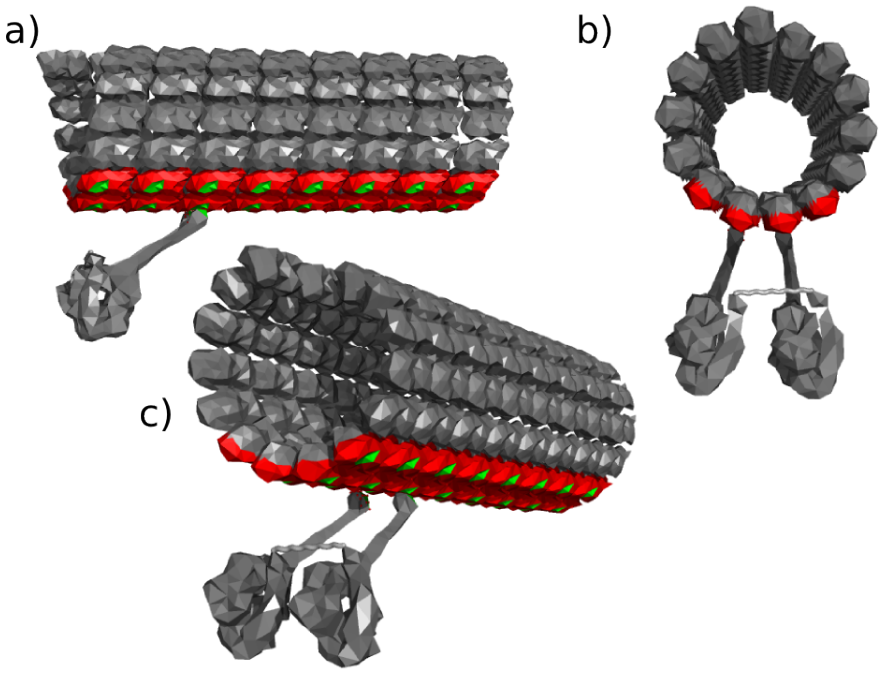


FIGURE 5.16: FFEA model of the completed cytoplasmic dynein system on the microtubule, visualised by the Van der Waals interaction locations. The two dynein monomers are initialised in the post-powerstroke state. **a)** shows the monomers to be in the superposed state, offset only by the 0.9nm difference in protofilament alignment in the axial direction. **b)** shows the GST domain, which we have modelled as a linear restraint between the tips of the two linkers. Finally, both **a)** and **c)** show the placement of the binding sites along the microtubule, coloured in green. The tip of each dynein monomer is also green, representing the MTBD in the high affinity state. The red colouring on the microtubules represent a possible weak attraction between the MTBDs and the protofilaments, similar to that seen between kinesin and tubulin. The remainder of the molecule is coloured grey to represent regions of zero attractive interaction.

the angular fluctuations between monomers will be correlated. We can use the superposed data to determine what this correlation should be, and hence find an effective spring constant for the GST domain.

If we reduce each dynein monomer to two beads (the motor domain and the stalkhead) connected by a spring (the stalk), with both monomers connected by an additional spring (the GST domain) we can write the total energy of this system as,

$$U_T = \frac{1}{2}\left(K(\theta_1^2 + \theta_2^2) + k_{12}\,|\vec{r}_2 - \vec{r}_1|^2\right), \tag{5.7}$$

where θ_1 and θ_2 are the angular deviations from the energy minimum state for each stalk, and $\vec{r}_1$ and $\vec{r}_2$ are the stalk vectors from stalkhead to motor domain. Equation 5.7, specifically the term involving k_{12} is valid only if the two monomers are bound in approximately the same location on the microtubule axis. In this case, we

can use the small angle approximation to give,

$$U_T = \frac{1}{2}\left(K(\theta_1^2 + \theta_2^2) + K_{12}(\theta_2 - \theta_1)^2\right), \tag{5.8}$$

where $K_{12} = k_{12}L^2$ for a stalk of average length L. As we have two degrees of freedom here, it is better to write Equation 5.8 as a matrix equation,

$$U_T = \frac{1}{2}\vec{\theta}^T\mathbf{K}\vec{\theta}, \tag{5.9}$$

where $\vec{\theta}^T = (\theta_1\theta_2)$ and,

$$\mathbf{K} = \begin{pmatrix} K + K_{12} & -K_{12} \\ -K_{12} & K + K_{12} \end{pmatrix} \tag{5.10}$$

The stiffness matrix $\mathbf{K}$ can be transformed into an angular covariance matrix $\mathbf{C}$ as we saw with Equation 3.42,

$$\begin{aligned} \mathbf{C} &= \langle\theta_i\theta_j\rangle = k_BT\mathbf{K}^{-1}, \\ &= \frac{k_BT}{K(K + 2K_{12})}\begin{pmatrix} K + K_{12} & K_{12} \\ K_{12} & K + K_{12} \end{pmatrix} \end{aligned} \tag{5.11}$$

which is sufficient to calculate all stiffness properties for the cytoplasmic dynein complex.

For the set of superposed monomers, where $\theta_1 = \theta_2 = \theta$, the observed angular standard deviation was $\sigma_\theta = 11.5°$, calculated on a sample of 318 images. Using the statistics of higher order moments [144], $\langle\theta^2\rangle = (130 \pm 10)^{\circ 2}$. To have rigidly superposed monomers is the equivalent of $K_{12} \to \infty$ in Equation 5.11. In this case, we find that,

$$\langle\theta^2\rangle = \frac{k_BT}{2K}, \tag{5.12}$$

and therefore, $K = (52 \pm 4)$pN.nm. This is approximately half of the stiffness calculated by Imai *et al.* , which is expected for a system of two springs (stalks) in parallel. Note that these values are angular stiffness, and are therefore independent of whether the flexibility is localised to a hinge or not.

5.4.2.1 Homogeneous Stalk Parametrisation

We first emulated these stiffness values within our FFEA models using the same techniques as in Section 5.3. By performing a suite of 7 FFEA simulations with homogeneous parametrisations of varying Young's moduli and a constant Poisson ratio, we

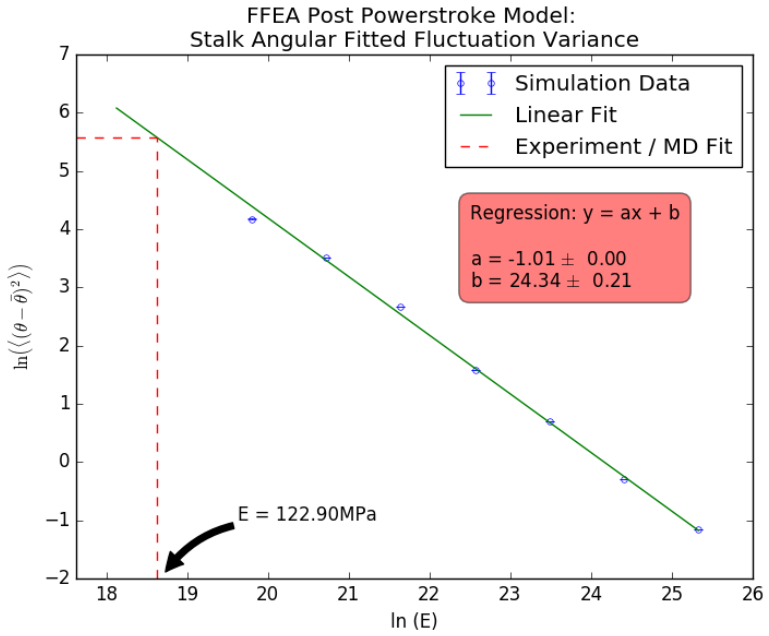


FIGURE 5.17: The effective Young's modulus required to reproduce the experimental observations of stalk fluctuations of the post-powerstroke state of cytoplasmic dynein.

were able to find a similar linear relationship between the logarithms of the stalk and linker angular fluctuations and their Young's moduli. For the stalk we were able to use the flexibility value calculated from Equation 5.12 to immediately determine an appropriate Young's modulus, which can be seen in Figure 5.17.

For the pre-powerstroke state, the experiments unfortunately provide us with no data. To compensate for this, we made the assumption that the proportional difference in flexibility between the ADP and ATP models from the MD simulations, K_{ADP} and K_{ATP} respectively, would be the same as for the experiment,

$$K_{pre} = K_{post}\frac{K_{ATP}}{K_{ADP}}. \tag{5.13}$$

Hence, using the values for stalk angular fluctuations from Table 5.1, we obtained $K_{pre} = (24 \pm 2)$pN.nm. The total set of Young's moduli extracted from these linear fits and MD comparisons, which give rise to the experimentally observed flexibilities, are shown in Table 5.4.

5.4.2.2 Hinge Region Stalk Parametrisation

The set of Young's moduli required to give the same experimentally measured stalk flexibilities but with the angular fluctuations localised to a hinge region, would be

Young's Moduli	Post-Powerstroke Model	Pre-Powerstroke Model
Stalk Angle	122.9MPa	73.73MPa
Linker Angle	9.29GPa	2.64GPa

TABLE 5.4: The effective homogeneous Young's moduli for the two dynein models required to reproduce the experimental dynamics within an FFEA simulation. Values obtained from fitting the theoretical analysis of the experimental systems to FFEA simulations, and comparison with MD simulations where necessary.

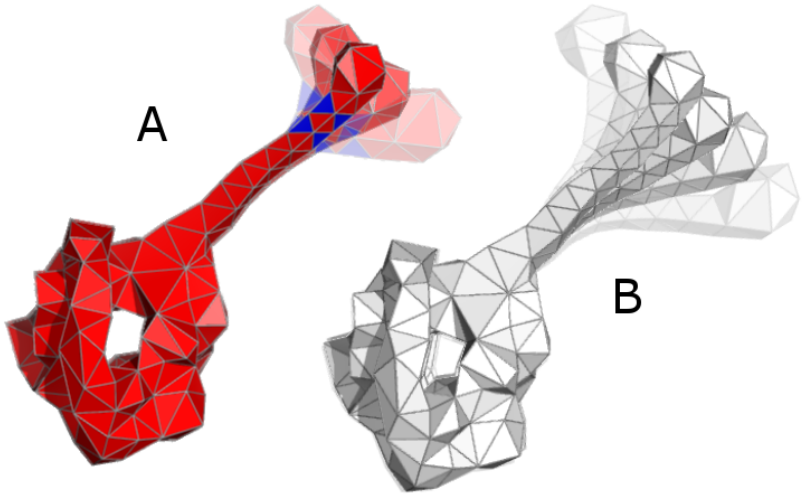


FIGURE 5.18: Animations of the most flexible normal mode in the FFEA models of the dynein monomers. **A** is the hinge model, with a significantly softer hinge region compared to the stalk, whereas **B** is homogeneously parametrised, with the bending beginning at the base of the stalk.

significantly lower than the homogeneous values. So low in fact that to perform the same process as above to parametrise the hinge (7 FFEA simulations with varying parameters) would require prohibitively long simulation run-times. Therefore, to find the set of Young's moduli that give the localised hinge flexibilities, we utilised the new linear elastic modelling capabilities of the software.

We first built two test parametrisations of the post-powerstoke state, one with a homogeneously parametrised stalk with Young's modulus $E = 1\text{GPa}$, and one with a smaller hinge region of Young's modulus $E = 1\text{GPa}$ and the remainder of the stalk 100GPa, representing a soft hinge attached to a rigid beam. Figure 5.18 shows the two primary elastic normal modes calculated using the FFEA LEM tool (see Chapter 3). We clearly see that for the hinge model, the motion is more localised to the hinge region compared with the homogeneous model. The difference is verified by the dot product between the two relevant eigenvectors, $\vec{e}_1 \cdot \vec{e}_2 = 0.725$, a significant deviation from unity.

Young's Moduli	Post-Powerstroke Model	Pre-Powerstroke Model
Stalk Angle	28.67MPa	17.20MPa
Linker Angle	9.29GPa	2.64GPa

TABLE 5.5: The effective hinge Young's moduli for the two dynein models required to reproduce the experimental dynamics within an FFEA simulation. Stalk values obtained from scaling the homogeneous values from the initial homogeneous parametrisation. Linker values remain unchanged.

Due to the Cartesian representation of the modes, the eigenvalue corresponding to each mode represents its linear stiffness and so is dependent upon the geometry of the stalk. For the hinge model flexibility, we obtained an eigenvalue $e_h = 1.360\text{N/m}$ and for the homogeneous model flexibility, $e_s = 1.714 \times 10^{-2}\text{N/m}$, which shows that the homogeneously parametrised stalk is more flexible overall. If we compare both the homogeneous stalk and hinge regions to linearly bending beams, then from Equation 3.26 we would expect our stiffness eigenvalues to have the following property for the same value of the Young's modulus,

$$\frac{e_h}{e_s} = \left(\frac{L_s}{L_h}\right)^3. \tag{5.14}$$

Substituting the eigenvalues into Equation 5.14 gives $e_h/e_s = 79.35$. Using the measurement tools in the PyMOL visualiser, we determined our total stalk length as $L_s = (15 \pm 1)\text{nm}$ and the hinge region as $L_h = (3.5 \pm 0.2)\text{nm}$, where our errors are the measurement uncertainty in how to define the correct stalk lengths given the modes. Substituting these into Equation 5.14 gives $\left(\frac{L_s}{L_h}\right)^3 = 78.7 \pm 20.7$, which is only a 0.82% difference from the eigenvalue ratio and shows that Equation 5.14 is valid within the FFEA environment.

We wish to find a Young's modulus for the hinge region that gives the same angular fluctuation as for the homogeneous model. Again using Equation 3.26 and remembering that an angular stiffness K is related to a linear stiffness k as $K \propto kL^2$, then the hinge Young's modulus E_h is related to the homogeneous Young's modulus E_s by,

$$\frac{E_h}{E_s} = \frac{L_h}{L_s}. \tag{5.15}$$

The values of the Young's moduli, then, are scaled by the fractional difference in length between the two models. Our Young's modulus values for the hinge region are shown in Table 5.5.

Due to the high sensitivity of this method of parametrisation on the geometry of the object, these hinge localised Young's moduli are a clear approximation to the true values. However, the experimental results of Imai *et al.* , and more recently simulation studies from Li *et al.* [138], clearly show the existence of localised flexibility about the

stalk neck region. Indeed, on closer inspection, PCA analysis on the stalk domains of the MD simulations also show the most flexible modes to be localised to a hinge region even when the structure is not bound to a microtubule, as it was in the experimental studies. Hence, we will use the hinge parameterisation for all simulations for the remainder of this chapter, with the remainder of the stalk region treated as a relatively rigid beam, ($E = 100$GPa).

5.4.3 Parametrisation of the Cytoplasmic Dynein Complex

All that remains for a reasonable structural parametrisation of the entire cytoplasmic dynein system is to calculate the effective stiffness of the GST domain. We can accomplish this by using the variance of the angular difference, $\langle(\theta_2 - \theta_1)^2\rangle$, reported by Imai *et al.* without the assumption of $K_{12} \to \infty$,

$$\langle(\theta_2 - \theta_1)^2\rangle = \langle\theta_1^2\rangle + \langle\theta_2^2\rangle - 2\langle\theta_1\theta_2\rangle. \tag{5.16}$$

Substituting the individual elements from $\mathbf{C}$ in Equation 5.11, we obtain,

$$\langle(\theta_2 - \theta_1)^2\rangle = \frac{2k_BT}{K + 2K_{12}}, \tag{5.17}$$

and rearranging for K_{12},

$$K_{12} = \frac{k_BT}{\langle(\theta_2 - \theta_1)^2\rangle} - \frac{1}{2}K. \tag{5.18}$$

The quoted experimental values for $\langle(\theta_2 - \theta_1)^2\rangle$ vary as the stalkhead separation increases, with the GST region becoming stiffer as the relative stalkhead positions force it into a high energy state. Given the relatively small sample size of experimental observations, we choose $\sigma_{\theta_2-\theta_1} = (17 \pm 2)^\circ$, a value approximately averaged over the entire range of possible stalkhead separations. Our error value is set to encompass the total range of measured variances. This gives us a final stiffness value for the GST domain of $K_{12} = (21 \pm 5)$pN.nm.

5.4.4 Simulations of Cytoplasmic Dynein - Microtubule Diffusion

From the fully parametrised system developed over the course of this chapter, we began simulations to study how dynein functions given its structure. We begin by looking at the potential diffusional dynamics along the microtubule track.

The current theoretical model of the ATP hydrolysis cycle of cytoplasmic dynein (shown in Figure 5.3) predicts that ADP release is the rate limiting process for each

step along the microtubule. With ADP bound to the AAA1 binding site, a dynein monomer will bind to the microtubule. We model this in the left-hand monomer of the system simply by using 'pinned nodes', node which do not move within the FFEA simulation, to prevent the stalkhead from moving away from the initial position. This allows the remainder of the monomer, and especially the stalk, to retain its flexibility. We leave the remaining monomer as it is, connected to the bound monomer by the GST domain but able to freely detach from the microtubule. By modelling the GST domain as a simple Hookean spring, our simulations will not capture the superposed states experimentally visualised by Imai *et al.* [8]. However, due to a lack of information on what the causes of these superposed states may be, we chose to neglect that particular interaction.

We studied the diffusion of this dynein monomer as a function of a non-specific, attractive potential between the MTBD and the microtubule track, similar to that seen in monomeric kinesin [121]. The potential form used was the Lennard-Jones and steric coupling developed in Section 3.8.1, a short range potential not dependent upon specific electrostatic interactions. We performed simulations with 12 different interaction potentials with minima ranging from $\epsilon = 1k_BT$ to $\epsilon = 500k_BT$, each with an equilibrium distance $r_{eq} = 1$nm. Each simulation was started with cytoplasmic dynein in the configuration shown in Figure 5.16, corresponding to the superposed configuration described by Imai *et al.* [8]. In this configuration, each monomer begins approximately at the VdW equilibrium position for the interaction. To maximise the search rate through the external fluid, we lowered the external viscosity of the system to a value $\mu = 1 \times 10^{-5}$Pa.s.

Using the central axis of the microtubule to define a cylindrical coordinate system, we calculated the diffusional properties of the right-hand MTBD, which was not pinned to the microtubule binding site, but was able to interact via VdW interactions. We firstly consider the radial diffusion, for which the resulting variance trace is shown in Figure 5.19.

We see that for many of the values of the VdW interaction strength there are two different regimes for the diffusion trace. For the first part of the simulation, the diffusional variance converges to a constant plateaus, becoming constant for a time. Following this, a large growth is observed in the trace. The $\epsilon = 3k_BT$ and $\epsilon = 5k_BT$ traces show this trend clearly, remaining constant at $\langle (r - r_0)^2 \rangle \approx 7.5\text{nm}^2$ and $\langle (r - r_0)^2 \rangle \approx 9.5\text{nm}^2$ respectively before experiencing a huge growth in variance at $t \approx 13\mu$s. Closer inspection of the two simulations show that this is the moment when the MTBDs overcame the local attraction to the microtubule and began to explore the full three-dimensional

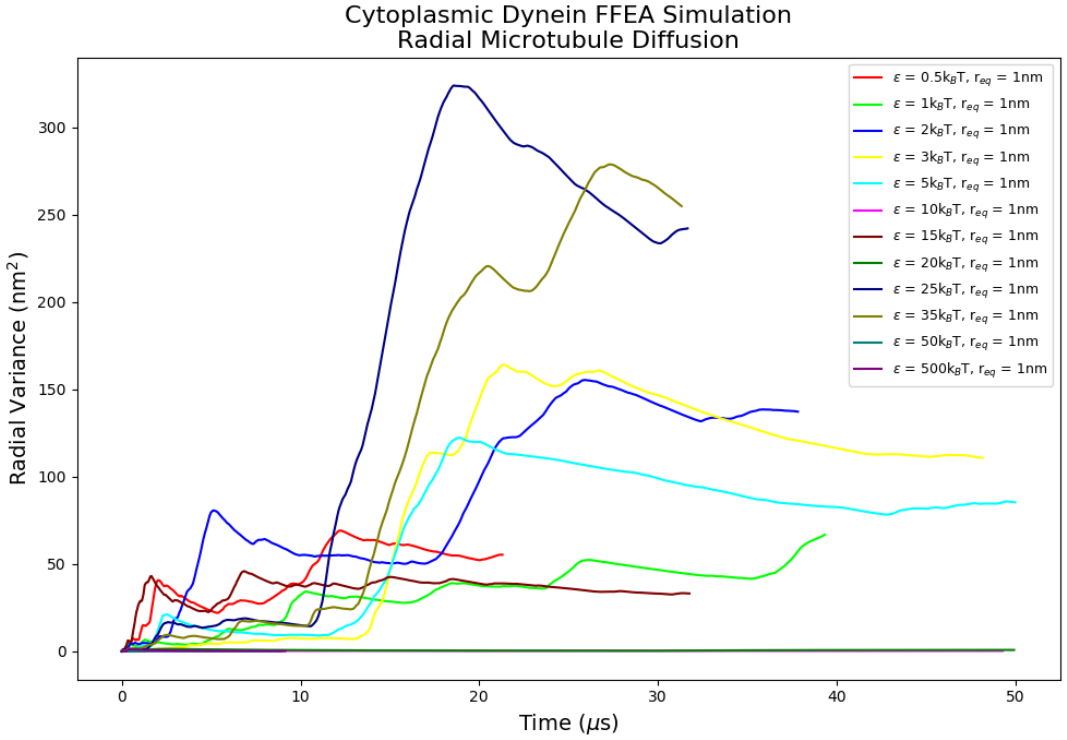


FIGURE 5.19: Radial variance of the free cytoplasmic dynein monomer under the action of a range of different Lennard-Jones potential strengths, each with the same equilibrium distance $r_{eq} = 1\text{nm}$.

conformational space. To emphasise the point more clearly, the same data has been plotted on logarithmic axes and only for the first 15μs in Figure 5.20.

From the strongest interaction potentials, we see something of a plateau forming with an approximately exponential decay. As we have seen in the previous flexibility calculations in this chapter, this represents constrained diffusion under the action of a linear restoring force, which is characteristic of a particle trapped in a harmonic potential. Beyond the steep increase in variance, if a plateau forms at all it is the constrained 3D diffusion limited only by the GST domain and flexibility of the monomer that remains bound to the microtubule.

The Lennard-Jones interaction potential cut-off within FFEA is at a distance of 3nm, and so any radial variance $\langle (r - r_0)^2 \rangle > 9\text{nm}^2$ represents effective dissociation from the microtubule under this potential. As we are not considering the free 3D diffusion of the monomer, but its diffusion along the microtubule, from this point forward we will only consider the simulations up to the point of their dissociation. We also disregard those simulations that dissociate too quickly to see any axial diffusion.

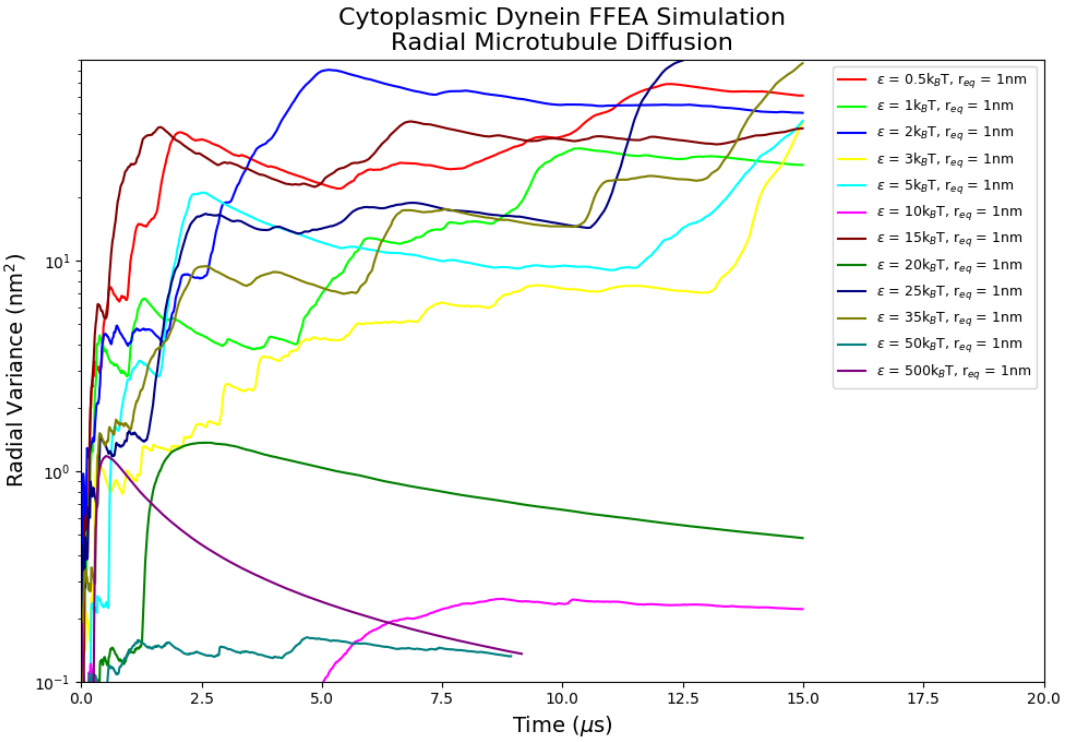


FIGURE 5.20: Radial variance of the free cytoplasmic dynein monomer under the action of a range of different Lennard-Jones potential strengths, each with the same equilibrium distance $r_{eq} = 1$nm, for the first 15μs of the total simulation.

Figure 5.21 shows the axial variance of each simulation, along the length of the microtubule, whilst the MTBD is still associated with the microtubule i.e. each graph stops as soon as the corresponding radial variance increases beyond the VdW cut-off distance. We notice the presence of a number of characteristic exponential decays forming with each curve, which implies a good approximation of a linearly constrained fluctuation in the axial direction before the dissociation occurs. We also see that higher potentials correspond approximately to lower axial variances. The diffusional behaviour of the MTBD changes significantly between the $\epsilon = 10k_BT$ simulation and the $\epsilon = 20k_BT$ simulations. As the interaction minimum decreases, so too does the local gradient of the Lennard-Jones potential well about the energy minimum, which means that the fluctuations about the local energy minima decrease in size. At some point between $\epsilon = 10k_BT$ and $\epsilon = 20k_BT$, the gradient is so large that thermal fluctuations, over the 15μs period, are insufficient to allow the MTBD to escape. A closer inspection of the respective simulations show that these local minima correspond to geometric artefacts at specific locations along the microtubule. The small perturbations in the $\epsilon = 10k_BT$ simulation are actually rotational variations within a local minimum allowing a small amount of increased axial motion, whereas in the $\epsilon = 20k_BT$ simulation even rotation is inhibited. However, the $\epsilon = 5k_BT$ simulation shows a continuously increasing variance

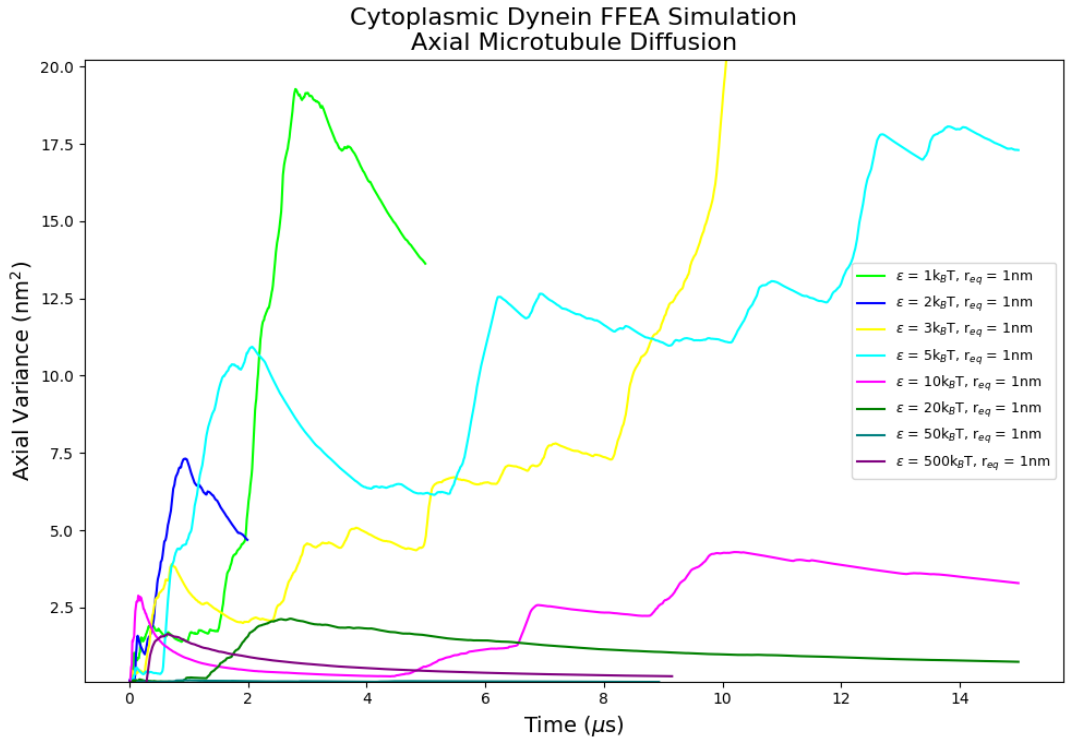


FIGURE 5.21: Axial variance of the free cytoplasmic dynein monomer under the action of a range of different Lennard-Jones potential strengths, each with the same equilibrium distance $r_{eq} = 1$nm.

with localised relaxations. It is likely within this range then, $5k_BT < \epsilon < 10k_BT$, that axial diffusion can occur whilst the MTBD remains bound to the microtubule. Assuming that the overall diffusion between the localised relaxations is approximately free diffusion, given the weakness of the GST restoring force, we attempted to characterise the diffusion numerically by performing linear fits to suitable portions of the data. These fits are shown in Figure 5.22. For the $\epsilon = 3k_BT$ and $\epsilon = 5k_BT$ simulations we obtain approximately the same diffusion constant, whereas for the $\epsilon = 10k_BT$ case we see a noticeable decrease in the diffusion constant even in the relatively small amount of time it was in motion. With an increasing interaction potential strength, then, we see that it is not simply radial diffusion that is affected. The complex geometry of the microtubule, which we were able to approximately capture in our initial meshing procedure, contains localised geometric minima which also inhibit axial diffusion with increasing interaction strength until eventually, axial diffusion is prevented altogether as can be seen clearly in the $\epsilon \geq 20k_BT$ traces in Figure 5.21.

We also find that none of our simulations, over the 15μs time scale, were able to diffuse far enough on average to reach the next available binding size before detaching from the microtubule. The distance between binding sites in the microtubule is 8nm, and

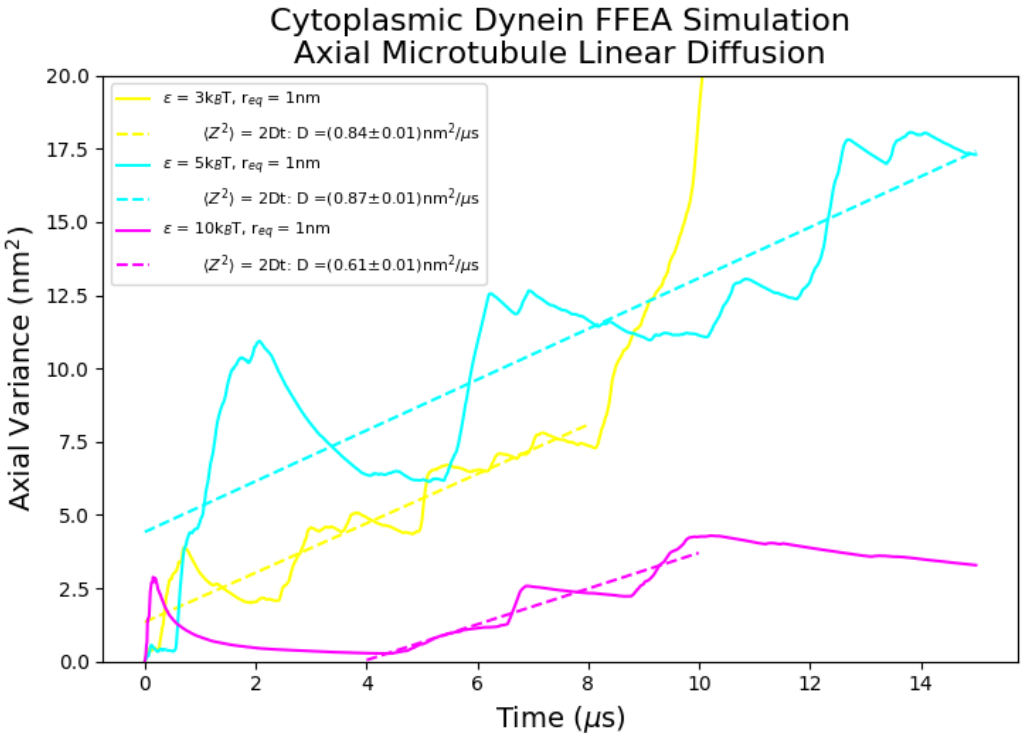


FIGURE 5.22: Axial variance of the free cytoplasmic dynein monomer under the action of a range of different Lennard-Jones potential strengths, each with the same equilibrium distance $r_{eq} = 1$nm, and their associated linear fits. Errors reported as the standard deviations of the least-squares fitted parameter.

so this would correspond approximately to an axial variance $\langle (Z - Z_0)^2 \sim 64\text{nm}^2$. We would expect our axial variance $\langle (Z - Z_0)^2 \propto t$ for an approximately free diffusion in the axial direction, and so with a maximum axial variance achieved over the course of 15μs, $\langle (Z - Z_0)^2 \approx 17.5\text{nm}^2$, we would expect that simulations of $\sim 60\mu$s are required to capture the full range of axial diffusion.

Further simulations performed to refine the specific required potential for surface diffusion were considered. However, whether this relatively simple and short-range functional form is itself sufficient to capture diffusional dynamics with respect to something as geometrically complex as the microtubule has also been called into question as a result of these simulations. Such short-range potentials could potentially have a problematic effect on the numerical integration procedure. As such, work is being pursued into other generalised functional forms which are less dependent upon the local mesh structure.

Chapter 6

Conclusions and Future Work

Although we have made significant progress in the development of FFEA over this project, the software has much room for improvement. What makes this an exciting project is that more than simply software optimisation remains. The underlying theoretical model, although currently self-consistent, can be expanded in a multitude of ways. Here we describe what the future direction of the software will likely be given the progress made in this thesis, and also what new systems can potentially be studied in the future.

6.1 Progress Overview

The theoretical development presented in this thesis has allowed FFEA to sample microsecond time scales with relative ease (compared to all-atom MD), taking us firmly into the mesoscale. Through development of the inertialess solution protocol we have seen that there are some biological systems, such as vesicles or small globular proteins, whose structurally isotropic nature makes their inertial modes of motion, those involving the velocity degrees of freedom, as important as the elastic dynamics. Yet for a biomolecule that is even slightly non-spherical, or for transport processes such as diffusion, the viscosity of the background material heavily overdamps the motion. This will likely have important consequences regarding the inclusion of external hydrodynamics to the theoretical model.

We note that the dynamical model does not converge to atomistic dynamics as the resolution of the mesh increases. Rather, FFEA converges instead to the continuum mechanical behaviour at the macroscale. Over long simulation times, the model explores the global conformational space available via continuous volumetric deformation,

the range of which we have seen is based more on the overall geometry than the intensive material parameters for many biomolecules. As the system size increases, the relative size of the thermal fluctuations naturally decrease, which effectively lowers the resolution of the simulation. These properties give FFEA a position at the upper limit of the coarse-grained simulation techniques introduced in Chapter 1, almost in parallel with DPD. However, note that the mapping procedure presented in Chapter 4 is not limited to transitions between two FFEA structures. The constant topology of an FFEA structure means that if an atomistic structure is available, we are able to map an FFEA trajectory back onto a higher resolution atomic structure. Although this process is an interpolation, and will therefore not introduce any additional higher resolution dynamical information, it does provide a stepping stone between the mesoscale and microscale. FFEA has been derived from fundamental equations of motion, and as such the constitutive models and mechanical stresses have real physical meaning with functional forms not specific to any particular simulation. With the work being done on continuum modelling of biological systems over the past decade [145] [146], FFEA has the potential to be key in developing a multi-scale framework between the mesoscale, macroscale and indeed the microscale as well.

The kinetic model was a first attempt at a direct link between the mesoscale and the macroscale. By using mesoscale mechanical processes to generate macroscopic observables, even in just a statistical sense, we have gained valuable insights into the interactions between these two length and time scales. The amounts of mechanical energy required to make substantial changes to the biological structures in comparision with the energy provided by ATP hydrolysis imply a highly specific pathway through the free energy environment for ATP hydrolysing proteins such as dynein. We found that in order to make the kinetic model feasible, both in a physical sense but also computationally, we are required to consider only the relevant degrees of freedom involved in the transition.

Our simulations of cytoplasmic dynein showed that the relevant flexibilities can indeed be modelled at the mesoscale, including the localisation of flexibility to a hinge region. However, the flexibility of the GST domain alone does not account for the prevelance of superposed states seen in experiments [8]. From this parametrisation, we saw the full cytoplasmic dynein system is completely free to diffuse in all three dimensions due to the flexibility of the GST domain. The inclusion of short range interaction potentials allowed us to associate the free MTBD with the microtubule and study the effects on its diffusional dynamics. We found that over the total 50μs simulation period, many interaction potentials were simply not strong enough to keep the monomer associated with the microtubule track. We also found that it was the shape of geometric artefacts

along the microtuble that enabled spatially localised energy minima to form. Whether there exists an interaction potential ϵ that gives the MTBD the freedom to move between these energy minima whilst staying associated to the microtubule is as yet unknown, but if it does exist then based on our current data we believe it will be on the range $5k_BT < \epsilon < 10k_BT$.

6.2 FFEA Development

6.2.1 Software Development

In parallel with the work done in this thesis, I have been part of a software development team working to make FFEA into a self-contained and user-friendly software package. We recently released the first public edition of the software [93], which contains the functionality of the FFEA dynamic model, linear elastic model, dynamic mode model and time scale calculator functionalities. These core simulation protocols are bundled with a fully integrated test suite which verifies the computational and physical stability of the installation. The tests presented in Chapter 3 form a key part of these physical tests, with minor modifications made so that they run faster for an standard user.

Also contained within the package are a set of software tools designed to help a user build the initial FFEA system ready for simulation. The majority of these tools, written in both C++ and Python, were developed by myself continuously throughout this research project. In preparation for the software release, I refactored all of these tools, making them dependent upon a core set of python modules designed to read, write and modify each of the FFEA file formats, and interact with any other required data structures in a consistent manner. Using these modules, we have developed a plugin for the PyMOL molecular visualiser [96]. The plugin enables a user to visualise FFEA objects alongside the electron density profiles / atomic structures they were built from, visualise simulation trajectories and modify the FFEA structures themselves through the PyMOL GUI. This toolkit makes interfacing with the software a much easier process. The software can be downloaded as a a pre-built binary or as source code from `https://bitbucket.org/FFEA/ffea/downloads/`. Keen users can link to the central repository through git version control: `https://bitbucket.org/FFEA/ffea.git`.

I envisage the future of the software to be as follows. For large proteins such as cytoplasmic dynein and others that have been simulated, hydrodynamic communication throughout the external solvent environment cannot be neglected. Indeed, axonomal

dynein works in coordination with many other identical monomers which are separated by $\sim$ 10nm [7]. This is close enough, given their size, that it is quite feasible that the monomers affect each other's motion through the external solvent velocity field. If we are to simulate systems like this, or any system with multiple macromolecules, then hydrodynamics cannot truly be assumed to be a local effect.

Hydrodynamics is inherently a long-range interaction, unlike the Lennard-Jones and steric interactions we have implemented up to this point. It will become vital, then, to develop an efficient parallelisation of the software to compensate for the increased number of calculations required per simulation timestep. Although the thermal noise is currently spatially delta-correllated throughout each simulated biomolecule due to the localisation of the viscous drag to each finite element, this may not be possible with external hydrodynamic coupling. The parallelisation will therefore need to go beyond the shared memory parallelisation we currently utilise, and into a full MPI implementation to compensate for the coupling.

Finally, the visualisation we currently use is simply a plugin to the PyMOL molecular graphics package. Although useful, it is a currently a reasonably inefficient interface, taking approximately 10minutes to load a standard FFEA trajectory. It may become necessary to write our own visualisation software that explicitly includes support for both continuum systems with associated mesh topologies, and atomistic systems with chemical bond topologies. This visualiser would ideally be linked to the various mesh processing tools as well as atomistic analysis packages, allowing a much more user-friendly interface for new users, a necessary requirement of any emerging technology.

6.2.2 FFEA Kinetics

The kinetic model developed in Chapter 4 has been shown to work in the general case. Through its development and application to test objects, we uncovered new biophysical questions regarding the free energy landscape of a biomolecular system. As we have repeatedly mentioned throughout this thesis, a Young's modulus $E \sim 1$GPa is widely regarded as representative of the biological mesoscale. Therefore, the required amount of energy calculated by FFEA to perform a kinetic transition from the $L = 5$nm cubic structure to the parallelohedron ($\sim 2000 k_B T$) is not an unrealistic value. Instead, it was the process itself that was unrealistic. We were able to compensate for the large energy difference by reducing the size of the cube to that of a single element, or a small number of degrees of freedom.

The only assumptions made in the core kinetic model, before it was integrated into FFEA, was that the 'chemical rates' r_{ij}^0 were completely independent of the position in phase space of the molecule, and the specific values of f_{ij}. Other than this, the model is exact with regards to equilibrium statistical mechanics. In future considerations of biological kinetics within FFEA, we must consider whether our mapping procedure is appropriate for consideration of only the relevant degrees of freedom which are affected by the conformational change. Also, to prevent the immediate reversal of any conformational change to a high mechanical energy state within FFEA, we may also need to consider the effect of energy dissipation over the course of a conformational change. These adaptations will likely prove necessary for the future applications of kinetic FFEA with regards to conformational change in real biological systems, such as cytoplasmic dynein.

Due to the specificity of chemical binding events on local atomistic structure as well as configurational entropy, parameterisation of specific binding is as yet an unsolved problem within FFEA. As such, we have also considered the inclusion of chemical binding as a kinetic event. This would require the consideration of an external work term ΔW into our energy modifications to the kinetic rates, but would allow us to study the effect on external forces on kinetic events, potentially enabling us to further elucidate the effect of mechanical gating in molecular motors.

6.3 FFEA Applications

Our simulations of cytoplasmic dynein generated several interesting results, but are only the beginning of a much larger project. In regards to the monomer, we saw that the geometry of the stalk domain, rather than its intrinsic flexibility, is responsible for homogeneous flexibility. Yet, when flexibility is localised atomistically due to a sequence breaking residue (proline), then the only way to replicate the flexibility within FFEA is to greatly reduce the intrinsic flexibility of the system by lowering the effective Young's modulus. When compared to the results obtained by Yun *et al.* [146], it seems that intrinsic flexibility in biomolecules is determined simply by the electron density (density of covalent bonds), whereas extrinsic flexibility is heavily dependent upon the shape of the object. From Chapter 3, we know that at the mesoscale the aspect ratio of an object determines which physical regime of motion, inertial or viscous, the system is in. This invites an interesting question in general: is there a relationship not just between structure and function, but also shape and function at the mesoscale? Studies into how high-resolution atomistic dynamics result in the mesoscopic behaviour which is often on the overall shape have a new avenue for analysis through combined

atomistic and FFEA simulations. Unlike many other mesoscale techniques, using the mapping procedure developed in Chapter 4 atomistic and FFEA simulations can be directly compared with one another, presenting many new opportunities for studying the interface between the two spatio-temporal regimes.

We saw that the amount of diffusional range a dynein monomer has when not bound to the microtubule is large enough to hint at the possibility of some form of interaction between the MTBD and the microtubule, as do recent simulations results by Li *et al.* [138]. Our simulations showed, qualitatively at least, that an increase in non-specific interaction potential strength between the MTBD and microtubule affects not only the radial diffusion, but the axial diffusion as well. How the specific local geometry of the binding sites and indeed, the spaces between protofilaments, affect the motion of the MTBD has not yet been studied in detail. The functional form of the Lennard-Jones interactions used, and its implementation in FFEA, may not be a suitable representation of non-specific interaction at the mesoscale, and so more work needs to be done on how the geometry and mesh topologies affect these interactions. At the conclusion of this project, new interaction potentials have been developed that are more weakly dependent upon the distance between the two interacting objects, and are being used to test these properties using simple sphere / cuboid systems.

Our original aim was to study how cytoplasmic dynein is able to walk at all, given only its structure. To accomplish this goal, we firstly need to perform further studies on the diffusional properties of both the post-powerstroke state, and indeed, the pre-powerstroke state too. To show that microtubule surface diffuson is possible within FFEA, given the current functional form of non-specific interaction, specification of an appropriate range of interaction parameters for microtubule would require a further simulation series of interaction potentials on the range $5k_BT < \epsilon < 10k_BT$ with $\sim 1k_BT$ intervals. If this system proved incapable of performing surface diffusion, then the next stage would be to consider a different energy minimum position, r_{eq}, between the MTBD and the microtubule. It may be the case that the proximity to the microtubule itself may allow the complex geometric structure a larger effect on the dynamics of interacting objects, which 'smooths out' at larger distances.

Whether surface diffusion is possible or not, the kinetic transition between the post and pre powerstroke states will likely have a significant effect on a dynein monomer finding the next binding site. Simulations including the kinetic model described in Chapter 4 can be performed to enhance the simple diffusional studies described above. It may be the case that the reverse-powerstroke of dynein acts to reduce the need for undirected diffusion by placing the MTBD much closer to the target binding site. Whether this

is the case or not will tell us whether or not undirected diffusion is a limiting step for the entire kinetic cycle. These simulations can be performed using just two kinetic transitions within a dynamic simulation (post-powerstoke to pre-powerstroke transition and vice versa) and analysing the probability distribution of the MTBD position with respect to the microtuble. Including some form of specific binding procedure, where the detachment rate is affected by the mechanical strain in the object, is the clear extension to this, and how the experimentally observed binding events affect long-time probability distributions of position along the microtubule is vital information in determining the energetics of translational motors. From here, we would have insights into which stages of the kinetic cycle are necessary for motion, and which are simply efficiency optimisations.

With an understanding of the above processes, we will have systematically built up to the inclusion of the entire set of kinetic transitions into the cytoplasmic dynein system. With insights from the work of Sarlah *et al.* [132], we will then be in a position to perform either a single step or, if time allows[1], multiple steps of the cytoplasmic dynein motor. Comparison with recent kinetic models, again from Sarlah *et al.* [147], this will allow us to understand how the mechanical structure of the motor truly affects the kinetic cycle, and study the effect of the kinetic cycle on emergent properties of the system, such as the stalling force, run-length and perhaps most importantly, the force that dynein can exert on attached cargo. A physically informed understanding of how molecular motors function will hopefully provide the necessary knowledge to aid the community in building our own motor systems at the mesoscale, expanding on work already performed [148] to help with real-world problems such as drug delivery, health enhancements, and so forth.

6.4 Final Remarks

Much work remains to be done on the FFEA software package with regards to functionality and usability. The work presented in this thesis, in my opinion, represents a vital step into the theoretical understanding of how biological system operate at the upper reaches of the mesoscale. The associated software release verifies the dynamic model, with the kinetic model next in line for publication and release. I hope that the software is of use to biologists, chemists and physicists alike in the future as we search for a robust and consistent mathematical and computational framework to describe the biological mesoscale, and indeed, multi-scale biological systems in general.

[1] By which we mean computational runtime!

My final remark will be a much bolder claim for structural biology and bioinformatics in general. Although our overarching goal in this thesis has been to use FFEA for predictive purposes, the sheer complexity of the dynein system has shown us that continuous experimental input is needed for this type of simulation. Peters *et al.* recently argued, for the field of chemistry, that such a strengthening of collaboration is necessary for the optimisation of research [149]. We can imagine worst-case scenarios, that theoreticians, without experimental input, may specify their mathematical generalisations to incorrect real-world conditions, and experimentalists, without theoretical input, may unnecessarily perform experiments which look for biophysical behaviour that cannot exist. In our case, the mesoscopic regime is so complex in the biophysics it presents, overlapping the mathematically distinct discrete (atomistic) and continuum (solid/fluid) mechanics, that 'minor' approximations and errors in theoretical models can have far reaching consequences.

We saw this problem emerge in our study of dynein, where we realised that a homogeneous parametrisation of the dynein stalk / linker, in effect treating the system using Euler beam theory, does not accurately describe the total range of available motion to the object. It may be that even inhomogeneous parametrisation is insufficient, and that these discrepancies are truly atomistic properties. Although we were able to make use of this parametrisation by considering only the angular fluctuations, it may be the case that length fluctuations are also important to the system in some way, a question which could be much more readily studied via experimentation. On the other hand, in our main citation for the simulation studies, we note that the measurements of Imai *et al.* [8] all utilised the binding of dynein to the microtubule in order to align images and extract structural data. This method clearly prevents the experimental study of dynamical and kinetic behaviour in the absence of the microtubule, yet this is something that can be accomplished in a simulation environment simply by removing the microtubule object. Simulation and theoretical calculations give us full control over our models, from which we can observe interesting behaviour that cannot be seen within a less-controlled experimental system. Yet those very same experiments must be performed to inform the model in the first place. A combination of the two methods through continuous collaboration, with simulation and theory informing experimental decisions and vice versa, is necessary to properly understand what truly drives biology at this scale, and to move biophysics as a whole into the mesoscale.

Appendix A

FFEA Kinetics and Activation Energy

In Chapter 4, Section 4.3.5 we postulated the existence of the values f_{ij} and f_{ji} without reference to what they physically represent. These terms exist with the FFEA kinetic framework for all possible biological processes that can be modelled kinetically, i.e. a process for which we know the initial and final structural states but have no knowledge of the underlying energy landscape. Equation 4.21 shows that for detailed balance to be preserved at equilibrium, f_{ij} is actually the only free parameter, with f_{ji} constrained by,

$$f_{ij} = 1 - f_{ji}. \tag{A.1}$$

This Appendix will look at the physical meaning of this value.

The Arrhenius equation for reaction rates provides us with an empirical relationship between the activation energy and the kinetic rate,

$$R = A \exp\left(\frac{-E_A}{k_B T}\right), \tag{A.2}$$

where R is an experimentally measured kinetic transition rate and E_A is the activation energy for the transition, with the pre-factor A being an unknown constant. In Chapter 4, we defined these measured rates in terms of free energies as opposed to standard energies, and so we rewrite Equation A.2 for a transition from mesostate i to mesostate j within a kinetic network as,

$$R_{ij} = A \exp\left(\frac{-\Delta F_{A,ij}}{k_B T}\right), \tag{A.3}$$

where $\Delta F_{A,ij} = F_{B,ij} - F_i$ and $F_{B,ij}$ is the absolute free energy barrier between mesostate i and j, from which it follows $F_{B,ij} = F_{B,ji}$ (see Figure 4.3c). This definition then gives,

$$\begin{aligned}\frac{R_{ij}}{R_{ji}} &= \exp\left(\frac{F_i - F_{B,ij} - F_j + F_{B,ji}}{k_B T}\right), \\ &= \exp\left(\frac{-\Delta F_{ij}}{k_B T}\right).\end{aligned} \tag{A.4}$$

which is identical to Equation 4.13, corresponding to the mesoscale detailed balance conditions.

As we are mainly working with microstates in our study of kinetic processes, we assume an equivalent relation for the microstate rates $r(\vec{x},\vec{p})_{ij}$,

$$r(\vec{x},\vec{p})_{ij} = a\exp\left(\frac{-\Delta E(\vec{x},\vec{p})_{A,ij}}{k_B T}\right), \tag{A.5}$$

where a is a new, unknown pre-factor and $\Delta E(\vec{x},\vec{p})_{A,ij} = E(\vec{x},\vec{p})_{B,ij} - E(\vec{x},\vec{p})_i$ is the activation energy in each microstate that contributes to the overall free energy of activation. $E(\vec{x},\vec{p})_{B,ij}$ is then an absolute energy barrier that must be overcome to begin a kinetic transition. Just as in Chapter 4, we can decompose this into chemical and non-chemical components such that $\Delta E(\vec{x},\vec{p})_{A,ij} = \Delta E(\vec{x},\vec{p})^c_{A,ij} + \Delta E(\vec{x},\vec{p})^{nc}_{A,ij}$, where we assume the chemical contributions to the energy to be independent of the location in phase space. Equation 4.20 then gives us an explicit form for $r(\vec{x},\vec{p})_{ij}$ within the FFEA kinetic framework, so we can substitute into Equation A.5 to obtain,

$$r^0_{ij}\exp\left(\frac{-f_{ij}\Delta E^{nc}_{ij}}{k_B T}\right) = a\exp\left(\frac{-\Delta E^c_{A,ij}}{k_B T}\right)\exp\left(\frac{-\Delta E^{nc}_{A,ij}}{k_B T}\right). \tag{A.6}$$

Our definition of r^0_{ij} was that it is comprised of all contributions to the overall rate $r(\vec{x},\vec{p})_{ij}$ not dependent upon the location in phase space. Therefore, from Equation A.6 we postulate that r^0_{ij} on the LHS is comprised of all of the chemical terms on the RHS, including the chemical component of the activation energy,

$$r^0_{ij} = a\exp\left(\frac{-\Delta E^c_{A,ij}}{k_B T}\right), \tag{A.7}$$

which allows us to cancel these from both sides of Equation A.6,

$$\exp\left(\frac{-f_{ij}\Delta E^{nc}_{ij}}{k_B T}\right) = \exp\left(\frac{-\Delta E^{nc}_{A,ij}}{k_B T}\right). \tag{A.8}$$

Finally, taking logarithms of both sides of Equation A.8 and making f_{ij} the subject gives,

$$f_{ij} = \frac{E^{nc}_{B,ij} - E^{nc}_{i}}{E^{nc}_{j} - E^{nc}_{i}}, \tag{A.9}$$

where $E^{nc}_{B,ij}$ is the non-chemical component of the activation energy barrier. Equivalently for the reverse process,

$$f_{ji} = \frac{E^{nc}_{B,ij} - E^{nc}_{j}}{E^{nc}_{i} - E^{nc}_{j}}. \tag{A.10}$$

With the chemical component of the activation energy being included in the definition of r^0_{ij}, Equations A.9 and A.10 show that the values f_{ij} and f_{ji} are proportional to the sizes of the relative mechanical energy barriers, $\Delta E^{nc}_{A,ij}$ and $\Delta E^{nc}_{A,ji}$, with respect to the total mechanical energy change between states ΔE^{nc}_{ij}. If these values are constrained to the range $0 < f_{ij} < 1$ as was assumed in Chapter 4, Section 4.4, then it also follows that $E^{nc}_{i} < E^{nc}_{B,ij} < E^{nc}_{j}$ i.e. the mechanical energy barrier is somewhere between the initial and final energy states, which makes intuitive sense for a process which has a continuous mechanical pathway from mesostate i to mesostate j. We might assume that the energy barrier for a given transition is in fact dominated by the chemical energy barrier[1] and that the mechanical energy contribution to the barrier is a relatively small perturbation. In such a case, the activation barrier will be at some point along a putative reaction co-ordinate between the mesostates i and j. If we further assume that the effective non-chemical energy varies smoothly along that reaction co-ordinate, the value of f_{ij} we use will depend on the location of the energy barrier along that reaction co-ordinate. For example, if the energy barrier is about half-way along the reaction co-ordinate, then we might expect the change in non-chemical energy from state i to the barrier $(E^{nc}_{B,ij} - E^{nc}_{i})$ to be around half the change in non-chemical energy from state i to state j $(E^{nc}_{j} - E^{nc}_{i})$ and so $f_{ij} \approx 0.5$. On the other hand, if the energy barrier is for some reason close to state j, for example in a sharp chemical binding event, then $E^{nc}_{B,ij} - E^{nc}_{i}$ will be almost the same as $E^{nc}_{j} - E^{nc}_{i}$ and $f_{ij} \approx 1$. The physical reason for this would be that the system needs to undertake almost all the mechanical energy change before it arrives at the chemical energy barrier.

However, there is no physical reason that the mechanical energy barrier is confined to this range, which has further implications for the value of f_{ij}.

The kinetics for the 1D dumbbell model derived in Section 4.4 show there to be fundamental limits on the range of f_{ij}. The true domain of f_{ij} can be found by analysing

[1] We would perhaps expect this for a process known to be dependent upon a chemical process such as ATP hydrolysis

the following term from Equation 4.42,

$$R_{\alpha\beta} \propto \sqrt{\frac{1}{f_{ij}\left(\frac{k_j}{k_i}-1\right)+1}}. \tag{A.11}$$

The presence of the square-root introduces a limit on its argument, such that for any transition from mesostate i to mesostate j to physically exist, we require,

$$f_{ij}\left(\frac{k_j}{k_i}-1\right) > -1. \tag{A.12}$$

This inequality is discontinuous depending on the relative sizes of the two spring constants,

$$f_{ij}: \begin{cases} f_{ij} < \frac{1}{1-\frac{k_j}{k_i}}, & k_j < k_i \\ f_{ij} > \frac{1}{1-\frac{k_j}{k_i}}, & k_j > k_i \\ f_{ij} \text{ undefinied}, & k_j = k_i \end{cases}$$

This piecewise inequality keeps the pre-factor to x'^2 in Equation 4.41 negative, so that the integral over the Boltzmann distribution exists and is non-infinite. These conditions allow the activation energy barrier to be any size within the FFEA kinetic framework, and not restricted to the range $E_i^{nc} < \Delta E_{B,ij} < E_j^{nc}$. The activation energy may be arbitrarily large compared with the final difference between mechanical energy states. For the specific case of $k_j = k_i$, Equation A.12 shows us that f_{ij} can take any value and is thus undefined.

As a final note, notice that the only mathematical restriction on f_{ij} is in Equation A.1, required for detailed balance to hold. In general then, f_{ij} and f_{ji} are not constants as they are considered in this work, but can be generalised as functionals of the activation energy barrier $E_{B,ij}^{nc}$ in this mathematical framework.

Appendix B

Numerical Stability within the Kinetic Framework

To generate our test case results in Section 4.4.4 we used arbitrarily defined sets of rate constants to show the effects of the kinetic scheme on the underlying dynamics. Equipartition of energy was adhered to even as the instantaneous state changes provided large amounts of energy to the system. However, if the rates are too high and the energy is introduced too quickly, this is the equivalent to standard numerical instability within a purely dynamical simulation. Here we include a derivation of the numerical instability conditions for the one-dimensional bead spring model from Section 4.4, and show how they can be applied to the much more complicated systems involved in FFEA simulations.

We begin again with our equation of motion for the bead separation in the bead-spring system,

$$\lambda_r \frac{dy}{dt} + ky = \frac{\lambda_r}{\lambda_2} N_2 - \frac{\lambda_r}{\lambda_1} N_1, \tag{B.1}$$

where $y = x' - l$ is the deviation from the energy minimum. Equation B.1 is a first order ODE in y and can therefore be solved exactly using the integrating factor method. However, as we are interested in numerical stability, we will solve numerically using a single-step Euler approach. Expanding the partial derivative $\frac{\partial y}{\partial t} \approx \frac{\Delta y}{\Delta t}$, we get the following form for a single integration step,

$$y(t + \Delta t) = y(t) \left(1 - \frac{\Delta t}{\tau}\right) + \Delta t \left(\frac{N_2(t)}{\lambda_2} - \frac{N_1(t)}{\lambda_1}\right), \tag{B.2}$$

where $\tau = \lambda_r/k$. From here, we can determine the result of a series of N applied integration steps from an initial extension $y(0)$,

$$\begin{aligned} y(t_N) &= y(0)\left(1-\frac{\Delta t}{\tau}\right)^N, \\ &+ \Delta t \sum_{n=0}^{N-1}\left(1-\frac{\Delta t}{\tau}\right)^n\left(\frac{N_2(t_{N-(1+n)})}{\lambda_2}-\frac{N_1(t_{N-(1+n)})}{\lambda_1}\right), \end{aligned} \tag{B.3}$$

where $t_i = i\Delta t$. Here we see our first numerical stability condition, required to keep the first term of Equation B.3 and many of the summation terms from diverging or oscillating,

$$\begin{aligned} 0 < 1-\frac{\Delta t}{\tau} \leq 1, \\ \Rightarrow 0 \leq \Delta t < \tau. \end{aligned} \tag{B.4}$$

Equation B.4 is a standard condition and common to all numerical integration techniques. From here we can immediately calculate the expectation value of our spring extension,

$$\langle y(t_N)\rangle = y(0)\left(1-\frac{\Delta t}{\tau}\right)^N, \tag{B.5}$$

as $\langle N_1\rangle = \langle N_2\rangle = 0$ [21]. At $N = 0$ we see that $\langle y(t_0)\rangle = y(0)$ as expected. Additionally, as $N \to \infty$, $\langle y(t_N)\rangle \to 0$ showing that as time progresses, the initial extension no longer affects position expectation.

We now calculate the variance in the extension,

$$\begin{aligned} \langle y(t_N)y(t_M)\rangle &= y(0)^2\left(1-\frac{\Delta t}{\tau}\right)^{2N}, \\ &+ \Delta t^2 \sum_{n=0}^{N-1}\sum_{m=0}^{N-1}\left(1-\frac{\Delta t}{\tau}\right)^{n+m}\left(\frac{\langle N_2(t_i)N_2(t_j)\rangle}{\lambda_2^2}+\frac{\langle N_1(t_i)N_1(t_j)\rangle}{\lambda_1^2}\right), \end{aligned} \tag{B.6}$$

where $i = N-(1+n)$, $j = N-(1+m)$ and the cross terms $\langle N_2(t_i)N_1(t_j)\rangle =$

$\langle N_1(t_i)N_2(t_j)\rangle = 0$ due to the spatial delta-correlation of stochastic noise when applied to a system modelling Brownian motion in this fashion. The noise is also delta-correlated in time [21] as follows,

$$\langle N_1(t_i)N_1(t_j)\rangle = \frac{2k_BT\lambda_1}{\Delta t}\delta_{ij},$$
$$\langle N_2(t_i)N_2(t_j)\rangle = \frac{2k_BT\lambda_2}{\Delta t}\delta_{ij},$$

and so we can substitute these into Equation B.6,

$$\langle y(t_N)^2\rangle = y(0)^2\left(1-\frac{\Delta t}{\tau}\right)^{2N} + \Delta t\frac{2k_BT}{\lambda_r}\sum_{n=0}^{N-1}\left(1-\frac{\Delta t}{\tau}\right)^{2n}. \tag{B.7}$$

We can rewrite both power terms in Equation B.7 under the assumption that $\Delta t << \tau$,

$$\langle y(t_N)^2\rangle = y(0)^2\left(1-2\frac{\Delta t}{\tau}\right)^{N} + \Delta t\frac{2k_BT}{\lambda_r}\sum_{n=0}^{N-1}\left(1-2\frac{\Delta t}{\tau}\right)^{n}. \tag{B.8}$$

Finally, solving for the geometric series and rearranging to a more useful form gives,

$$\langle y(t_N)^2\rangle = \frac{k_BT}{k} + \left(y(0)^2 - \frac{k_BT}{k}\right)\left(1-2\frac{\Delta t}{\tau}\right)^{N}. \tag{B.9}$$

Again, we look at the limiting cases. At $n = 0$, $\langle y(t_0)^2\rangle = y(0)^2$, again showing the expected behaviour. For $n \to \infty$, we find that $\langle y(t_N)^2\rangle \to \frac{k_BT}{k}$, which is the expected case of constrained diffusion for the extension of a spring.

Now we understand the equilibration process, we are in a position to understand the other numerical stability condition arising from the kinetic scheme. Imagine a coupled kinetic simulation where, at some time after equilibration the system makes a kinetic transition from state α, defined by k_α and l_α, to state β, defined by k_β and l_β,. We choose to define this time as $t = 0$, and because it is post equilibration we can make the assumption that the spring extension is the expected value of the original state. As such, immediately following the transition $y(0) = l_\alpha - l_\beta$. We can write an equation for the expectation value of the new energy as a function of time, $\langle U(t_N)\rangle = 0.5k_\beta\langle y(t_N)^2\rangle$,

$$\langle U(t_N)\rangle = \frac{1}{2}k_BT + \frac{1}{2}\left(k_\beta(l_\alpha - l_\beta)^2 - k_BT\right)\left(1-2\frac{\Delta t}{\tau_\beta}\right)^{n}, \tag{B.10}$$

where $\tau\beta$ is the time constant associated explicitly with the current kinetic state β. From here, we define the fractional error in $\langle U_(t_N)\rangle$, $e(t_N)$, as,

$$\begin{aligned} e(t_N) &= \frac{\left|\langle U(t_N)\rangle - \frac{1}{2}k_BT\right|}{\frac{1}{2}k_BT}, \\ &= \left|\frac{k_\beta}{k_BT}(l_\alpha - l_\beta)^2 - 1\right|\left(1 - 2\frac{\Delta t}{\tau_\beta}\right)^N, \\ \Rightarrow e(t_N) &= e(0)\left(1 - 2\frac{\Delta t}{\tau_\beta}\right)^N. \end{aligned} \tag{B.11}$$

Rearranging Equation B.11 for N will give us the expected number of time steps required to reach equilibration within a user defined error bound of $e(N)$,

$$N \approx \frac{\ln\left(e(N)\right) - \ln\left(e(0)\right)}{\ln\left(1 - 2\frac{\Delta t}{\tau_\beta}\right)}. \tag{B.12}$$

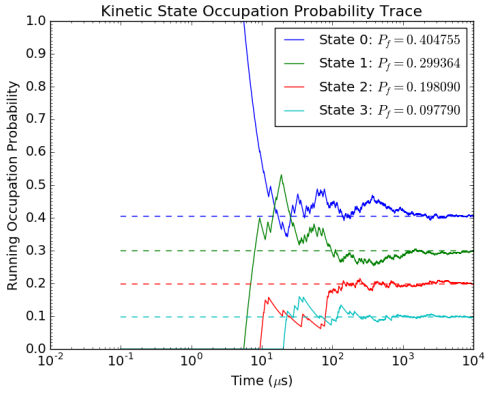
The denominator of Equation B.12 has a recognisable form; that of the first numerical stability condition from Equation B.4. This time however, the condition is more stringent such that for the logarithm to exist,

$$\begin{aligned} 1 - 2\frac{\Delta t}{\tau_\beta} &> 0, \\ \Rightarrow \Delta t &< \frac{1}{2}\tau_\beta \;\forall\; \beta. \end{aligned} \tag{B.13}$$

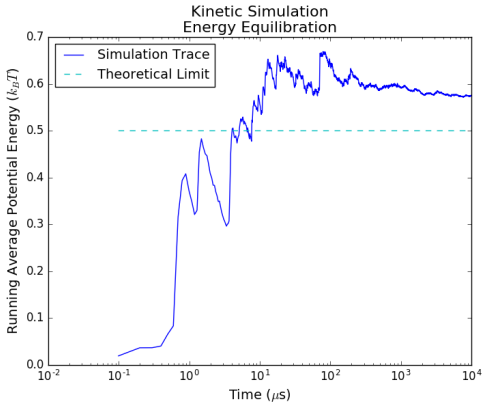
Although Equation B.4 shows that the integration procedure will be numerically stable if $\Delta t < \tau_\beta$, for the physical measurements to be accurate we require the stronger condition of Equation B.13.

For any value of error, $0 < e(N) < \infty$ we define, our integration scheme will be able to equilibrate within a finite number of time steps following a state transition, as if $e(N) - e(0) > 0$ we get a number of time steps required for equilibration. For the cases where $e(N) - e(0) \leq 0$, we the system has already equilibrated to our required accuracy level. Finally, we see that defining $e(N) = 0$ gives an equilibration time of infinite time steps, correctly showing that perfect accuracy is impossible.

Let us define an equilibration time, $t^{eq}_{\alpha\beta}$, as the equilibration time following a transition from state α to state β. This can be written simply as $t^{eq}_{\alpha\beta} = N\Delta t$, with N calculated from Equation B.12. However, if we choose our kinetic transition rates to be so quick that we do not leave the system time to re-equilibrate in between subsequent state transitions, then over the course of a simulation our system will never converge to equipartition within our defined error bounds. More formally, our final stability



(A) Running averages of kinetic state occupation probabilities.



(B) Equilibration of potential energy.

FIGURE B.1: A numerically inaccurate kinetic system (1D bead-spring model) in which the rates are so fast that the system cannot reach mechanical equilibrium between mesostate transitions. We see that mesoscopic detailed balance is adhered to, but equipartition is not, showing this to be a non-equilibrium system.

condition is,

$$r_{\beta\gamma} < \frac{1}{t^{eq}_{\alpha\beta}} \; \forall \; \alpha, \gamma, \tag{B.14}$$

where $r_{\beta\gamma}$ is the transition rate from state β to some other state γ. In words, this means that following a transition from state α into state β, all transition rates to further states γ must be slow enough to allow the system to equilibrate before the transition occurs.

This final condition is more an accuracy condition than a stability condition, as the numerical integration procedure would not diverge if the condition was not met. However, Equation B.14 must be statisfied for numerical and physical accuracy. An example of a system with a set of rates that are too high is shown reaching equilibration in Figure B.1, where we can see that the occupation probabilities have equilibrated to the correct values for the set of rates shown in Table 4.2 but the system has not equilibrated to $\langle U \rangle \neq \frac{1}{2} k_B T$, instead equilibrating to a slightly higher value. Within this simulation, the dynamics are stable but equipartition is not being adhered to. In the biological systems we have been looking at, rates are often slow enough such that we do not have to worry about non-equilibrium effects when performing these types of calculations [132]. However, in simulations where we may wish to accelerate these rates to save on runtime, we must take these conditions into account.

Appendix C

Flexibility Convergence

It is often the case with biomolecular simulation, and especially molecular dynamics, that the amount of conformational space sampled over the course of a simulation is insufficient to determine the true positional variance, and therefore the real flexibility, of the molecule in question. A recent paper brought light to this issue with evidence that certain dynamical modes within proteins have self-similar behavior over 13 orders of magnitude, with relaxation times of $10^{-13}s$ and $\sim 10^{2}s$ existing within the same molecules [150]. However, the way in which the variance evolves throughout a simulation can provide a small amount of detail in itself.

As we saw in Chapter 3, we can extract any over-damped degree of freedom we wish and, in an appropriate coordinate system, define a 1D equation of motion for that specific coordinate,

$$\lambda \frac{\partial x}{\partial t} + k(x - \bar{x}) = f(t), \tag{C.1}$$

where λ is a viscosity component, k is a spring constant and $f(t)$ is the stochastic force vector applied to the coordinate x, which has an average value $\bar{x}$. Equation C.1 can be partially solved using the integrating factor method. Multiplying through by a factor $\exp\left(\frac{t}{\tau}\right)$, where $\tau = \lambda/k$ gives,

$$\frac{\partial x}{\partial t} e^{\frac{t}{\tau}} + \frac{1}{\tau} e^{\frac{t}{\tau}} (x - \bar{x}) = \frac{1}{\lambda} f(t) e^{\frac{t}{\tau}}. \tag{C.2}$$

The LHS of Equation C.2 is the result of the product rule, and so can be rewritten as,

$$\frac{\partial}{\partial t} \left((x - \bar{x}) e^{\frac{t}{\tau}} \right) = \frac{1}{\lambda} f(t) e^{\frac{t}{\tau}}, \tag{C.3}$$

which can then be integrated over the length of the simulation, t, resulting in,

$$x(t) - \bar{x} = (x_0 - \bar{x})e^{-\frac{t}{\tau}} + \frac{1}{\lambda}\int_0^t f(t')e^{\frac{t'-t}{\tau}}dt'. \tag{C.4}$$

Equation C.4 contains the limiting properties that we need. At $t = 0$, $x = x_0$ and as $t \to \infty$, the term involving x_0 tends to 0, leaving a pure stochastic process.

We can use Equation C.4 to calculate the expected statistical moments of a simulation utilising this equation of motion by taking ensemble averages of the simulation at any time t. For the first central moment,

$$\begin{aligned}\langle x(t) - \bar{x}\rangle &= (x_0 - \bar{x})e^{-\frac{t}{\tau}} + \frac{1}{\lambda}\int_0^t \langle f(t')\rangle e^{\frac{t'-t}{\tau}}dt', \\ &= (x_0 - \bar{x})e^{-\frac{t}{\tau}},\end{aligned} \tag{C.5}$$

where $\langle f(t)\rangle = 0$ is defined for the RNG contained within FFEA. Equation C.5 shows that at $t = 0$, $\langle x\rangle = x_0$ and the expected value of x decays towards the mean as $t \to \infty$.

The second moment, the variance at any time t is calculated as,

$$\langle (x(t) - \bar{x})^2\rangle = (x_0 - \bar{x})^2 e^{-\frac{2t}{\tau}} + \frac{1}{\lambda^2}\int_0^t\int_0^t \langle f(t')f(t'')\rangle e^{\frac{t''+t'-2t}{\tau}}dt'dt'', \tag{C.6}$$

where the cross multiplied terms equal zero as $\langle f(t)\rangle = 0$. We can again implement the statistical properties defined within FFEA, where the fluctuation-dissipation theorem is used to give $\langle f(t')f(t'')\rangle = 2k_BT\lambda\delta(t' - t'')$ within the RNG [71], so,

$$\langle (x(t) - \bar{x})^2\rangle = (x_0 - \bar{x})^2 e^{-\frac{2t}{\tau}} + 2\frac{k_BT}{\lambda}\int_0^t\int_0^t e^{\frac{t''+t'-2t}{\tau}}\delta(t' - t'')dt'dt''. \tag{C.7}$$

The delta function reduces the number of integrals in Equation C.7,

$$\langle (x(t) - \bar{x})^2\rangle = (x_0 - \bar{x})^2 e^{-\frac{2t}{\tau}} + 2\frac{k_BT}{\lambda}\int_0^t e^{2\frac{t'-t}{\tau}}dt', \tag{C.8}$$

which can then be solved exactly,

$$\langle (x(t) - \bar{x})^2\rangle = \frac{k_BT}{k} + \left((x_0 - \bar{x})^2 - \frac{k_BT}{k}\right)e^{-\frac{2t}{\tau}}. \tag{C.9}$$

Equation C.9 shows that the variance calculated as an ensemble average at any time t should decay from the initial value of $(x_0 - \bar{x})^2$ towards the final value k_BT/k as $t \to \infty$, which is based solely on the true stiffness k. Using this, we can determine the

expected energy at any time, $\langle U \rangle = \frac{1}{2}k\langle (x(t) - \bar{x})^2 \rangle$,

$$\frac{1}{2}k\langle (x(t) - \bar{x})^2 \rangle = \frac{1}{2}k_B T + \left(\frac{1}{2}k(x_0 - \bar{x})^2 - \frac{1}{2}k_B T \right) e^{-\frac{2t}{\tau}}. \tag{C.10}$$

Now as $t \to \infty$, the dependence on both the starting state and the equilibrium state again decays away, leaving only the theoretical value $\langle U \rangle_\infty = \frac{1}{2}k_B T$ as predicted by the equipartition theorem.

We see that equipartition statistics only hold as $t \to \infty$, or more accurately for $t \gg \tau$, the time constant of this degree of freedom. In many studies however, we see the equipartition theorem applied to systems for which $t \sim \tau$ or even worse, $t < \tau$. Let us define a new spring constant, k_t, defined as the spring constant with the equipartition theorem applied at time t,

$$k_t = \frac{k_B T}{\langle (x(t) - \bar{x})^2 \rangle}. \tag{C.11}$$

We can substitute Equation C.9 into Equation C.11 to obtain,

$$k_t = k \frac{1}{1 + \left(\frac{k(x_0 - \bar{x})^2}{k_B T} - 1 \right) e^{-\frac{2t}{\tau}}}, \tag{C.12}$$

Equation C.12 shows that at $t = 0$,

$$k_t = k_0 = \frac{k_B T}{(x_0 - \bar{x})^2}, \tag{C.13}$$

which is an elasticity parameter based solely upon the initial extension. At $t \gg \tau$ however, as with the variance, the term involving both this initial spring constant decays away, tending towards the true spring constant k when $t \gg \tau$. Substitution of Equation C.13 into Equation C.12 gives,

$$k_t = k \frac{1}{1 + \left(\frac{k}{k_0} - 1 \right) e^{-\frac{2t}{\tau}}}. \tag{C.14}$$

Equations C.9 and C.14 are both forms of exponential decay which we can fit to the variance and stiffness trajectories of a simulation. If we assume that the system tends towards ergodicity, then values of k_t or $\langle (x(t) - \bar{x})^2 \rangle$ can be measured from any single simulation. Then, by either approximating the value of λ based upon the geometry of the object in question, or leaving τ as a free parameter, we can least-squares fit an equation of the form of Equation C.9 or Equation C.14 to extrapolate our simulation data and calculate a better approximation of the elasticity of the selected coordinate.

It should be noted that we have explicitly used $\bar{x}$ and not μ, the true mean, to represent

the mean value as the entire reason for this analysis is that we do not have enough trajectory data to know the final value of the mean. As such, the most appropriate implementation of this extrapolation is to calculate the variance at each time t as variations away from the sample mean of the entire available trajectory.

Bibliography

[1] Radostin Danev and Wolfgang Baumeister. Cryo-em single particle analysis with the volta phase plate. *Elife*, 5:e13046, 2016. PDB ID:5NI1.

[2] Wikimedia Commons. Redbloodcells, 2015. URL https://commons.wikimedia.org/wiki/File:Redbloodcells.jpg.

[3] Wikimedia Commons. Drops of blood., 2009. URL https://commons.wikimedia.org/wiki/File:NIK_3232-Drops_of_blood_medium.JPG.

[4] Blender Foundation. Blender, 2017. https://www.blender.org/.

[5] J. Schöberl. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and visualization in science*, 1(1):41–52, 1997.

[6] Ronald D Vale. The molecular motor toolbox for intracellular transport. *Cell*, 112(4):467–480, 2003.

[7] A. J. Roberts, T. Kon, P. J. Knight, K. Sutoh, and S. A. Burgess. Functions and mechanics of dynein motor proteins. *Nature Reviews: Mol. Cell. Biol.*, 14(11): 713–26, 2013.

[8] H. Imai, T. Shima, K. Sutoh, M. L. Walker, P. J. Knight, T. Kon, and S. A. Burgess. Direct observation shows superposition and large scale flexibility within cytoplasmic dynein motors moving along microtubules. *Nat. Comm.*, 6(8179), 2015.

[9] Helgo Schmidt, Ruta Zalyte, Linas Urnavicius, and Andrew P Carter. Structure of human cytoplasmic dynein-2 primed for its powerstroke. *Nature*, 518(7539): 435, 2015.

[10] Gordon E Moore et al. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.

[11] Berni J Alder and T E Wainwright. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.

[12] M. Levitt. The birth of computational structural biology. *Nat. Struct. Biol.*, 8 (5):392–3, 2001.

[13] Shneior Lifson and Arieh Warshel. Consistent force field for calculations of conformations, vibrational spectra, and enthalpies of cycloalkane and n-alkane molecules. *The Journal of Chemical Physics*, 49(11):5116–5129, 1968.

[14] Michael Levitt and Shneior Lifson. Refinement of protein conformations using a macromolecular energy minimization procedure. *Journal of molecular biology*, 46(2):269–279, 1969.

[15] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[16] Paul D Morris, Andrew Narracott, Hendrik von Tengg-Kobligk, Daniel Alejandro Silva Soto, Sarah Hsiao, Angela Lungu, Paul Evans, Neil W Bressloff, Patricia V Lawford, D Rodney Hose, et al. Computational fluid dynamics modelling in cardiovascular medicine. *Heart*, 102(1):18–28, 2016.

[17] Dieter H Pahr and Philippe K Zysset. Finite element-based mechanical assessment of bone quality on the basis of in vivo images. *Current osteoporosis reports*, 14(6):374–385, 2016.

[18] Eli J Weinberg, Danial Shahmirzadi, and Mohammad Reza Kaazempur Mofrad. On the multiscale modeling of heart valve biomechanics in health and disease. *Biomechanics and modeling in mechanobiology*, 9(4):373–387, 2010.

[19] X. Yin, T. Thomas, and J. Zhang. Multiple red blood cell flows through microvascular bifurcations: Cell free layer, cell trajectory, and hematocrit separation. *Microvascular Research*, 89:47–56, 2013.

[20] Samuel C Zeeman, Jens Kossmann, and Alison M Smith. Starch: its metabolism, evolution, and biotechnological modification in plants. *Annual review of plant biology*, 61:209–234, 2010.

[21] M. Doi and S.F. Edwards. *The Theory of Polymer Dynamics.* Oxford University Press, New York City, NY 10016, United States, 1999.

[22] Carl Ivar Branden et al. *Introduction to protein structure.* Garland Science, 1999.

[23] Erich Sackmann. Biological membranes architecture and function. *Structure and Dynamics of Membranes*, 1:1–63, 1995.

[24] Reinhard Lipowsky and Erich Sackmann. *Structure and dynamics of membranes: I. from cells to vesicles/II. generic and specific interactions.* Elsevier, 1995.

[25] Daegeun Yoon and Donghyun You. Continuum modeling of deformation and aggregation of red blood cells. *Journal of biomechanics*, 49(11):2267–2279, 2016.

[26] Marc André Meyers, Po-Yu Chen, Albert Yu-Min Lin, and Yasuaki Seki. Biological materials: structure and mechanical properties. *Progress in Materials Science*, 53(1):1–206, 2008.

[27] Harvey T McMahon and Emmanuel Boucrot. Membrane curvature at a glance. *J Cell Sci*, 128(6):1065–1070, 2015.

[28] Ting Ye, Nhan Phan-Thien, and Chwee Teck Lim. Particle-based simulations of red blood cells—a review. *Journal of biomechanics*, 49(11):2255–2266, 2016.

[29] Zhen Li, Alireza Yazdani, Alexandre Tartakovsky, and George Em Karniadakis. Transport dissipative particle dynamics model for mesoscopic advection-diffusion-reaction problems. *The Journal of chemical physics*, 143(1):014101, 2015.

[30] Z. G. Mills, W. Mao, and A. Alexeev. Mesoscale modeling: solving complex flows in biology and biotechnology. *Trends in Biotechnology.*, 31(7):426–34, 2013.

[31] Jacob Dunningham and Vlatko Vedral. *Introductory quantum physics and relativity.* World Scientific, 2011.

[32] K. Burke. Perspective on density functional theory. *J. Chem. Phys.*, 136(15), 2012.

[33] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.

[34] Axel D Becke. Perspective: Fifty years of density-functional theory in chemical physics. *The Journal of chemical physics*, 140(18):18A301, 2014.

[35] Stefan Grimme, Jens Antony, Tobias Schwabe, and Christian Mück-Lichtenfeld. Density functional theory with dispersion corrections for supramolecular structures, aggregates, and complexes of (bio) organic molecules. *Organic & Biomolecular Chemistry*, 5(5):741–758, 2007.

[36] David A Pearlman, David A Case, James W Caldwell, Wilson S Ross, Thomas E Cheatham, Steve DeBolt, David Ferguson, George Seibel, and Peter Kollman.

Amber, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications*, 91(1-3):1–41, 1995.

[37] Yulian Gavrilov, Jessica D Leuchter, and Yaakov Levy. On the coupling between the dynamics of protein and water. *Physical Chemistry Chemical Physics*, 19 (12):8243–8257, 2017.

[38] Saeed Izadi, Ramu Anandakrishnan, and Alexey V Onufriev. Building water models: a different approach. *The journal of physical chemistry letters*, 5(21): 3863–3871, 2014.

[39] Alexey Onufriev. Continuum electrostatics solvent modeling with the generalized born model. *Modeling Solvent Environments: Applications to Simulations of Biomolecules*, pages 127–165, 2010.

[40] Tim Meyer, Marco D'Abramo, Adam Hospital, Manuel Rueda, Carles Ferrer-Costa, Alberto Pérez, Oliver Carrillo, Jordi Camps, Carles Fenollosa, Dmitry Repchevsky, et al. Model (molecular dynamics extended library): a database of atomistic molecular dynamics trajectories. *Structure*, 18(11):1399–1409, 2010.

[41] N. Kamiya, T. Mashimo, Y. Takano, T. Kon, G. Kurisu, and H. Nakamura. Elastic properties of dynein motor domain obtained from all-atom molecular dynamics simulations. *Prot. Eng. Des. & Sel.*, 29(8):317–25, 2016.

[42] Tom Darden, Darrin York, and Lee Pedersen. Particle mesh ewald: An nlog(n) method for ewald sums in large systems. *The Journal of chemical physics*, 98 (12):10089–10092, 1993.

[43] Jong Cheol Jo and Byeong Cheol Kim. Determination of proper time step for molecular dynamics simulation. *Bulletin of the Korean Chemical Society*, 21(4): 419–424, 2000.

[44] D. E. Shaw, M. M. Deneroff, and R. O. Dror *et al.* Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM*, 51 (7):91–97, 2008.

[45] David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[46] Shweta Purawat, Pek U Ieong, Robert D Malmstrom, Garrett J Chan, Alan K Yeung, Ross C Walker, Ilkay Altintas, and Rommie E Amaro. A kepler workflow tool for reproducible amber gpu molecular dynamics. *Biophysical Journal*, 112 (12):2469–2474, 2017.

[47] Alan Gray, Oliver G Harlen, Sarah A Harris, Syma Khalid, Yuk Ming Leung, Richard Lonsdale, Adrian J Mulholland, Arwen R Pearson, Daniel J Read, and Robin A Richardson. In pursuit of an accurate spatial and temporal model of biomolecules at the atomistic level: a perspective on computer simulation. *Acta Crystallographica Section D: Biological Crystallography*, 71(1):162–172, 2015.

[48] Arieh Warshel and Michael Levitt. Theoretical studies of enzymic reactions: dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *Journal of molecular biology*, 103(2):227–249, 1976.

[49] Hans Martin Senn and Walter Thiel. Qm/mm methods for biomolecular systems. *Angewandte Chemie International Edition*, 48(7):1198–1229, 2009.

[50] Sergei Izvekov and Gregory A Voth. A multiscale coarse-graining method for biomolecular systems. *The Journal of Physical Chemistry B*, 109(7):2469–2473, 2005.

[51] Siewert J Marrink, H Jelger Risselada, Serge Yefimov, D Peter Tieleman, and Alex H De Vries. The martini force field: coarse grained model for biomolecular simulations. *The journal of physical chemistry B*, 111(27):7812–7824, 2007.

[52] Siewert J Marrink and D Peter Tieleman. Perspective on the martini model. *Chemical Society Reviews*, 42(16):6801–6822, 2013.

[53] Priyanka Oroskar, Cynthia J Jameson, and Sohail Murad. Molecular dynamics simulations reveal how characteristics of surface and permeant affect permeation events at the surface of soft matter. *Molecular Simulation*, 43(5-6):439–466, 2017.

[54] AJ Rader, Chakra Chennubhotla, Lee-Wei Yang, Ivet Bahar, and Q Cui. The gaussian network model: Theory and applications. *Normal mode analysis: Theory and applications to biological and chemical systems*, 9:41–64, 2006.

[55] Yongmei Wang, AJ Rader, Ivet Bahar, and Robert L Jernigan. Global ribosome motions revealed with elastic network model. *Journal of structural biology*, 147 (3):302–314, 2004.

[56] P. Español and P. Warren. Statistical mechanics of dissipative particle dynamics. *Europhys. Lett.*, 30(4):191–96, 1995.

[57] Pep Español and Patrick B Warren. Perspective: Dissipative particle dynamics. *The Journal of Chemical Physics*, 146(15):150901, 2017.

[58] Sean R McGuffee and Adrian H Elcock. Diffusion, crowding & protein stability in a dynamic molecular model of the bacterial cytoplasm. *PLoS Comput Biol*, 6 (3):e1000694, 2010.

[59] Xuejin Li, Bruce Caswell, and George Em Karniadakis. Effect of chain chirality on the self-assembly of sickle hemoglobin. *Biophysical journal*, 103(6):1130–1140, 2012.

[60] W. M Lai, D. Rubin, and E. Krmpl. *Introduction to Continuum Mechanics.* Pergamon Press Ltd., third edition, 1993.

[61] Christian Grossmann, Hans-Görg Roos, and Martin Stynes. *Numerical treatment of partial differential equations*, volume 154. Springer, 2007.

[62] Konstantina Trivisa and Franziska Weber. A convergent explicit finite difference scheme for a mechanical model for tumor growth. *ESAIM: Mathematical Modelling and Numerical Analysis*, 51(1):35–62, 2017.

[63] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method.* Pearson Education, 2007.

[64] Adamandios Sifounakis, Sangseung Lee, and Donghyun You. A conservative finite volume method for incompressible navier–stokes equations on locally refined nested cartesian grids. *Journal of Computational Physics*, 326:845–861, 2016.

[65] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS)*, 41(2):11, 2015.

[66] J. N. Reddy. *An Introduction to the Finite Element Method.* New York: McGraw-Hill, New York, 1993.

[67] QY Lin, HQ Yang, SS Xie, YH Wang, Z Ye, and SQ Chen. Detecting early breast tumour by finite element thermal analysis. *Journal of medical engineering & technology*, 33(4):274–280, 2009.

[68] Idir Mellal, Emmanuel Kengne, Karim El Guemhioui, and Ahmed Lakhssassi. 3d modeling using the finite element method for directional removal of a cancerous tumor. *Journal of Biomedical Sciences*, 5(4), 2016.

[69] Andrew McCulloch, L Waldman, J Rogers, and Julius Guccione. Large-scale finite element analysis of the beating heart. *Critical reviews in biomedical engineering*, 20(5-6):427–449, 1992.

[70] Ashley E Morgan, Joe Luis Pantoja, Jonathan Weinsaft, Eugene Grossi, Julius M Guccione, Liang Ge, and Mark Ratcliffe. Finite element modeling of mitral valve repair. *Journal of biomechanical engineering*, 138(2):021009, 2016.

[71] R. Oliver, D.J. Read, O.G. Harlen, and S.A. Harris. A stochastic finite element model for the dynamics of globular macromolecules. *J. Comp. Phys.*, 239:147–65, 2013.

[72] RA Crowther. *The Resolution Revolution: Recent Advances In cryoEM*, volume 579. Academic Press, 2016.

[73] R. A. Richardson. *A mesoscale model for coarse-grained protein dynamics.* PhD thesis, University of Leeds Soft Matter Physics Group, 2014.

[74] Jeremy RH Tame and Beatrice Vallone. The structures of deoxy human haemoglobin and the mutant hb tyrα42his at 120 k. *Acta Crystallographica Section D: Biological Crystallography*, 56(7):805–811, 2000. PDB ID:1A3N.

[75] Daniel K Clare, Daven Vasishtan, Scott Stagg, Joel Quispe, George W Farr, Maya Topf, Arthur L Horwich, and Helen R Saibil. Atp-triggered conformational changes delineate substrate-binding and-folding mechanics of the groel chaperonin. *Cell*, 149(1):113–123, 2012. PDB ID:4AAU.

[76] Carlos J Bustamante, Christian M Kaiser, Rodrigo A Maillard, Daniel H Goldman, and Christian AM Wilson. Mechanisms of cellular proteostasis: insights from single-molecule approaches. *Annual review of biophysics*, 43:119–140, 2014.

[77] Maren Freutel, Hendrik Schmidt, Lutz Dürselen, Anita Ignatius, and Fabio Galbusera. Finite element modeling of soft tissues: material models, tissue interaction and challenges. *Clinical Biomechanics*, 29(4):363–372, 2014.

[78] Vladimir P Voloshin, Nikolai N Medvedev, Nikolai Smolin, Alfons Geiger, and Roland Winter. Exploring volume, compressibility and hydration changes of folded proteins upon compression. *Physical Chemistry Chemical Physics*, 17(13): 8499–8508, 2015.

[79] Marcio Gameiro, Yasuaki Hiraoka, Shunsuke Izumi, Miroslav Kramar, Konstantin Mischaikow, and Vidit Nanda. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics*, 32(1): 1–17, 2015.

[80] LD Landau and EM Lifshitz. Fluid mechanics. 1987. *Course of Theoretical Physics*, 1987.

[81] Krishan Kumar Chawla and MA Meyers. *Mechanical behavior of materials*. Prentice Hall, 1999.

[82] Raymond W Ogden. *Non-linear elastic deformations*. Courier Corporation, 1997.

[83] R. A. Richardson, K. Papachristos, and D. J. Read *et al.* Understanding the apparent stator-rotor connections in the rotary atpase family using coarse-grained computer modeling. *Proteins:Structure, Functions and Bioinformatics*, 82(12): 3298–311, 2014.

[84] S. M. Shontz and S. A. Vavasis. A robust solution procedure for hyperelastic solids with large boundary deformation. *Engineering with Computers.*, 28(2): 135–47, 2011.

[85] LD Landau and EM Lifshitz. Mechanics. 3rd. *Edition (Buttersworth-Heinemann, Burlington, MA, 2003)*, 16, 1976.

[86] E. Volterra and E. C. Zachmanoglou. *Dynamics of Vibrations*. Charles E. Merill Books, Inc, Columbus, Ohio, 1965.

[87] J. Howard. *Mechanics of Motor Proteins and the Cytoskeleton.* Sinauer Associates, Inc, Sunderland, MA 01375, United States, 2001.

[88] E. Volterra and E. C. Zachmanoglou. *Dynamics of Vibrations*. Charles E. Merill Books, Inc, Columbus, Ohio, 1965.

[89] R. G. Cox. The motion of long slender bodies in a viscous fluid. *J. Fluid. Mech.*, 44(4):791–810, 1970.

[90] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[91] Ardita Shkurti, Ramon Goni, Pau Andrio, Elena Breitmoser, Iain Bethune, Modesto Orozco, and Charles A Laughton. pypcazip: A pca-based toolkit for compression and analysis of molecular simulation data. *SoftwareX*, 2016.

[92] Grégoire Allaire and Sidi Mahmoud Kaber. *Numerical linear algebra*, volume 55. Springer, 2008.

[93] A. Solernou, B. Hanson, R. A. Richardson, Rob. Welch, D. J. Read, O. G. Harlen, and S. A. Harris. Fluctuating finite element analysis (ffea): A continuum mechanics software tool for mesoscale simulation of biomolecules. Submitted following revisions, September 2017.

[94] M. Bathe. A finite element framework for computation of protein normal modes and mechanical response. *Proteins-Structure, Function and Bioinformatics*, 70 (4):1595–1609, 2008.

[95] Gaël Guennebaud and Benoît Jacob *et al.* Eigen v3. http://eigen.tuxfamily.org, 2010.

[96] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. PyMOL The PyMOL Molecular Graphics System, Version 1.8, Schrödinger, LLC., November 2015.

[97] S Walter Englander and Leland Mayne. The nature of protein folding pathways. *Proceedings of the National Academy of Sciences*, 111(45):15873–15880, 2014.

[98] Jamie Snider, Guillaume Thibault, and Walid A Houry. The aaa+ superfamily of functionally diverse proteins. *Genome biology*, 9(4):216, 2008.

[99] Richard Serfozo. *Basics of applied stochastic processes*. Springer Science & Business Media, 2009.

[100] X. Y. Zhao, W. Sun, and J. P. Zhang *et al.* A model for the coordinated stepping of cytoplasmic dynein. *Biochem. Biophys. Res. Comm.*, 453(4):686–91, 2014.

[101] B. Hanson, R. Richardson, R. Oliver, D. J. Read, O. Harlen, and S. Harris. Modelling biomacromolecular assemblies with continuum mechanics. *Biochem. Soc. Trans.*, 43:186–92, 2015.

[102] WG Noid, Jhih-Wei Chu, Gary S Ayton, Vinod Krishna, Sergei Izvekov, Gregory A Voth, Avisek Das, and Hans C Andersen. The multiscale coarse-graining method. i. a rigorous bridge between atomistic and coarse-grained models. *The Journal of chemical physics*, 128(24):244114, 2008.

[103] B. Chance, H. Lees, and J. R. Postgte. The meaning of "reversed electron flow" and "high energy electron" in biochemistry. *Nature*, 238(5363):330–331, 1972.

[104] J Rosing and EC Slater. The value of δg for the hydrolysis of atp. *Biochimica et Biophysica Acta (BBA)-Bioenergetics*, 267(2):275–290, 1972.

[105] Paola Llinas, Tatiana Isabet, Lin Song, Virginie Ropars, Bin Zong, Hannah Benisty, Serena Sirigu, Carl Morris, Carlos Kikuti, Dan Safer, et al. How actin initiates the motor activity of myosin. *Developmental cell*, 33(4):401–412, 2015.

[106] Arnau Sebé-Pedrós, Xavier Grau-Bové, Thomas A Richards, and Iñaki Ruiz-Trillo. Evolution and classification of myosins, a paneukaryotic whole-genome approach. *Genome biology and evolution*, 6(2):290–305, 2014.

[107] M. Batchelor, M. Wolny, L. Dougan, E. Paci, and P. J. Knight. Myosin tails and single α -helical domains. *Biochem. Soc. Trans.*, 43(1):58–63, 2015.

[108] Lynne M Coluccio. *Myosins: a superfamily of molecular motors*, volume 7. Springer Science & Business Media, 2007.

[109] Kathleen M Trybus. Myosin v from head to tail. *Cellular and Molecular Life Sciences*, 65(9):1378–1389, 2008.

[110] Jeremy R Knowles. Enzyme-catalyzed phosphoryl transfer reactions. *Annual review of biochemistry*, 49(1):877–919, 1980.

[111] M Enrique, Amber L Wells, Steven S Rosenfeld, E Michael Ostap, and H Lee Sweeney. The kinetic mechanism of myosin v. *Proceedings of the National Academy of Sciences*, 96(24):13726–13731, 1999.

[112] Richard L Lieber. *Skeletal muscle structure, function, and plasticity*. Lippincott Williams & Wilkins, 2002.

[113] Amit D Mehta, Ronald S Rock, Matthias Rief, James A Spudich, Mark S Mooseker, and Richard E Cheney. Myosin-v is a processive actin-based motor. *Nature*, 400(6744):590–593, 1999.

[114] M Cecchini, A Houdusse, and Martin Karplus. Allosteric communication in myosin v: from small conformational changes to large directed movements. *PLoS computational biology*, 4(8):e1000129, 2008.

[115] Anne Houdusse and H Lee Sweeney. How myosin generates force on actin filaments. *Trends in Biochemical Sciences*, 41(12):989–997, 2016.

[116] Zhiyuan Shen, Angelo R Collatos, Jeffrey P Bibeau, Fabienne Furt, and Luis Vidali. Phylogenetic analysis of the kinesin superfamily from physcomitrella. *Frontiers in plant science*, 3:230, 2012.

[117] Nobutaka Hirokawa and Yosuke Tanaka. Kinesin superfamily proteins (kifs): various functions and their relevance for important phenomena in life and diseases. *Experimental cell research*, 334(1):16–25, 2015.

[118] Gohta Goshima and Ronald D Vale. Cell cycle-dependent dynamics and regulation of mitotic kinesins in drosophila s2 cells. *Molecular biology of the cell*, 16 (8):3896–3907, 2005.

[119] Christopher P Garnham and Antonina Roll-Mecak. The chemical complexity of cellular microtubules: tubulin post-translational modification enzymes and their roles in tuning microtubule functions. *Cytoskeleton*, 69(7):442–463, 2012.

[120] Johan OL Andreasson, Shankar Shastry, William O Hancock, and Steven M Block. The mechanochemical cycle of mammalian kinesin-2 kif3a/b under load. *Current Biology*, 25(9):1166–1175, 2015.

[121] Yasushi Okada and Nobutaka Hirokawa. Mechanism of the single-headed processivity: diffusional anchoring between the k-loop of kinesin and the c terminus of tubulin. *Proceedings of the National Academy of Sciences*, 97(2):640–645, 2000.

[122] Yuichi Inoue, Atsuko Hikikoshi Iwane, Takayuki Miyai, Etsuko Muto, and Toshio Yanagida. Motility of single one-headed kinesin molecules along microtubules. *Biophysical Journal*, 81(5):2838–2850, 2001.

[123] Andrew F Neuwald, L Aravind, John L Spouge, and Eugene V Koonin. Aaa+: A class of chaperone-like atpases associated with the assembly, operation, and disassembly of protein complexes. *Genome research*, 9(1):27–43, 1999.

[124] Michael A Cianfrocco, Morgan E DeSantis, Andres E Leschziner, and Samara L Reck-Peterson. Mechanism and regulation of cytoplasmic dynein. *Annual review of cell and developmental biology*, 31:83–108, 2015.

[125] Helgo Schmidt. Dynein motors: How aaa+ ring opening and closing coordinates microtubule binding and linker movement. *BioEssays*, 37(5):532–543, 2015.

[126] William B Redwine, Rogelio Hernández-López, Sirui Zou, Julie Huang, Samara L Reck-Peterson, and Andres E Leschziner. Structural basis for microtubule binding and release by dynein. *Science*, 337(6101):1532–1536, 2012.

[127] Helgo Schmidt and Andrew P Carter. Structure and mechanism of the dynein motor atpase. *Biopolymers*, 105(8):557–567, 2016.

[128] Weiyi Wang, Luyan Cao, Chunguang Wang, Benoît Gigant, and Marcel Knossow. Kinesin, 30 years later: Recent insights from structural studies. *Protein Science*, 24(7):1047–1056, 2015.

[129] Shiori Toba, Kotaro Koyasako, Takuo Yasunaga, and Shinji Hirotsune. Lis1 restricts the conformational changes in cytoplasmic dynein on microtubules. *Microscopy*, page dfv055, 2015.

[130] Katerina Toropova, Sirui Zou, Anthony J Roberts, William B Redwine, Brian S Goodman, Samara L Reck-Peterson, and Andres E Leschziner. Lis1 regulates dynein by sterically blocking its mechanochemical cycle. *Elife*, 3:e03372, 2014.

[131] W. Qui, N. D. Derr, and B. S. Goodman *et al.* Dynein achieves processive motion using both stochastic and coordinated stepping. *Nat. Struct. Mol. Biol.*, 19(2): 193–200, 2012.

[132] A. Sarlah and A. Vilfan. The winch model can explain both coordinated and uncoordinated stepping of cytoplasmic dynein. *Biophys. J.*, 107(3):662–71, 2014.

[133] Samara L Reck-Peterson, Ahmet Yildiz, Andrew P Carter, Arne Gennerich, Nan Zhang, and Ronald D Vale. Single-molecule analysis of dynein processivity and stepping behavior. *Cell*, 126(2):335–348, 2006.

[134] Bason E Clancy, William M Behnke-Parks, Johan OL Andreasson, Steven S Rosenfeld, and Steven M Block. A universal pathway for kinesin stepping. *Nature structural & molecular biology*, 18(9):1020–1027, 2011.

[135] Hailong Lu, M Yusuf Ali, Carol S Bookwalter, David M Warshaw, and Kathleen M Trybus. Diffusive movement of processive kinesin-1 on microtubules. *Traffic*, 10(10):1429–1438, 2009.

[136] Alexander R Dunn and James A Spudich. Dynamics of the unbound head during myosin v processive translocation. *Nature structural & molecular biology*, 14(3): 246, 2007.

[137] Yasunori Komori, Atsuko H Iwane, and Toshio Yanagida. Myosin-v makes two brownian 90 rotations per 36-nm step. *Nature structural & molecular biology*, 14 (10):968, 2007.

[138] Lin Li, Joshua Alper, and Emil Alexov. Cytoplasmic dynein binding, run length, and velocity are guided by long-range electrostatic interactions. *Scientific reports*, 6, 2016.

[139] Narutoshi Kamiya, Yukihisa S Watanabe, Satoshi Ono, and Junichi Higo. Amber-based hybrid force field for conformational sampling of polypeptides. *Chemical physics letters*, 401(1):312–317, 2005.

[140] R. C. Oliver. *A Stochastic Finite Element Model for the Dynamics of Globular Proteins.* PhD thesis, University of Leeds Soft Matter Physics Group, 2012.

[141] Takahide Kon, Takuji Oyama, Rieko Shimo-Kon, Kenji Imamula, Tomohiro Shima, Kazuo Sutoh, and Genji Kurisu. The 2.8 a crystal structure of the dynein motor domain. *Nature*, 484(7394):345, 2012.

[142] Kassandra M Ori-McKenney, Jing Xu, Steven P Gross, and Richard B Vallee. A cytoplasmic dynein tail mutation impairs motor processivity. *Nature cell biology*, 12(12):1228, 2010.

[143] T. Kon, K. Sutoh, and G. Kurisu. X-ray structure of a functional full-length dynein motor domain. *Nat. Struct. Mol. Biol.*, 18(6):638–U26, 2012. PDB ID:3VKH.

[144] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[145] D. Kim, C. Nguyen, and M. Bathe. Conformational dynamics of supramolecular protein assemblies. *J. Struct. Biol.*, 173(2):261–70, 2011.

[146] Giseok Yun, Jaehoon Kim, and Do-Nyun Kim. A critical assessment of finite element modeling approach for protein dynamics. *Journal of Computer-Aided Molecular Design*, pages 1–16, 2017.

[147] Andreja Šarlah and Andrej Vilfan. Minimum requirements for motility of a processive motor protein. *PloS one*, 12(10):e0185948, 2017.

[148] Salma Kassem, Thomas van Leeuwen, Anouk S Lubbe, Miriam R Wilson, Ben L Feringa, and David A Leigh. Artificial molecular motors. *Chemical Society Reviews*, 46(9):2592–2621, 2017.

[149] A. W. Peters, A. J. Howarth, and O. K. Farha. Experimentalists and theorists need to talk, 2017. Nature Comment.

[150] Xiaohu Hu, Liang Hong, Micholas Dean Smith, Thomas Neusius, Xiaolin Cheng, and Jeremy c Smith. The dynamics of single protein molecules is non-equilibrium and self-similar over thirteen decades in time. *Nature Physics*, 12(2):171, 2016.