Multi-resolution Modelling of Biomolecules



Ravinda Gunaratne

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2021

Acknowledgements

I would like to begin by thanking my supervisor, Radek Erban, for all of the motivation, guidance, and support he has provided over the course of my DPhil studies. Further regards go to Daniel Wilson, Mark Flegg, Cal Floyd, Garyk Papoian, and Haoran Ni for their collaboration on the work that went into forming this thesis.

I am grateful for the opportunity afforded to me when I was given a place to study at the Doctoral Training Centre as part of the Systems Biology cohort (special thanks to Rasa, Naomi, and the Squash Buds! Rohan, Lance, and Jacob) and for the funding provided by EPSRC. In addition, Further thanks go to my colleagues in the Mathematical Institute for stimulating conversation, providing a welcoming environment for work, and making it fun to come into the office.

I also want to express my deep gratitude to the friends who made my time away from my studies in Oxford so wonderful, especially my ever-patient housemates Florian, Sam, and Cam; I will always have fond memories of the years spent in 39 Bullingdon Road. To the rest of the Wolfson crew (Maddie, Louise, Danielle, Fig, James, Laurynas, Leo, Joe) for that magical first year, John for his boundless enthusiasm, Lynn and Mary for always being around for a gossip over a bottle of wine, Dan for the hundreds of hours spent on the squash court together, Laura for not leaving me behind on our cycles, Emma for being a legend, Jonny for his delicious home-brews, Ross for helping me find infinity, and Poppy for always being smashing company.

Special thanks also go to my friends away from Oxford. To Jasmine, Damien and Rosie for always having space on the sofa for me in London, the Pedlars (Rob, Sam, Jack, Judith) who are a constant source of inspiration, and my Belfast mates (Jake, Aaron, Shane, Rory, James, Laura, Rachel) for pulling me through the dreariest bits of the pandemic.

I would like to finish by thanking the Halahakoons, Ramasubbus, AP, Archana and Shevanthi for a lifetime of support and good company, my extended family for their well wishes from afar, Florence for being my rock and my comfort throughout the ups and downs of writing this thesis, and finally my parents, to whom greatest credit goes, for their guidance and giving me the encouragement I've needed to flourish throughout my life.

Abstract

This work investigates how multi-resolution modelling can be applied to study biomolecules as a means to improve the efficiency of simulations. In practice, we resolve aspects of a model that are of greatest biological or physical interest in high detail, while describing the remainder more coarsely. We begin by taking a multi-resolution approach to solvent in simulations, replacing the forces a simple dimer model experiences in short-range and long-range heat baths with macroscopic Langevin and generalised Langevin models. The parameters for these stochastic models are determined analytically, or inferred from molecular dynamic simulation data via force and velocity correlation statistics for comparison. Additionally, the solvent of each monomer in the dimer are also modelled with different level of resolution, with these methodologies used to investigate the effect of a shared heat bath versus two uncoupled heat baths, one for each monomer. Attention is then turned to models of a target molecule, with a multi-resolution methodology for modelling F-actin filaments presented. It provides detailed microscopic information at the level of individual monomers at a lower computational cost by replacing the monomer-based model in parts of the simulated filament by a rod-based Cosserat description. In the monomer-based description G-actin is represented by ellipsoids bound at the surface in a double helical configuration to form F-actin. The rod-based model is coarser, in which F-actin is described using a Cosserat model. The multi-resolution methodology is illustrated using case studies, designed to test the properties of F-actin under stretching, bending, shearing and twisting.

Contents

List of Figures **vi**

List of Tables **viii**

1 Introduction **1**
- 1.1 Introduction to molecular dynamics . . . 3
- 1.2 Coarse-grained molecular dynamics . . . 11
- 1.3 Multi-resolution and multiscale approaches in molecular dynamics . 15
- 1.4 Modelling of actin and the cytoskeleton . . . 18
- 1.5 Structure of thesis . . . 25

2 Multi-Resolution Molecular Models with Langevin Dynamics **27**
- 2.1 Multi-resolution dimer model . . . 28
  - 2.1.1 The dimer model . . . 29
  - 2.1.2 Short-range interaction heat bath . . . 32
  - 2.1.3 Long-range interaction heat bath . . . 47
- 2.2 Extension to generalised Langevin model . . . 54
  - 2.2.1 Alternative oscillator frequencies and interaction constant parametrisations . . . 56
  - 2.2.2 Non-Markovian random noise . . . 61
  - 2.2.3 Kernel estimation . . . 70
- 2.3 Discussion . . . 75

3 Multi-Resolution Model of an Actin Filament **80**
- 3.1 Ellipsoid and Cosserat models of actin filaments . . . 81
- 3.2 Multi-resolution coupling region . . . 83
- 3.3 Matching the Ellipsoid and Cosserat models . . . 86
  - 3.3.1 Ellipsoid model parametrisation . . . 87
  - 3.3.2 Cosserat model parametrisation . . . 90
  - 3.3.3 Matching Ellipsoid and Cosserat parameters . . . 91
  - 3.3.4 Cosserat twist-stretch coupling . . . 97
- 3.4 Discussion . . . 100

4 Multi-Resolution Actin Modelling Applications and Extensions 103
4.1 Integration of equations of motion for multi-resolution model 104
4.1.1 Symplectic integrators 104
4.1.2 Other numerical methods 108
4.2 Effect on dynamics due to dissipation, mass, and inertia 110
4.3 Example cases 113
4.3.1 Cantilever filament 114
4.3.2 External linker forces 115
4.3.3 Twisted filaments 117
4.4 Discussion 120

5 Summary and conclusions 124

Appendices

A Analytic Solution for Hamiltonian System in Chapter 2 128

B Alternate Multi-Resolution Coupling Methods for Chapter 3 133
B.1 Simpler multi-resolution couplings and their limitations 133
B.2 Constraint algorithm coupling methods 138
B.2.1 SHAKE algorithm 138
B.2.2 RATTLE algorithm 140
B.2.3 Projecting positions and velocities into coupling region . . . 142

C Further Details and Simulation Timings for Chapter 4 146
C.1 Full energy plots for example cases 146
C.2 Simulation timings 150

References 152

List of Figures

1.1 Illustration of bound interactions in glycine, an amino acid 6
1.2 Illustration of unbound interactions between water molecules 8
1.3 All-atom and cartoon representation of F-actin filament 10
1.4 Molecular dynamics at different time and length scales 12
1.5 Multi-resolution simulation of DNA molecule in a salt solution . . . 16
1.6 Coarse-grain multi-resolution model of a DNA chain 17
1.7 Schematic of various biological molecules involved in large scale cytoskeleton simulations, using software package MEDYAN 19
1.8 Coarse-grained representation of cofilin decorated actin filament . . 24

2.1 A diagrammatic representation of multi-resolution approaches for a dimer in a heat bath with short-range interactions 35
2.2 Plot of the acceptance probability for acceptance-rejection algorithm 38
2.3 The extension of the average length of a dimer from its separation distance ℓ_0 for short-range interaction heat bath models 43
2.4 The velocity autocorrelation function for the multi-resolution model for short-range interactions 46
2.5 The extension of the average length of a dimer from its separation distance ℓ_0 for long-range interaction heat bath models 50
2.6 The velocity autocorrelation function for the multi-resolution model for long-range interactions 55
2.7 Accuracy of Dirac delta function approximation of the friction kernel when sampling frequencies from exponential distribution 57
2.8 Accuracy of Dirac delta function approximation of the friction kernel when sampling frequencies from uniform distribution 59
2.9 Comparison of friction kernels for different choices of solvent particle oscillator numbers 60
2.10 Effect of rate of decay of friction kernel on properties of noise sampled according to fluctuation-dissipation theorem 65
2.11 Demonstration of efficacy of numerical scheme to solve Volterra integro-differential equations for the velocity auto-correlation function 68

2.12 Simulating generalised Langevin equation friction kernels derived from Hamiltonian dynamics 70
2.13 Kernel estimation techniques used on generalised Langevin model simulation data 73
2.14 Estimating friction kernel directly from Hamiltonian simulation data, resulting in updated Hamiltonian-GLE multi-resolution model . . . 76

3.1 A schematic of Ellipsoid and Cosserat models in the multi-resolution framework. 81
3.2 Illustration of multi-resolution coupling region. 83
3.3 Twisting rigidity of ellipsoidal actin filament for varying curvature. . 95
3.4 Stretching of ellipsoidal actin filament for varying curvature. 99

4.1 Dissipation used to reach equilibrium energy and configuration. . . 111
4.2 Illustration of example cases for multi-resolution actin filaments. . . 113
4.3 Multi-resolution actin filament under a cantilever force. 114
4.4 Multi-resolution actin filament attached to 3 planar external linkers. 116
4.5 Multi-resolution actin filament attached to 4 non-planar external linkers. 117
4.6 The spatial distribution of potential energy under torsion for Cosserat and ellipsoidal filaments 118
4.7 Multi-resolution actin filaments under compression and twist. 119
4.8 Multi-resolution actin filaments under compression and twist, with variable twisting rigidity $B_3(\kappa_{\mathcal{L},3})$ and length scale factor $\xi(\kappa_{\mathcal{L},3})$ . . 121

A.1 Qualitative comparison of position and velocity trajectories for simulated single particle Hamiltonian model compared to analytic solutions 131
A.2 The velocity autocorrelation function for the single particle for long-range interactions model 131

C.1 Full dynamic energy profile for the cantilever example case. 147
C.2 Full dynamic energy profile for the planar external linkers example case. 148
C.3 Full dynamic energy profile for the non-planar external linkers example case. 149
C.4 Full dynamic energy profile for the over-twist example case. 149
C.5 Full dynamic energy profile for the under-twist example case. 150
C.6 Relative simulation times for example cases presented. 151

List of Tables

1.1 Variables and parameters found in example molecular dynamics potential energy expression 4

2.1 Acceptance-rejection algorithm for sampling random numbers according to the probability distribution $p(z;\beta)$ 37
2.2 One iteration of the multi-resolution simulation algorithm of the dimer in a co-moving frame 42
2.3 One iteration of the multi-resolution simulation algorithm of the dimer in the heat bath with short-range interactions 45
2.4 One iteration of the multi-resolution simulation algorithm of the dimer in the heat bath with long-range interactions 53
2.5 One iteration of generalised Langevin model simulation algorithm for single monomer 66
2.6 One iteration of the updated multi-resolution simulation algorithm of the dimer in the heat bath with long-range interactions 75

3.1 Parameters used to calibrate the Ellipsoid model used in Schramm et al. [119], along with maximum interface bond lengths. 90
3.2 Parameters of the Cosserat model for F-actin used in Floyd et al. [3]. 91
3.3 Twisting rigidity of ellipsoidal actin filament at various curvatures. . 94
3.4 Parameters used for the Ellipsoid and Cosserat models for multi-resolution model. 96
3.5 Compression/extension of an ellipsoidal actin filament at various curvatures. 98

4.1 One iteration of the multi-resolution simulation algorithm for F-actin filaments 109

1 Introduction

Over the course of this work, we explore how multi-resolution modelling approaches can be applied to studies involving biomolecules. Our motivation is to find a balance between the accuracy provided by high resolution models and the efficiency of low resolution models of the same particles or molecules. High resolution approaches like all-atom molecular dynamics (MD) simulations give a microscopic view, with subtle biochemical properties and interactions accounted for. This is useful when, for example, we need to know the specific shape of a protein binding site to determine how likely a drug molecule is to bind to the surface. However, this comes at a cost, and it quickly becomes infeasible to scale these methods up for large structures. There are situations where this level of detail is not needed to study specific phenomena of interest. When considering a protein's movement through water, in high resolution we would model each individual H_2O molecule and their interactions with the protein surface. Alternatively, we can model the collective effect of water, replacing multiple chemical forces with cumulative drag and diffusive forces. Similar reasoning can be applied when we are more interested in the macroscopic physical properties of a molecule. Large proteins can be approximated as spheres or ellipsoids, while long biopolymers can be described as simple chains or thin rods. Though seemingly simplistic at the small length scales explored using high resolution models, these low resolution models enable the study of complex

interacting systems for large length and time scales; for example, we could run an MD simulation for a 100 nm actin filament, or use a rod-based model to simulate a complex system of cross-linked actin filaments which are several μm long over the same time scale using similar computational power. We explore situations where it is possible to have the best of both worlds; using both low and high resolution models simultaneously in the same simulation. Our choice of low resolution models and their parameters is informed directly by the high resolution representations, either by statistical inference or direct measurement of physical properties. This means that when 'zoomed out', the macroscopic behaviour of the multi-resolution model resembles that of both low and high resolution schemes. By switching regions of less interest in our studies to low resolution, we are able to improve model efficiency while maintaining high resolution accuracy where it is needed.

In this chapter we will lay out the background to research carried out in later chapters. Section 1.1 introduces details of the mathematical and computational elements that go into designing robust and accurate MD simulations. We then move from all-atom MD to coarse-grained simulations in Section 1.2, looking at the benefits that can arise by compromising on accuracy and detail of simulations to widen the scope of biological applications. Methods used to parameterise these models are described, often utilising fine-grain data, spatially large scale biological experiments, and measurable material properties. Section 1.3 then turns to the fusion of these paradigms in multi-resolution and multiscale models, where spatial and temporal resolution is not necessarily uniform across a single model. Methods in the literature range from varying the level of detail of solvent and target molecules, to using different numerical integration timesteps sizes for different regions of a simulation. In Section 1.4, we give an overview of actin and cytoskeleton modelling, and introduce monomer-based and rod-based coarse-grain models of F-actin filaments, giving detailed descriptions of the equations of motion for models utilised in Chapters 3–4. Finally, in Section 1.5, this chapter ends with a brief outline of the substantive work of this thesis.

1.1 Introduction to molecular dynamics

MD approaches, based on the rules of classical mechanics, have been widely used to study the behaviour of complex biomolecules in biological applications. They are written in terms of the positions and velocities of particles, representing either individual atoms or groups of atoms, describing parts of a biomolecule, and tend to be carried out with solvent interactions accounted for [1–3]. We begin by looking at classical molecular dynamics, in which we numerically integrate Newton's equations of motion for all of the particles involved in the simulation. For example, if we take N particles, with $\boldsymbol{X}_i \in \mathbb{R}^3$ and $\boldsymbol{V}_i \in \mathbb{R}^3$, representing the position and velocity of the i^{th} particle respectively, $i = 1, 2, \ldots, N$, we can write the energy of this system as

$$E = \Phi(\boldsymbol{X}) + \frac{1}{2}\sum_{i=1}^{N} M_i \boldsymbol{V}_i^2,$$

where $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N\}$ is the $3N$-dimensional position state vector and $\Phi(\boldsymbol{x}) \in \mathbb{R}$ is the potential energy of the system in a given conformation of $\boldsymbol{X}$. Defining $\boldsymbol{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_N\}$ and using Lagrangian formalism, it is then possible to describe the dynamics of the system in time $t \in \mathbb{R}$ by using the Euler-Lagrange equations to find the equations of motion

$$\frac{\mathrm{d}\boldsymbol{X}_i}{\mathrm{d}t} = \boldsymbol{V}_i, \qquad M_i \frac{\mathrm{d}\boldsymbol{V}_i}{\mathrm{d}t} = -\frac{\partial \Phi}{\partial \boldsymbol{X}_i}. \tag{1.1}$$

Inter-particle forces in MD models include combinations of short-range and long-range interactions [4, 5]. Choosing how to model these interactions is a crucial aspect of most molecular dynamics simulations, and the most computationally expensive element of the process. Usually, we must consider bound and unbound interactions, defined by certain potential energy expressions; the bound potentials describe the bonding structure of the molecule, while the unbound potentials describes general interactions between atoms. Here is an example of a possible potential covering short-range and long-range interactions, containing terms regularly used

R_{ij}	Distance between the i^{th} and j^{th} particles
θ_{ijk}	Angle between bonds from i^{th} to j^{th} and j^{th} to k^{th} particles
ϕ_{ijkl}	For i, j, k and l, angle between (i, j, k) plane and (j, k, l) plane
$k_{ij,bonds}$	Spring constant for bond
ℓ_0	Equilibrium bond length
$k_{ijk,angle}$	Angle spring constant
θ_0	Equilibrium bond angle
$k_{ijkl,dihedral}$	Dihedral scaling constant
m_{ijkl}	Periodicity of dihedral potential for particles i, j, k and l
δ_{ijkl}	Phase shift angle for particles i, j, k and l
σ_{ij}	Separations of i^{th} and j^{th} particles at minimum of potential
ε_{ij}	Minimum of Lennard-Jones potential for i^{th} and j^{th} particles
ε_0	Dielectric constant
q_i	Charge on the i^{th} particle

Table 1.1: *Variables and parameters found in example molecular dynamics potential energy expression shown in equation* (1.2)

in computational MD simulations [6],

$$\begin{aligned}\Phi(\boldsymbol{X}) = &\sum_{\text{bonds}} k_{ij,\text{bonds}}(R_{ij} - \ell_0)^2 + \sum_{\text{angles}} k_{ijk,\text{angles}}(\theta_{ijk} - \theta_0)^2 \\ &+ \sum_{\text{dihedrals}} k_{ijkl,\text{dihedrals}}(1 + \cos(m_{ijkl}\phi_{ijkl} + \delta_{ijkl})) \\ &+ \sum_{i}^{N}\sum_{j\neq i}^{N} 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{R_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{R_{ij}}\right)^{6}\right] + \sum_{i}^{N}\sum_{j\neq i}^{N}\frac{q_i q_j}{\varepsilon_0 R_{ij}},\end{aligned} \tag{1.2}$$

with variables and parameters defined in Table 1.1. These terms are validated against physical theory and experimental expectations to ensure they can be trusted to give accurate simulation results [7]. Here, the first term is a summation over all pairs of particles that are connected by a simple covalent bond, giving the oscillations between two atoms by representing them as beads attached by a spring using Hooke's Law, with the distance between particles $R_{ij} = |\boldsymbol{X}_i - \boldsymbol{X}_j|$, kept close to a rest length ℓ_0. In Chapter 2 we investigate a very simple example using this potential, describing the single bond between two monomers to form a dimer. However, when extending beyond two particle interactions to capture more complex structures, angle constraints are also required. The second term in equation (1.2) is a summation over all bond angle constraint triplets, describing the oscillations in the bond angle between three atoms bonded in a row, and can

be defined as a function of their positions

$$\theta_{ijk} = \arccos\left(\frac{|(\boldsymbol{X}_j - \boldsymbol{X}_i)\cdot(\boldsymbol{X}_j - \boldsymbol{X}_k)|}{|\boldsymbol{X}_j - \boldsymbol{X}_i||\boldsymbol{X}_j - \boldsymbol{X}_k|}\right).$$

In addition to these 3-particle angle constraints, Urey-Bradley potentials [8] are sometimes incorporated into models (e.g. for water molecules), and in this case would simply take the form of a virtual harmonic bond between $\boldsymbol{X}_i$ and $\boldsymbol{X}_k$. While these can improve behaviour, they can lead to over-fitting issues when parameterising a model, so are less widely used than simple angle constraints. The third term is a summation over all dihedral interactions, describing the rotational torsion force between a group of four atoms. This can also be thought of as a constraint on the angle between the normal vectors of the planes formed by the atoms, defined again with only position being a variable

$$\phi_{ijkl} = \arccos\left(\frac{|\boldsymbol{n}_{ijk}\cdot\boldsymbol{n}_{jkl}|}{|\boldsymbol{n}_{ijk}||\boldsymbol{n}_{jkl}|}\right),$$
$$\boldsymbol{n}_{ijk} = (\boldsymbol{X}_j - \boldsymbol{X}_i)\times(\boldsymbol{X}_j - \boldsymbol{X}_k), \quad \boldsymbol{n}_{jkl} = (\boldsymbol{X}_k - \boldsymbol{X}_j)\times(\boldsymbol{X}_k - \boldsymbol{X}_l),$$

while m_{ijkl} and δ_{ijkl} are constants that determine the shape of the potential, dictating its maxima and minima, and therefore the energetically preferred orientation of these bonds. In Figure 1.1 we see an illustration of these potentials in action, where a consistent choice of dihedral potential is

$$k_{ijkl}\left(1 + \cos\left(\phi_{ijkl} + \frac{2\pi}{3}\right)\right),$$

which is 2π periodic example of, with a minimum at $\phi = \pi/3$. This form of potential is often used to model proper dihedrals, where 4 particles involved are bonded in a row, with full rotations permissable. Improper dihedral effects, which occur when 3 particles are bound to a central particle, generally use same form of potential as simple 3-particle harmonic angle constraints as these interactions tend to be very stiff with a single minimum. Using a combination of these potentials it is possible to account for the bound interactions, and thus the structure of a complex biomolecule locally. In Chapter 3 we will also look at an extended coordinate system, where in addition to spatial positions like the $\boldsymbol{X}_i$, we also explicitly track the evolution

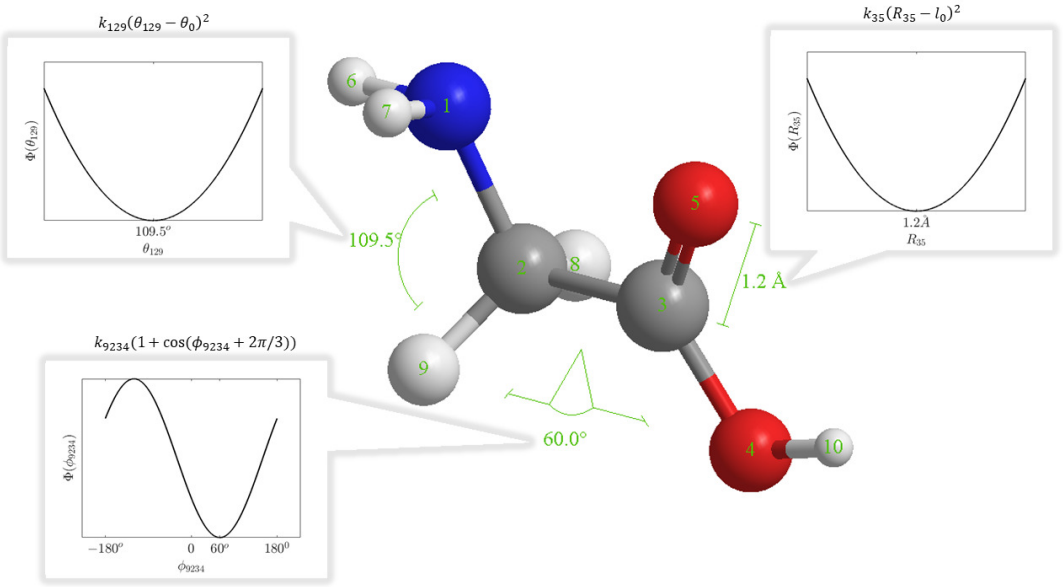


Figure 1.1: *Illustration of bound interactions in glycine, an amino acid. Despite being a reasonably simple molecule, we can see that harmonic bond, angle, and dihedral constraint potentials (as seen in equation* (1.2)*) are required to fully describe its spatial configuration. Forms of potential are highlighted for bond length* R_{35}*, angle* θ_{129}*, and dihedral* ϕ_{9234} *(where each index corresponds to atom number labels), with minima at* $1.2\mathring{A}$*,* 109.5^o*, and* 60^o*, respectively.*

of the orientation of segments of the model using Euler angles [9], where further reference to these kinds of angular constraints will be studied.

By limiting ourselves to localised interactions (in the examples above only particles one, two and three bonds apart are involved in bond, angle and harmonic interactions) it is possible to account for most of the behaviour of a target molecule accurately, while putting a cap on the computations cost of simulations, i.e. keeping the total number of interaction $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$, rather than $\mathcal{O}(N^2)$. However, this can break down in the case of very large biomolecules, such as proteins, where structures are folded and tangled, and atoms that begin a simulation spatially separated can end up close together. To prevent overlapping molecules and to account for electrostatic effects the final terms of equation (1.2) are included, and are known as the Lennard-Jones [10, 11] and Coulomb potentials [4] respectively. Most modern molecular dynamics software packages can keep the number of evaluated interactions at $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$ even when non-local interactions are considered. Assuming $\varepsilon_{ij} > 0$, the Lennard-Jones potential has a positive spike at $R_{ij} = 0$

showing there will be strong repulsion when particles get too close together, a minimum of $-\varepsilon_{ij}$ at $R_{ij} = 2^{1/6}\sigma_{ij}$ indicating the potential can also give rise to attractive forces, and then tails off to zero as particles get too far apart. In more common terms, these are known as van der Waals interactions, and are often only evaluated for non-bonded interaction as they introduce redundancy when combined with harmonic bond potentials. In comparison the Coulomb potential (final term of equation (1.2)) is simpler as it has no maxima or minima, and results in a force that decays in R_{ij} which is attractive if the charges q_i and q_j are of the opposite sign, and repulsive if they are the same sign.

Similar potentials can be designed to account for the dynamics of the water molecules and their interactions with the molecules [12], with an illustration of effects of the Lennard-Jones potential given in Figure 1.2. Biologically relevant simulations have to be done in aqueous solutions, and fully solvating a target molecule is one of the most computationally taxing aspects of MD simulations. A number of efficient water models have been developed in the literature for use in all-atom MD simulations, including commonly used three-site models [13, 14]. These sites usually correspond to the oxygen and two hydrogen atoms in each molecule. Beyond this, 4-site [15], 5-site [16, 17], and even 6-site [18] models of water have also been formulated and incorporate dummy sites, which try to account for the uneven electrostatic distribution in water that arises from the positioning of lone pairs on the oxygen. Each additional site increases the computational complexity, but can bring simulations closer to the physical reality, especially under more extreme pressures or temperatures. As an illustrative point, under atmospheric pressure the 5-site TIP5P model has a computational melting point of 273.9K, while in the 3-site SPC/E model the corresponding value is 215K [19], which compares worse with the theoretically expected 273.2K. These discrepancies are borne out across many of the other physical properties of water [20], and for this reason it is important to make the correct choice of solvent medium based on the investigation carried out [21], with changes in water model often affecting the binding energies and overall behaviour of biomolecules like proteins [22].

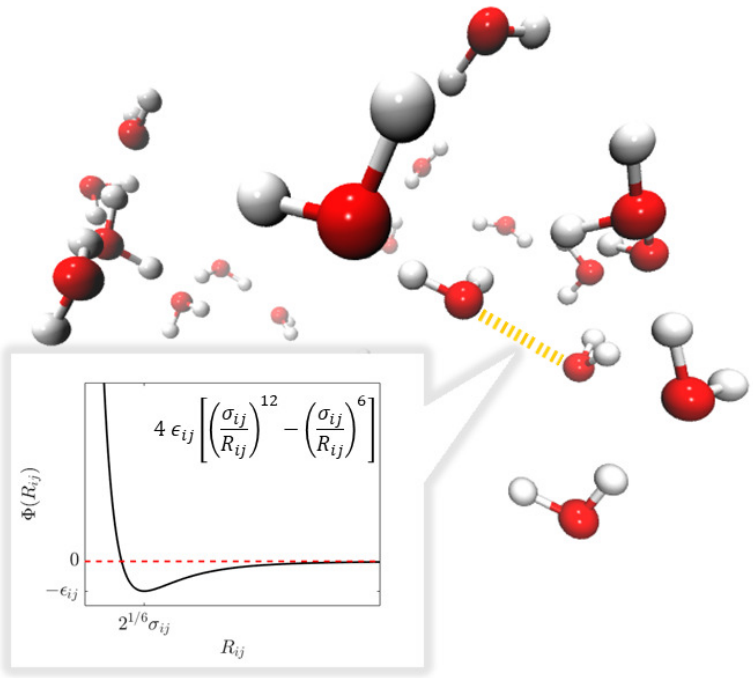


Figure 1.2: *Illustration of unbound interactions between water molecules. These can be modelled by a Lennard-Jones potential, where forces are a function of the radial distance between particles, as seen in equation* (1.2)*. As the particles get close together the potential rises sharply, creating a repulsive force. When particles are far apart the energy tails off to zero, so force evaluation can be cut off past a certain point, improving efficiency.*

With this in mind, it also makes sense to implement controls that maintain a constant temperature or pressure during simulations so that we are able to apply the results for various statistical physics ensembles, along with the fact that these are conditions that are often controlled when collecting experimental data. This can be achieved by coupling new equations to the existing model [23, 24], scaling velocities in a systematic manner [25], or even adding stochastic forces or velocities to a system [26]. In the Nosé-Hoover regime [23, 24] a friction term which is proportional to particle velocity is added to the forces in the original model. The magnitude of this force is scaled by new fictional coordinates, and can be thought of as coupling the original system with a virtual heat bath, and maintains ensemble properties while controlling the temperature of the system. On the other hand, the Berendsen thermostat instead scales the velocities directly at each timestep in a manner that ensures the change in temperature is proportional to the difference in

current and target temperatures [25]. In a different, non-deterministic approach, the Anderson thermostat uses a stochastic collision method as a means of controlling temperature, with random instantaneous collisions changing the momentum of particles in the system, with the frequency of these collisions decided by the user via a Poisson distribution [26]. By using these schemes to limit the temperature to a given range we also consequently reduce numerical errors by keeping resultant forces in a constrained range. However, if the parameterisation or style of thermostat is not chosen carefully, the dynamics of a simulations can be adversely affected and results can diverge from the those observed in reality [27].

Having constructed a computational model in which we can account for a target molecule and solvent, we then need to set up a simulation using realistic molecular coordinates and model parameters. Data for all-atom MD simulations can be obtained via x-ray crystallography [28] or cryogenic electron microscopy [29, 30], with researchers often uploading structures of interest to the RCSB Protein Data Bank (PDB) [31], with standard format `.pdb` and `.psf` files containing information used to set up the initial spatial geometry of the molecules and parameters for the potentials respectively. In Figure 1.3 we see an example all-atom and cartoon representation of actin generated from the coordinate files mentioned above, illustrating the level of detail that can be incorporated into MD simulations of this scale. Here 13 G-actin monomers are bound to form a F-actin filament around 70 nm long, which is shorter than the μm length filaments seen in the cell cytoskeleton; this level of accuracy comes at a computational cost. Once all the inter-particle interactions, coordinates, parameters are specified, the resulting system of ordinary or stochastic differential equations of motions must be numerically integrated in time so that we can find the trajectories of the particles in the system [33–35]. One of the most commonly used methods in MD simulations is the velocity Verlet algorithm [36], which is a type of symplectic integrator, meaning that the energy of a system is conserved in time. Taking the system in equation (1.1) as an example, the

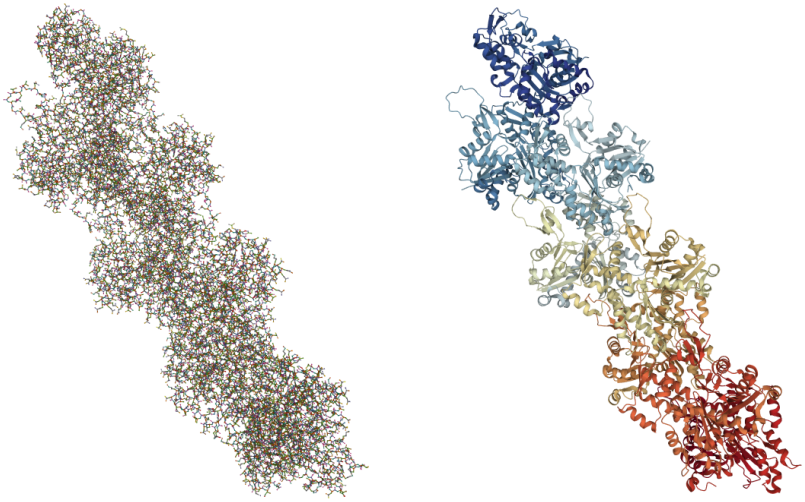


Figure 1.3: *All-atom and cartoon representation of F-actin filament. In the all-atom representation, we can see the individual bonded elements which form the protein structure. In this particular structure, there are around 30,000 non-hydrogen atoms. While this level of detail is useful for capture subtle chemical properties, it is hard to qualitatively study. The cartoon representation is a lower resolution way of visualising a protein, making it easier to identify important sections of the molecule. Figure created using data available from the Protein Data Bank of structure* 6BNO [32] *with permission (Creative Commons attribution license).*

procedure for taking one timestep would be

$$\begin{aligned}
\boldsymbol{V}_i\left(t+\tfrac{1}{2}\Delta t\right) &= \boldsymbol{V}_i(t) - \frac{1}{2M_i}\left[\frac{\partial \Phi}{\partial \boldsymbol{X}_i}(t)\right]\Delta t, \\
\boldsymbol{X}_i(t+\Delta t) &= \boldsymbol{X}_i(t) + \boldsymbol{V}_i\left(t+\tfrac{1}{2}\Delta t\right)\Delta t, \\
\boldsymbol{V}_i(t+\Delta t) &= \boldsymbol{V}_i\left(t+\tfrac{1}{2}\Delta t\right) - \frac{1}{2M_i}\left[\frac{\partial \Phi}{\partial \boldsymbol{X}_i}(t+\Delta t)\right]\Delta t,
\end{aligned}$$

and avoids evaluation of a term of order $(\Delta t)^2$ for the acceleration, which can help improve numerical accuracy and avoid finite precision errors [37]. We make use of this integrator in Chapter 2, while in Chapter 4 we use higher order symplectic integrators to improve the stability and accuracy of our simulations, with the caveat that they result in extra evaluations of the force term for each timestep. In all-atom molecular dynamics simulations this is usually too restrictive so the simple Verlet algorithm is most commonly used, though other methods such as Runge-Kutta schemes [38] can also be used as integrators. Constraint algorithms are often used to maintain rigid bond lengths as an alternative to evaluating the harmonic bond

forces at every timestep. Often a high spring constant value is the limiting factor in choice of timestep size and the small vibrations often have a negligible impact on overall dynamics. An optimisation problem can be formulated using Lagrange multipliers and solved at each timestep to maintain bond lengths within a tolerance of their desired length. The most well known version is known as the SHAKE algorithm, with the RATTLE algorithm being analogous version designed to fit in with the Verlet integration scheme [39]. Further methods have been designed that provide faster convergence of the optimisation calculation at each time step [40], account for the angular velocity and orientation of rigid molecules [41], along with versions that have been tailored to water models [42]. In Chapter 3 we provide a detailed picture of the mathematics behind some of these methods and their efficacy in application. Many comprehensive software packages exist that incorporate most of these elements, with some of the most well known including NAMD and VMD [43, 44], GROMACS [45], CHARMM [6], and LAMMPS [46].

1.2 Coarse-grained molecular dynamics

While we have seen that, by capturing the fine chemical bonding structure of biomolecules and bound ligands, all-atom MD simulations can yield great scientific results, this detail comes at a high computational cost. Simulations of this form can be used to investigate the physical properties of small numbers of proteins but it is infeasible to use this level of detail for large systems which may also be of biological interest [47, 48]. Often to capture phenomena of interest, tens of thousands of atoms must be simulated (e.g. the short F-actin filament in Figure 1.3). To give a sense of the actual timescales involved, we can expect some bond vibrations to have a period of around 100 fs (where fs = femtosecond = 10^{-15} s) meaning that the timestep should be at least an order of magnitude smaller than this, with a common choice being $1-2$ fs. While the inclusion of constraint algorithms can allow for more freedom in the time discretisation, this still limits the scope of all-atom simulations either to processes that occur over picoseconds, nanoseconds, or for small molecules potentially microseconds. In practice, biological processes of interest often occur

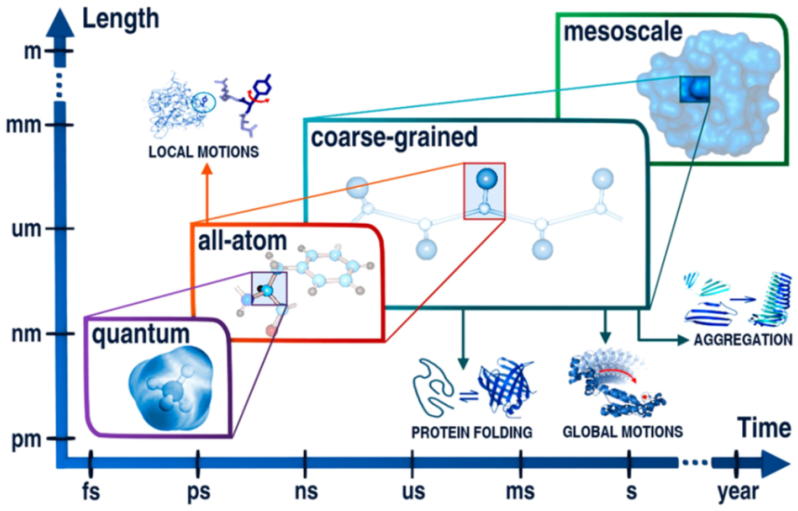


Figure 1.4: *Illustration of length and timescales involved at various detail levels of molecular dynamics modelling. By considering coarser descriptions of target molecules longer and larger simulations may be attempted for the same computational cost. Figure reproduced here with permission (ACS AuthorChoice license) from* [48]

over seconds, for huge systems of millions of atoms. To address this problem, there is much work in the literature on the topic of coarse-grained molecular dynamics. Some or all of a target molecule or solvent are modelled in a caricatured manner, with clumps of atoms replaced by single, coarse units [48–51]. By doing this we can model larger systems for longer periods of time opening up new avenues of research. A well known coarse-graining approach is to use the MARTINI force fields [50, 52], where structures are broken down into small groups of atoms, with each group represented by a single subunit. The properties of these new elements are set using thermodynamic properties depending on their underlying atomistic structure. The types of potential involved are often identical to those seen in equation (1.2) (with different parameters), but, as a result of having fewer subunits degrees of freedom, complexity is still reduced.

Trajectories from all-atom MD simulations can also be used to estimate the correct parameters for a coarse-grained model, with one of the most common methods being a Boltzmann inversion [53, 54]. If we think of the coarse-grained model as having coordinates associated with those of the all-atom model (e.g. a subunit

having its centre of mass placed at the most spatially central atom in the underlying structure) these can be tracked through a simulation. As a simple example, if we consider two subunits with positions $\boldsymbol{x}_1$, $\boldsymbol{x}_2 \in \mathbb{R}^3$. Over the course of a simulation it is possible to track their separation $R = |\boldsymbol{x}_2 - \boldsymbol{x}_1|$, and calculate the mean distance ℓ_0 and the variance β^2 over time. Assuming this gives a Gaussian distribution

$$p(R) = \frac{1}{\sqrt{2\pi\beta^2}} \exp\left(\frac{(R-\ell_0)^2}{2\beta^2}\right),$$

by carrying out a Boltzmann inversion we can find a harmonic potential

$$\Phi(R) = -k_B T \log(p(R)) = \frac{k(R-\ell_0)^2}{2}. \tag{1.3}$$

for some bond length ℓ_0, and spring constant k determined by the fluctuations of the system. We make use of this potential when binding two monomers together to form a dimer in Chapter 2. Additionally, it is used to bond ellipsoids representing actin together, with harmonic bonds attached to their surfaces, in Chapter 3. This can be extended into an iterative scheme to find a general potential that matches that of the atomistic model, applied to large systems there is room for incorrect potentials to be found which do not match the true reality, and more sophisticated methods have been developed to improve results by including target data from multiple phase states [55, 56]. Similarly, Savelyev et al. [57] present a coarse-grain representation of DNA, with each base pair represented by two subunits, representing an even greater reduction in complexity, with parameters for the model matched to data from an all-atom model of the same system via a Monte Carlo optimisation method involving the partition function [58]. While this still requires all atom simulations to be carried out, the size and length of these can be kept reasonably small, with the scope of investigation increased after the coarse-grained parameters are determined. Other methods instead match the fluctuation of the atomistic data to those of the coarse-grained model [59] to infer model parameters, where large sites or even entire proteins are modelled by single subunits [49, 60], and this level of reduction in resolution is sometimes referred to as ultra-coarse-graining [51].

In coarse-grained MD models, water is often treated with the same level of coarse-graining as other molecules in the system, as by losing the fine-grain details of the target molecule, it is likely that fine-grain solvent interactions will play a smaller role in the dynamics of a system. For example, four water molecules are combined into a single coarse-grained water bead in the Martini model [50], while Wat Four water model [61] uses four linked beads placed at the corners of a tetrahedron to collectively represent 11 water molecules. Parameters are often set by considering physical principles for the force fields, though machine learning approaches [62] can also be employed to infer these parameters from MD trajectories. The choice of coarse-grained solvent model can have just as profound effects on system dynamics as in traditional all atom alternatives, with a Bayesian model selection study [63] suggesting that electrostatic properties of the target molecule being investigated are among the most important factors to consider. However, there are situations where the individual solvent behaviour is not of interest, only its overall dynamic effect on the target molecules in solution. Rather than fully modelling the solvent, we can instead capture these forces using a stochastic Langevin model, where it is possible to replace the forces that would result from the potentials of the water molecules with an averaged, stochastic force, with motion dictated by the equations

$$\begin{aligned} \mathrm{d}\mathbf{X} &= \mathbf{V}\,\mathrm{d}t, \\ \mathrm{d}\mathbf{V} &= \frac{1}{M}\frac{\partial \Phi}{\partial \boldsymbol{X}} - \gamma \mathbf{V}\,\mathrm{d}t + \gamma\sqrt{2D}\,\mathrm{d}\mathbf{W}, \end{aligned} \tag{1.4}$$

where $\mathbf{X}, \mathbf{V} \in \mathbb{R}^3$ represent the position and velocity, $\Phi \in \mathbb{R}$ the potential energy, γ is the friction coefficient, and $\mathrm{d}\mathbf{W} \in \mathbb{R}^3$ is a Wiener process representing the stochastic Gaussian random noise. This can be seen as the simplest caricature description of single particle diffusion, with uncorrelated noise forces, unlike the generalised Langevin model and other non-Gaussian force distributions, which can improve consistency with fine-grain simulations [64]. Energy enters the system via the noise term in a non-conservative manner, but by including the dissipative friction term the simulations remain stable, with a natural capping on particle velocity allowing the temperature of a simulation to be controlled via model parameterisation, making this a form of thermostat. In Chapter 2 we will explore how stochastic models of

solvent interactions can be parameterised in a manner consistent with larger scale MD simulations. By using models introduced in [65] and extending on the work by Robert Zwanzig [66, 67], which demonstrate ways to derive Langevin dynamics from nonlinear systems, we use an adapted version of these long range heat baths models as a test case. Other key papers such as [68] discuss the Mori-Zwanzig formalism, where projection operators used to separate relevant aspects of the dynamics of system from noise terms, while [69] covers analytic details around the fluctuation-dissipation theorem, relating the memory effect of noise in a system to the friction kernel. More recent work in this area includes methods to directly calculate the form of the friction force from MD force and velocity trajectories using iterative [70] and non-iterative numerical schemes [71], with a higher order formulation of the non-iterative version presented. Langevin dynamics can be extended to dissipative particle dynamics for situations where there are multiple target particles, with extra terms included to capture the friction caused by these large particles on each other in conjunction with the solvent drag effects [72].

1.3 Multi-resolution and multiscale approaches in molecular dynamics

While coarse-graining a biomolecule can lead to improved scope for molecular dynamics simulations, the further from the physical reality we move, the more likely we are to miss subtle details. It is worth asking the question of whether it is possible to design models where we have the best of both worlds, with fine details in areas of biological and physical interest. Some of the most well known multiscale work resulted in the 2013 Nobel Prize in Chemistry being awarded to Levitt, Warshell, and Karplus [73, 74], who managed to make Newton's classical physics work side-by-side with the fundamentally different quantum physics. They designed simulations where the majority of a protein was simulated by the rules of classical mechanics, while the atoms in a binding site were described with more accurate quantum physics. Multi-resolution (hybrid) methods of this style aim to use detailed and coarse-grained simulations in different parts of the simulation

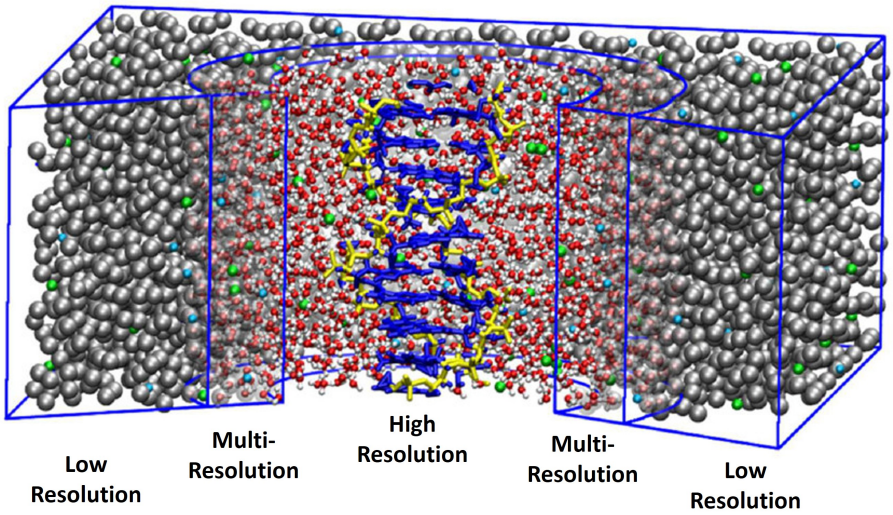


Figure 1.5: *Multi-resolution simulation of DNA molecule in a salt solution. Far away from the target molecule of interest the solvent is described coarsely with each water represented by single bead, and close to molecule it is described with the standard TIP3P all-atom water model, with a hybrid region at the boundary between the two. Figure reproduced here with edited labels with permission (Springer Nature license) from* [82]

domain during the same dynamic simulation [75–78]. Such methods have been developed in different application areas and at different spatial and temporal scales in the literature, including dual-resolution approaches AdResS and H-AdResS for all-atom MD simulations [79–83], where the solvent in a simulation is resolved at different levels of detail in different spatial regions, with an example seen in Figure 1.5. In some multi-resolution MD approaches, the region of high resolution moves together with the large microscopic structure of interest so that the high resolution model is always used for the whole considered structure, which can range in size from a single biomolecule (a protein or a DNA in solution [81, 82]) to virus-like particles [84, 85]. The structure of interest is placed in the centre of the simulation domain and it is solvated using a detailed atomistic MD water in its immediate neighbourhood, which is coupled with a coarse-grained water description in the rest of the computational domain. In Chapter 2 we consider a theoretical short-range interaction heat bath with a co-moving frame that operates in a similar manner to this, along with a long-range interaction bath where the solvent particles naturally move with the target particle. Techniques have also been developed for coupling Brownian dynamics approaches with lattice-based stochastic reaction-diffusion

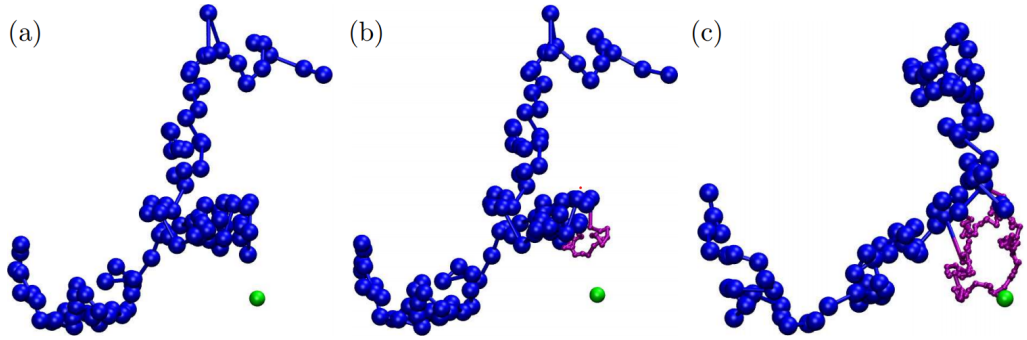


Figure 1.6: *Coarse-grain multi-resolution model of a DNA chain. DNA molecules are modelled using a coarse bead-spring polymer The resolution of model is adapted on the fly as a protein (green) enters withing a certain radius of the polymer. The low resolution chain (blue) is replaced with high-resolution (purple) chain, which can be thought of as the underlying reality. Figure reproduced here with permission (Creative Commons attribution license) from* [95].

models [86–88] in addition to methods which make use of continuum mean-field equations for the macroscopic component of the simulation [89–91]. There have also been studies where the atomistic detail of the active part of an enzyme has been coupled with a coarser model of the rest of the protein [92], as well as in the modelling of polymer melts [93, 94]. In a similar manner, as seen in Figure 1.6, a multi-resolution approach has been utilised for a coarse-grain bead-spring model of DNA [95], with the resolution of the model adapted on-the-fly in response to a binding protein entering withing a certain radius of the polymer. In [96] the size of timestep is also varied across different regions of the model. Areas of detail are updated more frequently than the coarser regions, which would reflect the differences in stability; fine-grain models generally require smaller timesteps than coarse models, so taking extra steps in this region compounds the computational advantage gained. In Chapter 3 we will present a multi-resolution model of an F-actin filament in a manner similar to this, where the detailed region is modelled by a coarse-grained representation of the filaments set out in De La Cruz et al. [97], and the remainder is modelled by a Cosserat rod model [98].

1.4 Modelling of actin and the cytoskeleton

This section will give the biological and modelling background for much of the work in Chapter 3 and Chapter 4, where we will study the physical properties of actin using coarse-grained, multi-resolution models. Actin is a globular protein that forms microfilaments which can be found in almost all eukaryotic cells [99]. It exists either as free G-actin monomers, or in F-actin filaments with the monomers forming thin, stiff double helical strands [28]. Actin filaments are found in the cellular cytoskeleton in structures like filopodia and lamellipodia, playing a role in many important processes, such as cell migration and muscle contraction via actomyosin networks. In these settings we find actin interacting with numerous binding proteins and cross-linkers to give the networks the biological structure required to perform their function [100].

As a result of its importance and varied role in these situations, the behaviour of actin has been investigated at many biological length and time scales. Detailed all-atom molecular dynamics (MD) simulations can be parametrised using data obtained via X-ray crystallography [28] or cryogenic electron microscopy [29, 30] available in repositories such as the RCSB Protein Data Bank [31]. These simulations capture the fine chemical bonding structure of actin and bound ligands, both in monomer and filamentous forms [101–105], though this detail comes at a high computational cost. Atomistic simulations can be used to investigate the physical properties of short F-actin sections [106], but it is infeasible to use this level of detail for large systems of biological interest. When larger, more complex systems are involved, software such as MEDYAN [107, 108] (as seen in Figure 1.7) or Cytosim [109, 110] allow for the mechanochemical modelling of large cytoskeletal actin networks with crosslinkers and binding proteins included in simulations. In models of this style, actin is often described as a simple cylindrical filament, with simple angle and harmonic bond potentials similar to the first three terms of equation (1.2), with the detail of the chemical structure and double helix lost as a compromise for efficiency. Alternatively, a more detailed continuum description can be used [111, 112] incorporating new mechanical properties of the filaments,

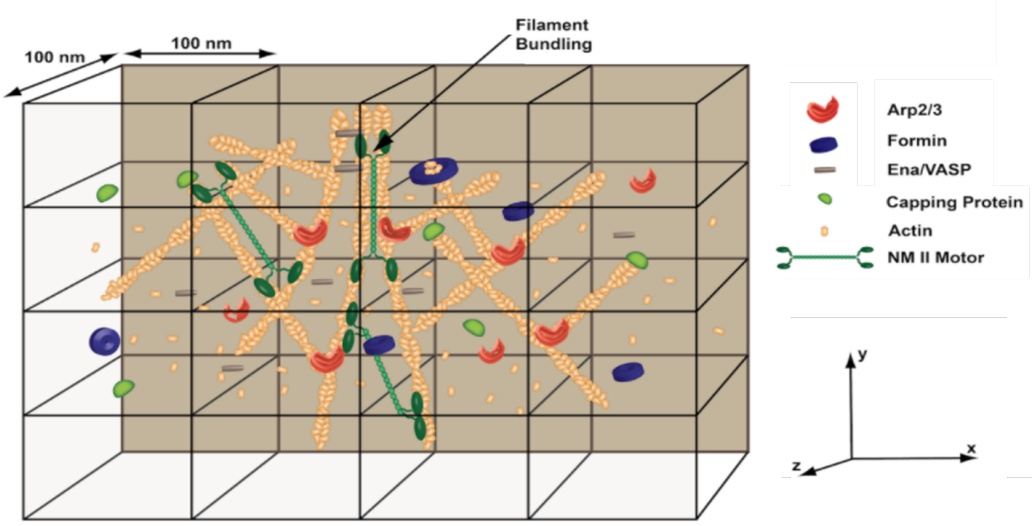


Figure 1.7: *Schematic of various biological molecules involved in large scale cytoskeleton simulations, using software package MEDYAN [107]. Here we see a network of actin filaments bound to myosin motors, capping proteins, and other biomolecules, achieving a level of complexity and scale that is practically impossible using all-atom MD simulations; it would only be able to evaluate a MD simulation of this size for a timespan that is biologically irrelevant. Figure reproduced here with permission (Creative Commons attribution license) from* [107].

such as torsion and bend-twist couplings [98, 113], allowing large scale structural properties of the networks to be captured, in addition to some of features that arise from the double helical nature of actin.

In Chapter 3 we will use a coarse Cosserat rod representation of actin, where torsional, shearing, stretching and bending properties of actin are set from values measured from both physical experiments and MD simulation, allowing parametrisation decisions to be informed by large and small scale investigations. A filament is described by its centre-line $\boldsymbol{r}(s,t) \in \mathbb{R}^3$ with velocity $\boldsymbol{v}(s,t) \in \mathbb{R}^3$, where $t \in \mathbb{R}$ is time and $s \in \mathbb{R}$ is the centre-line arc coordinate. We also define an orientated frame of reference given by vectors $\boldsymbol{d}_1(s,t)$, $\boldsymbol{d}_2(s,t)$, $\boldsymbol{d}_3(s,t) \in \mathbb{R}^3$, which form an orthonormal basis of unit vectors, and the corresponding matrix $\boldsymbol{Q}(s,t) = \Big(\boldsymbol{d}_1(s,t), \boldsymbol{d}_2(s,t), \boldsymbol{d}_3(s,t)\Big)$. The $\boldsymbol{d}_3(s,t)$ basis vector is tangential to the filament centreline We define the laboratory and body-convected frames as $\boldsymbol{x} = \widehat{x_1}\,\widehat{\boldsymbol{d}}_1 + \widehat{x_2}\,\widehat{\boldsymbol{d}}_2 + \widehat{x_3}\,\widehat{\boldsymbol{d}}_3$ and $\boldsymbol{x}_{\mathcal{L}} = \boldsymbol{Q}^T\boldsymbol{x} = x_1\boldsymbol{d}_1 + x_2\boldsymbol{d}_2 + x_3\boldsymbol{d}_3$, respectively, where the laboratory reference basis is denoted $\Big\{\widehat{\boldsymbol{d}}_1, \widehat{\boldsymbol{d}}_2, \widehat{\boldsymbol{d}}_3\Big\}$. The variation of

this matrix $\boldsymbol{Q}$ in space (along the filament length) and in time can be used to formally consider the curvature and angular momentum. We define the generalised curvature and rod angular velocity as

$$\boldsymbol{\kappa} = \mathrm{ax}\left[\frac{\partial \boldsymbol{Q}}{\partial s}\boldsymbol{Q}^T\right], \qquad \boldsymbol{\omega} = \mathrm{ax}\left[\frac{\partial \boldsymbol{Q}}{\partial t}\boldsymbol{Q}^T\right],$$

where the ax[·] operation returns the pseudovector associated with the skew-symmetric matrix $\boldsymbol{U} \in \mathbb{R}^{3\times 3}$ using the following relations holding

$$\boldsymbol{U} = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} \qquad \mathrm{ax}(\boldsymbol{U}) = \boldsymbol{u} = (a, b, c)^T,$$

where $a, b, c \in \mathbb{R}$. Its inverse operation, skew[·], acts on $\boldsymbol{u} \in \mathbb{R}^3$ such that $\mathrm{skew}(\boldsymbol{u}) = \boldsymbol{U}$. These have the property that for any vector $\boldsymbol{x}$, we have $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{u} \times \boldsymbol{x}$. These expressions for curvature and angular velocity can be derived by considering a simple example with arbitrary vector $\boldsymbol{x}_0$, and letting $\boldsymbol{x}(t) = \boldsymbol{Q}\boldsymbol{x}_0$. Differentiating this gives $\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{Q}}{\mathrm{d}t}\boldsymbol{x}_0 = \frac{\mathrm{d}\boldsymbol{Q}}{\mathrm{d}t}\boldsymbol{Q}^T\boldsymbol{x}(t)$. As $\boldsymbol{Q}$ is a rotation matrix, $|\boldsymbol{x}(t)| = |\boldsymbol{x}_0|$, the tip of these vectors move tangentially on the surface of a circle, so the derivative can be expressed in terms of the angular velocity as $\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{\omega} \times \boldsymbol{x}(t) = \frac{\mathrm{d}\boldsymbol{Q}}{\mathrm{d}t}\boldsymbol{Q}^T\boldsymbol{x}(t)$. As we chose a vector arbitrarily, our expression for $\boldsymbol{\omega}$ holds, with similar logical applicable for $\boldsymbol{\kappa}$.

Using [114] it is possible to find the following equations of motion for unshearable, inextensible filaments

$$\frac{\partial \boldsymbol{r}}{\partial t} = \boldsymbol{v}, \tag{1.5}$$

$$\frac{\partial \boldsymbol{d_j}}{\partial t} = \boldsymbol{\omega} \times \boldsymbol{d_j}, \tag{1.6}$$

$$\frac{\partial(\rho A \boldsymbol{v})}{\partial t} = \frac{\partial \boldsymbol{n}}{\partial s} + \boldsymbol{F}, \tag{1.7}$$

$$\frac{\partial(\rho \boldsymbol{I}\boldsymbol{\omega}_{\mathcal{L}})}{\partial t} = \frac{\partial \boldsymbol{\tau}_{\mathcal{L}}}{\partial s} + \boldsymbol{\kappa}_{\mathcal{L}} \times \boldsymbol{\tau}_{\mathcal{L}} + \boldsymbol{Q}^T\left(\frac{\partial \boldsymbol{x}}{\partial s} \times \boldsymbol{n}\right) + (\rho \boldsymbol{I}\boldsymbol{\omega}_{\mathcal{L}}) \times \boldsymbol{\omega}_{\mathcal{L}} + \boldsymbol{C}_{\mathcal{L}}\,, \tag{1.8}$$

where ρ is the material density, A is the cross-sectional area of the filament, $\boldsymbol{n}_{\mathcal{L}} \in \mathbb{R}^3$ is the internal force resultant, $\boldsymbol{\tau}_{\mathcal{L}} \in \mathbb{R}^3$ the torque resultant, and $\boldsymbol{I} \in \mathbb{R}^{3\times 3}$ the second mass moment of inertia. We can write the following torque-curvature relation

$$\boldsymbol{\tau}_{\mathcal{L}} = \boldsymbol{B}(\boldsymbol{\kappa}_{\mathcal{L}} - \boldsymbol{\kappa}^0_{\mathcal{L}}),$$

where $\boldsymbol{B} \in \mathbb{R}^{3\times3}$ is the shear/stretch stiffness matrix, and $\boldsymbol{\kappa}^0_{\mathcal{L}} \in \mathbb{R}^3$ is the rest curvature of the filament, where $\boldsymbol{\kappa}^0_{\mathcal{L}} = \mathbf{0}$ would signify that the filaments are naturally straight and untwisted. When the rods are unshearable and inextensible, $\boldsymbol{d}_3$ is aligned with the centreline of the segment, though when shear and extension are allowed, the deviation $\boldsymbol{d}_3$ from the centreline can be formulated as a measure of this. We define the strain formally as

$$\boldsymbol{\sigma}_{\mathcal{L}} = \boldsymbol{Q}^T \left(\frac{\partial \boldsymbol{r}}{\partial s} - \boldsymbol{d}_3 \right) = \boldsymbol{Q}^T \left(e\boldsymbol{t} - \boldsymbol{d}_3 \right), \quad e(s,t) = \frac{\mathrm{d}s}{\mathrm{d}\hat{s}},$$

where e is the local stretching/compression relative to the rest arc length coordinate $\hat{s}$, and $\boldsymbol{t}$ is the unit tangent vector. We can also write the following load-strain relation in a similar manner to the torque-curvature relations

$$\boldsymbol{n}_{\mathcal{L}} = \boldsymbol{S}(\boldsymbol{\sigma}_{\mathcal{L}} - \boldsymbol{\sigma}^0_{\mathcal{L}}),$$

where $\boldsymbol{S} \in \mathbb{R}^{3\times3}$ is the shear/stretch stiffness matrix, and $\boldsymbol{\sigma}^0_{\mathcal{L}} \in \mathbb{R}^3$ is the default strain. We set $\boldsymbol{\kappa}^0_{\mathcal{L}} = 0$ and $\boldsymbol{\sigma}^0_{\mathcal{L}} = 0$ for the remainder of this work.

For the spatial discretisation, we split the filament into n segments, with each segment defined by their endpoints $\boldsymbol{r}_i(t) \in \mathbb{R}^3$ for $i = 1, 2, \ldots, n+1$. Additional quantities associated with the endpoints (their velocity $\boldsymbol{v}_i(t)$ and applied force $\boldsymbol{F}_i(t) \in \mathbb{R}^3$) have $n+1$ elements, while discretised quantities associated with the segments have n elements. We define the segment vector, deviation from its rest length (i.e. the local stretching/compression) and normal tangent as

$$\boldsymbol{l}_i = \boldsymbol{r}_{i+1} - \boldsymbol{r}_i\,, \qquad e_i = \frac{|\boldsymbol{l}_i|}{|\hat{\boldsymbol{l}}_i|}\,, \qquad \boldsymbol{t}_i = \frac{\boldsymbol{l}_i}{|\boldsymbol{l}_i|}, \qquad \text{for } i = 1, 2, \ldots, n,$$

where $|\hat{\boldsymbol{l}}_i|$ is the rest length of the segment, and for all following definitions the hat notation indicates a rest, un-stretched quantity. In the continuum setting all quantities can be defined point-wise, whereas in a discrete setting some quantities (such as $\hat{\boldsymbol{\kappa}}_{\mathcal{L}}$) are naturally expressed over an integrated domain along the filament [115], with the Voronoi region, and associated compression factor defined as

$$\mathcal{D}_i = \frac{|\boldsymbol{l}_{i+1}| + |\boldsymbol{l}_i|}{2} \qquad \text{and} \qquad \mathcal{E}_i = \frac{\mathcal{D}_i}{\hat{\mathcal{D}}_i}, \qquad \text{for } i = 1, 2, \ldots, n-1.$$

As the generalised curvature expresses a rotation per unit length, the quantity $\hat{\mathcal{D}}_i\hat{\boldsymbol{\kappa}}^i_{\mathcal{L}}$ expresses the rotation that transforms a material frame to its neighbouring frame over the segment, therefore

$$\exp\left(\hat{\mathcal{D}}_i\hat{\boldsymbol{\kappa}}^i_{\mathcal{L}}\right)\boldsymbol{Q}_i^T = \boldsymbol{Q}_{i+1}^T.$$

Having set out these quantities, we can now define the discrete shear and curvature vectors as

$$\begin{aligned}\boldsymbol{\sigma}^i_{\mathcal{L}} &= \boldsymbol{Q}_i^T\left(e_i\boldsymbol{t}_i - \boldsymbol{d}_{3,i}\right), && \text{for } i = 1, 2, \ldots, n,\\ \hat{\boldsymbol{\kappa}}^i_{\mathcal{L}} &= \frac{\log\left(\boldsymbol{Q}_{i+1}^T\boldsymbol{Q}_i\right)}{\hat{\mathcal{D}}_i}, && \text{for } i = 1, 2, \ldots, n-1.\end{aligned}$$

We can finally write the full spatially discretised expressions for equations (1.7) and equations (1.8)

$$m_i\frac{\partial \boldsymbol{v}_i}{\partial t} = \Delta_i\left(\left\{\frac{\boldsymbol{Q}_j\hat{\boldsymbol{S}}_j\boldsymbol{\sigma}^j_{\mathcal{L}}}{e_j}\right\}\right) + \boldsymbol{F}_i, \qquad \text{for } i = 1, 2, \ldots, n+1. \tag{1.9}$$

$$\begin{aligned}\left(\frac{\hat{\boldsymbol{I}}_i}{e_i}\right)\frac{\partial \boldsymbol{\omega}^i_{\mathcal{L}}}{\partial t} &= \Delta_i\left(\left\{\frac{\hat{\boldsymbol{B}}_j\hat{\boldsymbol{\kappa}}^j_{\mathcal{L}}}{\mathcal{E}_j^3}\right\}\right) + \mathcal{A}_i\left(\left\{\frac{\hat{\boldsymbol{\kappa}}^j_{\mathcal{L}}\times\hat{\boldsymbol{B}}_j\hat{\boldsymbol{\kappa}}^j_{\mathcal{L}}}{\mathcal{E}_j^3}\hat{\mathcal{D}}_j\right\}\right)\\ &\quad + \left(\boldsymbol{Q}_i^T\boldsymbol{t}_i\times\hat{\boldsymbol{S}}_i\boldsymbol{\sigma}^i_{\mathcal{L}}\right)|\hat{\boldsymbol{l}}_i| + \frac{\hat{\boldsymbol{I}}_i\boldsymbol{\omega}^i_{\mathcal{L}}}{e_i^2}\frac{\partial e_i}{\partial t}\\ &\quad + \left(\frac{\hat{\boldsymbol{I}}_i\boldsymbol{\omega}^i_{\mathcal{L}}}{e_i}\right)\times\boldsymbol{\omega}^i_{\mathcal{L}} + \boldsymbol{C}^i_{\mathcal{L}}, \qquad \text{for } i = 1, 2, \ldots, n,\end{aligned} \tag{1.10}$$

where $m_i = \rho\,\mathcal{A}_i(\{|\hat{\boldsymbol{l}}_j|\})A$ is the mass concentrated at $\boldsymbol{r}_i$, $\hat{\boldsymbol{I}}_i$ is the inertia tensor, $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{S}}$ the bend/twist and shear/stretch rigidity matrices, and $\boldsymbol{F}_i$ and $\boldsymbol{C}^i_{\mathcal{L}}$. The difference operator $\Delta : \mathbb{R}^{k\times 3} \to \mathbb{R}^3$ and trapezoidal quadrature operator $\mathcal{A} : \mathbb{R}^{k\times 3} \to \mathbb{R}^3$ used are defined for a set of vectors $\{\boldsymbol{x}_j\} \equiv \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$ as

$$\Delta_i(\{\boldsymbol{x}_j\}) = \begin{cases}\boldsymbol{x}_1, & \text{if } i = 1,\\ \boldsymbol{x}_i - \boldsymbol{x}_{i-1}, & \text{if } 1 < i < k,\\ -\boldsymbol{x}_N, & \text{if } i = k.\end{cases}$$

$$\mathcal{A}_i(\{\boldsymbol{x}_j\}) = \begin{cases}\boldsymbol{x}_1/2, & \text{if } i = 1,\\ (\boldsymbol{x}_i + \boldsymbol{x}_{i-1})/2, & \text{if } 1 < i < k,\\ \boldsymbol{x}_i/2, & \text{if } i = k.\end{cases}$$

The discretisation for equations (1.5) and equations (1.6) will be given in detail in Section 4.1.2 when describing numerical integration methods. When calculating

potential energy, we integrate over the length of the filament for both the twist/bend and shear/stretch terms

$$E = \frac{1}{2}\int_0^L \boldsymbol{\kappa}_{\mathcal{L}}^T \boldsymbol{B}\boldsymbol{\kappa}_{\mathcal{L}}\,\mathrm{d}s + \frac{1}{2}\int_0^L \boldsymbol{\sigma}_{\mathcal{L}}^T \boldsymbol{S}\boldsymbol{\sigma}_{\mathcal{L}}\,\mathrm{d}s. \tag{1.11}$$

This treatment of the Cosserat model is largely derived from [98], though in Chapter 3 we will discuss extensions which may be made to improve its ability to capture the physical anisotropies of F-actin. Other applications of rod based models of this style exist, with inextensible Kirchhoff models used to study the behaviour of surgical thread [116], or even modelling filaments as being viscous and capturing the dynamics of hair [117].

Coarse-grained models of actin monomers offer a significant reduction in computational cost when compared to all-atom simulations. Actin monomers, either in free or bound filament form, can be represented as spheres or ellipsoids [49, 118, 119], in a more detailed 4-bead description with the sites mapped to its subdomains [60, 106], or an even greater level of detail that begins to bridge the gap in resolution to all-atom simulations [120, 121]. Parameters for these coarse-grained models can be set either from experimental data or by inferring them, using various techniques [57, 60], from all-atom MD simulation data. [122, 123]. In Chapter 3 we make use of the model first presented in De La Cruz et al. [97] and adapted for the various applications mentioned previously [49, 119] as a finer representation of actin to couple to the Cosserat rod model. The actin filament is constructed out of ellipsoid shaped monomers placed along a double helix, as is schematically shown in Figure 1.8. The dimensions and spatial positions of these monomers are taken from crystal structures of F-actin filaments. The centre of mass of the i^{th} monomer is denoted by $\bar{\boldsymbol{R}}_i \in \mathbb{R}^3$, where $i = 1, 2, \ldots, N$, and N is the total number of monomers. Each monomer is connected by harmonic bonds to its two nearest neighbours up and down the chain, if they exist. That is, the i^{th} monomer is connected by $n_{\parallel}$ harmonic bonds to the next and previous monomer in their strand of the double helix, which are labelled as the $(i+2)^{\text{th}}$ and $(i-2)^{\text{th}}$ monomers. Moreover, the i^{th} monomer is also connected by $n_{\perp}$ harmonic bonds to each of the adjacent monomers

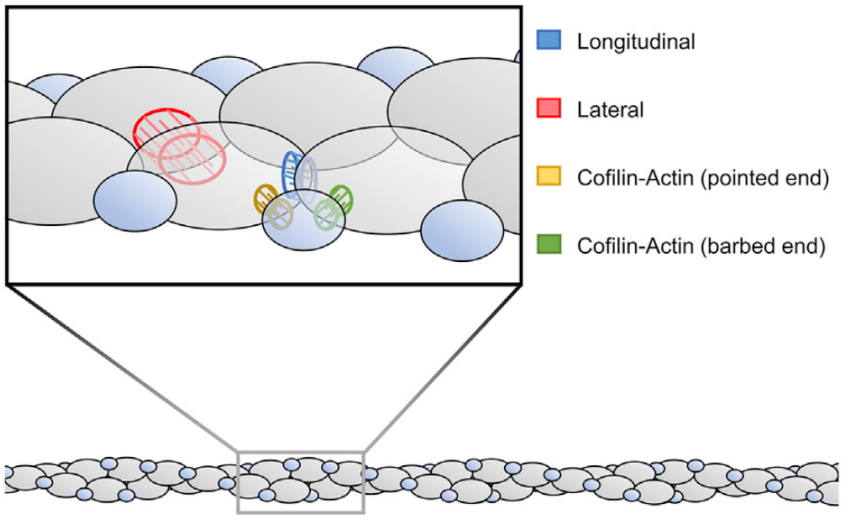


Figure 1.8: *Coarse-grained representation of cofilin decorated actin filament, with each G-actin monomer represented by a single ellipsoid and connected to others via harmonic springs attached to its surface. The stiffness of these springs is determined by matching the parameters directly with all-atom MD simulation data and corresponds well to experimentally expected values. Figure reproduced here with permission (Elsevier license) from* [119]

in the offset strand of the double helix, which are labelled as the $(i+1)^{\text{th}}$ and $(i-1)^{\text{th}}$ monomers. The monomers are connected to each other by harmonic bonds, with the end points of these placed on the surfaces of the ellipsoids. This allows for both the distance between monomer centres and their relative angular orientations to be constrained. Considering the k^{th} bond between the i^{th} and j^{th} monomers, where $i = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, n_\perp$ for $j = i \pm 1$ or $k = 1, 2, \ldots, n_\parallel$ for $j = i \pm 2$, we denote by $\boldsymbol{r}_i^k$, $\boldsymbol{r}_j^k \in \mathbb{R}^3$ the end points of the k^{th} bond on the surface of the i^{th} and j^{th} monomers, respectively. Then we define the force at $\boldsymbol{r}_i^k$, resulting from the k^{th} bond between the i^{th} and j^{th} monomers, as

$$\boldsymbol{F}_{ij}^k = S_{ij}\left(\boldsymbol{r}_j^k - \boldsymbol{r}_i^k\right)\left(1 - \frac{L_{ij}^k}{|\boldsymbol{r}_j^k - \boldsymbol{r}_i^k|}\right), \qquad \text{for} \quad |i-j| = 1, 2, \tag{1.12}$$

where the spring constants S_{ij} take different values for lateral and longitudinal bond interactions, namely

$$S_{ij} = \begin{cases} S_\perp/n_\perp, & \text{for } |i-j| = 1, \\ S_\parallel/n_\parallel, & \text{for } |i-j| = 2, \end{cases}$$

where $S_\parallel$ and $S_\perp$ are the total strains, and L_{ij}^k in equation (1.12) is the equilibrium length of the k^{th} bond between the i^{th} and j^{th} monomers. By summing up

forces (1.12), we can define the overall translational and torsional effects on the i^{th} monomer, relative to the centre of mass of this ellipsoid $\bar{\boldsymbol{R}}_i$ as follows:

$$\bar{\boldsymbol{F}}_i = \sum_{j,k} \boldsymbol{F}_{ij}^k \,, \qquad \text{and} \qquad \bar{\boldsymbol{\tau}}_i = \sum_{j,k} (\boldsymbol{r}_i^k - \bar{\boldsymbol{R}}_i) \times \boldsymbol{F}_{ij}^k \,, \qquad \text{for } i = 1, 2, \ldots, N \,,$$

where we sum over all bonds between the i^{th} monomer and each of the monomers that it is bonded to. Equations of motion are written in terms of positions $\boldsymbol{r}_i^k$ and $\bar{\boldsymbol{R}}_i$ as follows

$$\frac{\mathrm{d}\boldsymbol{r}_i^k}{\mathrm{d}t} = \boldsymbol{\omega}_i \times (\boldsymbol{r}_i^k - \bar{\boldsymbol{R}}_i) + \boldsymbol{V}_i \,, \qquad \frac{\mathrm{d}\bar{\boldsymbol{R}}_i}{\mathrm{d}t} = \boldsymbol{V}_i \,, \tag{1.13}$$

$$M\frac{\mathrm{d}\boldsymbol{V}_i}{\mathrm{d}t} = \bar{\boldsymbol{F}}_i \,, \qquad \boldsymbol{I}\frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = (\boldsymbol{I}\boldsymbol{\omega}_i) \times \boldsymbol{\omega}_i + \bar{\boldsymbol{\tau}}_i \,, \tag{1.14}$$

where $\boldsymbol{V}_i$, $\boldsymbol{\omega}_i \in \mathbb{R}^3$ are the velocity and angular velocity of the i^{th} monomer, respectively, M is the mass of a monomer, and $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ is the moment of inertia tensor. This is equivalent to the model presented in De La Cruz et al. [97], though we also include a rigid body term $(\boldsymbol{I}\boldsymbol{\omega}_i) \times \boldsymbol{\omega}_i$ for consistency with the Cosserat implementation previously described. In Chapters 3–4, details of the exact methods we use to numerically integrate these model are provided, along with how to physically couple them and match their parameters to for a multi-resolution of F-actin.

1.5 Structure of thesis

The substantive work for this thesis begins in Chapter 2, where we cover an initial multi-resolution approach to modelling the solvent interaction for a dimer molecule, based on my work published as R. S. Gunaratne et al. "Multi-resolution dimer models in heat baths with short-range and long-range interactions", *Interface Focus* 9.3, 2019 [1], in addition to some further work on generalised Langevin equations as a means of coarse-graining solvent interactions. The latter half of the thesis (Chapters 3 and 4) is concerned with multi-resolution models of the target molecule, rather than solvent interaction. In Chapter 3 we introduce a multi-resolution model of F-actin, based on my work submitted for publication as R. S. Gunaratne et

al. "On Stretching, Bending, Shearing and Twisting of Actin Filaments II: Multi-Resolution Modelling" [2], and accompanying paper C. Floyd et al. "On Stretching, Bending, Shearing and Twisting of Actin Filaments I: Variational Models" [3]. Some of the filament regions are modelled in detail with spatial representation of the double helical nature of the filaments, while others are more coarsely represented by a continuum rod model approximation. We first formulate the coupling region connecting the two models, then explain how the models are validated against experimental data, parameterised, and matched to have consistent macroscopic behaviour. In Chapter 4, we present simulation results of simple example cases that have some biological relevance to demonstrate the performance of our novel multi-resolution model of actin, along with extensions to existing rod-based models of actin which better capture twisting anisotropies. We conclude in Chapter 5, and discuss future extensions and applications of the work.

2

Multi-Resolution Molecular Models with Langevin Dynamics

In this chapter we first approach multi-resolution modelling by considering how to capture the interactions between solvent molecules and a target molecule we are investigating. In essence, the work that follows will try to provide mathematical justification for choices of friction coefficient γ and diffusion constant D for the Langevin equation (1.4), based on the form of the solvent-particles interaction equations. As we are mainly concerned with capturing the solvent behaviour and interactions, we restrict our focus to simple target particles, making the problems more mathematically tractable. The challenges encountered when tackling more detailed (and consequently more biologically/physically realistic) potential energies will be discussed later in Chapters 3 and 4. Section 2.1 is based on the work carried out in R. S. Gunaratne et al. "Multi-resolution dimer models in heat baths with short-range and long-range interactions", *Interface Focus* 9.3, 2019 [1] (in which all figures featured in the section appear) where, in a simple example, the macroscopic behaviour of a target molecule can be captured consistently, despite the solvent interactions for different parts of this molecule being modelled using Langevin or Hamiltonian dynamics, respectively. In Section 2.2, we extend on these ideas by carrying out analysis on generalised Langevin models, allowing

more complex solvent behaviour to be described, using a long-range Hamiltonian interaction model as an example.

2.1 Multi-resolution dimer model

In this section, we first introduce a multi-resolution approach using a simple dimer model consisting of two monomers (beads) connected by a spring. Similar models, where a macromolecule is described as several beads, representing parts of the simulated biomolecule, connected by springs, have been obtained in the literature using the method of ultra-coarse-graining [51]. Thus our dimer model can be considered as a caricature of an ultra-coarse-grained model of a macromolecule. We study its behaviour in two theoretical heat baths (large systems with essentially constant temperature), which enables more analytical progress than solvent models based on all-atom or coarse-grained water models. In both cases, the convergence to the Langevin description of the solute particle can be established in a certain limit [65, 67, 124–127]. Our investigation focuses on multi-resolution (multiscale) descriptions of the solvent which can be described at the microscopic level of individual solvent molecules or at the macroscopic (dimer) level with the introduction of extrinsic random thermal forces on the monomers. We present models of the same dimer with various multi-resolution descriptions for the solvent and highlight the conditions and reasons, when and why, different model approximations of the solvent may be made in simulations.

The section is organised as follows. In Section 2.1.1, we introduce the macroscopic dimer model with a macroscopic description for solvent forces. This macroscopic model is fully described by Langevin equations. The Langevin macroscopic model is commonly used in simulation due to ease of implementation and analysis. We discuss in Section 2.1.1 the properties of this description with the intent to use these properties and corresponding Boltzmann statistics as benchmarks against which to compare microscopic and multi-resolution solvent models for the same dimer. Two theoretical microscopic approaches to model the solvent are introduced and studied through multi-resolution (simultaneous microscopic and macroscopic

coupled) modelling in Sections 2.1.2 and 2.1.3. One of them is based on (very) short-range interactions, as heat bath particles only interact with the dimer on contact. The other one is at the opposite extreme, as it is based on (very) long-range interactions, where the heat bath is modelled as a system of many harmonic oscillators.

2.1.1 The dimer model

In this section we will talk exclusively about the construction of the model for the dimer which will be used throughout Section 2.1. In doing so, we describe the solvent at the macroscopic level as an extrinsically added random force. The result will be a set of Langevin equations. Throughout the section we will modify the treatment of the solvent forces at various scales and hybrid resolutions but the underlying dimer model will be the same.

We consider a model of a dimer which is described by positions of its two monomers, denoted by $\mathbf{X}_1 = [X_{1;1}, X_{1;2}, X_{1;3}]$ and $\mathbf{X}_2 = [X_{2;1}, X_{2;2}, X_{2;3}]$, respectively, so we have $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^3$. Each monomer has the same mass, M. We denote by $\mathbf{R}$ the vector describing the separation between the monomers, i.e. $\mathbf{R} = \mathbf{X}_2 - \mathbf{X}_1$, and by R its magnitude $R = |\mathbf{R}|$. The interaction between monomers is given in terms of the potential $\Phi \equiv \Phi(R) : [0, \infty) \to \mathbb{R}$, which generates a force on each of the monomers with magnitude $\Phi'(R) = \frac{\partial \Phi}{\partial R}$.

When the dimer is placed into a heat bath, there are additional forces on the two monomers caused by interactions with solvent molecules.

We discussed a number of ways to model these interaction in Chapter 1 such as using fine-grain water models [13], along with coarser representation of solvent particles [50]. In this section, we consider two classes of models to describe the solvent-dimer interactions. The first, presented in Section 2.1.2, models the solvent as a bath of point particles which collide with the monomers and elastic collisions (short-range interactions) contribute to the generation of the forces. In the second case, described in Section 2.1.3, solvent molecules are point particles which oscillate around and interact at a distance (through long-range interactions) with the monomers. The

solvent-dimer interactions are the sum of harmonic oscillatory forces acting on each of the monomers. Importantly, both descriptions under suitable assumptions lead to a macroscopic description of the dimer given by the following set of Langevin equations

$$\begin{aligned} \mathrm{d}\mathbf{X}_1 &= \mathbf{V}_1\,\mathrm{d}t, && (2.1)\\ \mathrm{d}\mathbf{V}_1 &= \frac{\Phi'(R)}{M}\frac{\mathbf{R}}{R}\,\mathrm{d}t - \gamma\mathbf{V}_1\,\mathrm{d}t + \gamma\sqrt{2D}\,\mathrm{d}\mathbf{W}_1, && (2.2)\\ \mathrm{d}\mathbf{X}_2 &= \mathbf{V}_2\,\mathrm{d}t, && (2.3)\\ \mathrm{d}\mathbf{V}_2 &= -\frac{\Phi'(R)}{M}\frac{\mathbf{R}}{R}\,\mathrm{d}t - \gamma\mathbf{V}_2\,\mathrm{d}t + \gamma\sqrt{2D}\,\mathrm{d}\mathbf{W}_2, && (2.4) \end{aligned}$$

where $\mathbf{V}_1 = [V_{1;1}, V_{1;2}, V_{1;3}]$ and $\mathbf{V}_2 = [V_{2;1}, V_{2;2}, V_{2;3}]$ are velocities of the first and second monomer, respectively, $\mathbf{W}_1$ and $\mathbf{W}_2$ are three-dimensional vectors of independent Wiener processes, D is a constant diffusion coefficient and γ is a friction coefficient, with dimension $[\gamma] = [\text{time}]^{-1}$.

System (2.1)–(2.4) provides a macroscopic model of the dimer, which we compare with microscopic (or multi-resolution) MD simulations which explicitly model the solvent. Its validity for different MD models can be tested by comparing values of different dimer's statistics at equilibrium, including its expected length L_d, dimer velocity autocorrelation function $C_d(\tau)$ (which varies in time τ) and dimer diffusion constant D_d, defined by

$$\begin{aligned} L_d &= \lim_{t\to\infty}\langle R\rangle\,, \\ C_d(\tau) &= \lim_{t\to\infty}\frac{1}{3}\left\langle \overline{\mathbf{V}}(t+\tau)\cdot\overline{\mathbf{V}}(t)\right\rangle, \qquad (2.5)\\ D_d &= \lim_{t\to\infty}\frac{1}{6t}\left\langle \left(\overline{\mathbf{X}}(t)-\overline{\mathbf{X}}(0)\right)^2\right\rangle, \end{aligned}$$

where $\overline{\mathbf{X}} = (\mathbf{X}_1 + \mathbf{X}_2)/2$ is the centre of mass of the dimer and $\overline{\mathbf{V}} = (\mathbf{V}_1 + \mathbf{V}_2)/2$ is its velocity. These quantities can be obtained analytically for our macroscopic model (2.1)–(2.4) as follows. Adding equation (2.2) and equation (2.4) and noting that the sum of two independent Wiener processes is another Wiener processes $\mathbf{W}$ with an infinitesimal variance which is the sum of the variances of the original two processes, we obtain an Ornstein-Uhlenbeck process for $\overline{\mathbf{V}}$ in the following form

$$\mathrm{d}\overline{\mathbf{V}} = -\gamma\overline{\mathbf{V}}\,\mathrm{d}t + \gamma\sqrt{D}\,\mathrm{d}\mathbf{W}.$$

Therefore, we have

$$C_d(\tau) = \frac{D\gamma}{2} \exp[-\gamma\tau]. \tag{2.6}$$

Integrating over τ, we deduce

$$D_d = \int_0^\infty C_d(\tau)\,\mathrm{d}\tau = \frac{D}{2}. \tag{2.7}$$

Taking the difference of equation (2.4) minus equation (2.2), implementing the over-damped assumption (where γ is large) and combining the independent Weiner processes into a single Weiner processes $\mathbf{W}$ gives

$$\mathrm{d}\mathbf{R} = -\frac{2\Phi'(R)}{M\gamma}\frac{\mathbf{R}}{R}\,\mathrm{d}t + 2\sqrt{D}\,\mathrm{d}\mathbf{W}.$$

The stationary distribution corresponding to this process is proportional to $\exp[-\Phi(R)/(MD\gamma)]$. Normalising, we find the distribution of dimer lengths equal to

$$\varrho(R) = \frac{\exp\left[-\frac{\Phi(R)}{MD\gamma}\right]}{4\pi\int_0^\infty r^2 \exp\left[-\frac{\Phi(r)}{MD\gamma}\right]\,\mathrm{d}r}.$$

In the simulations that follow in this section, we shall be assuming the dimer potential acts like a linear spring with a rest length of ℓ_0 and a spring constant of k between the two monomers, as described in Equation (1.3). Each monomer within the dimer is representing a half of a molecule of interest and the value of the spring constant indicates the flexibility in which the molecule can change its shape. In this section, we consider the parameter regime where the spring constant k is sufficiently large so that the dimer has a well-defined structure. In the limit of large k, we have $\varepsilon = MD\gamma/(k\,\ell_0^2) \ll 1$. Then, L_d can be calculated by integrating over $\varrho(R)$ and taking a Taylor expansion

$$L_d \approx \ell_0\,(1+2\varepsilon) = \ell_0\left(1 + \frac{2MD\gamma}{k\,\ell_0^2}\right), \tag{2.8}$$

which is valid up to the first order in ε. In particular, the presence of heat baths extends the dimer from its rest length on average. In the following two sections, we study two theoretical MD models, where we use the analytic expressions in equations (2.6), (2.7) and (2.8) to compare the macroscopic theory with the results obtained statistically from our simulations data.

2.1.2 Short-range interaction heat bath

We describe the two monomers as balls with radius r_0 and mass M which interact with point solvent particles when they collide with them. In particular, this is a theoretical model of a (very) short-range interaction heat bath. Between collisions, monomers follow Newton's second law of motion in the form

$$M\frac{\mathrm{d}\mathbf{V}_1}{\mathrm{d}t} = \Phi'(R)\,\frac{\mathbf{R}}{R}, \tag{2.9}$$

$$M\frac{\mathrm{d}\mathbf{V}_2}{\mathrm{d}t} = -\Phi'(R)\,\frac{\mathbf{R}}{R}, \tag{2.10}$$

where, following our notation introduced in Section 2.1.1, positions and velocities of the monomers are denoted by $\mathbf{X}_i$ and $\mathbf{V}_i$, respectively, and $\mathbf{R} = \mathbf{X}_2 - \mathbf{X}_1$.

Our short-range interaction heat bath is described in terms of positions $\mathbf{x}_i^j$ and velocities $\mathbf{v}_i^j$, of heat bath particles, where $i = 1, 2$ is the monomer number and $j = 1, 2, 3, \ldots$, is the number of the heat bath particle. Notice that this formulation allows us to consider two important cases: (a) each monomer has its own heat bath; (b) a single heat bath is shared by both monomers. By comparing our simulation results in cases (a) and (b), we can explicitly investigate whether there are any significant hydrodynamic interactions between the monomers. In the case (b), we simplify our notation by describing particles of the single heat bath by

$$\mathbf{x}^j = \mathbf{x}_1^j = \mathbf{x}_2^j, \quad \text{and} \qquad \mathbf{v}^j = \mathbf{v}_1^j = \mathbf{v}_2^j, \tag{2.11}$$

meaning each heat bath particle can interact with both monomers. In both cases (a) and (b), we assume that all heat bath particles have the same mass, m, and define (dimensionless) parameter μ by

$$\mu = \frac{M}{m}.$$

We are interested in the parameter regime where $\mu \gg 1$. Our MD model is based on elastic collisions of heavy monomers (balls with mass M and radius r_0) with point heat bath particles with masses m. We assume that the collisions are without

friction, then conservation of momentum and energy yields the following formulae for post-collision velocities [125]

$$\widetilde{\mathbf{V}}_i = [\mathbf{V}_i]^{\parallel} + \frac{\mu-1}{\mu+1}\,[\mathbf{V}_i]^{\perp} + \frac{2}{\mu+1}\,\left[\mathbf{v}_i^j\right]^{\perp}, \tag{2.12}$$

$$\widetilde{\mathbf{v}}_i^j = \left[\mathbf{v}_i^j\right]^{\parallel} + \frac{1-\mu}{\mu+1}\,\left[\mathbf{v}_i^j\right]^{\perp} + \frac{2\mu}{\mu+1}\,[\mathbf{V}_i]^{\perp}, \tag{2.13}$$

where $\mathbf{v}_i^j$ is the velocity of the heat bath particle which collided with the i^{th} monomer, tildes denote post-collision velocities, superscripts $\perp$ denote projections of velocities on the line through the centre of the monomer and the collision point on its surface, and superscripts $\parallel$ denote tangential components.

Heat bath models based on elastic collisions (2.12)–(2.13) have been studied by a number of authors [124–127]. Consider a single monomer in infinite domain $\mathbb{R}^3$, and let the heat bath consist of an infinite number of particles with positions distributed according to the spatial Poisson process with density

$$\lambda_\mu = \frac{3}{8r_0^2}\sqrt{\frac{(\mu+1)\,\gamma}{2\pi D}}. \tag{2.14}$$

This means that the number of points in a subset Ω of $\mathbb{R}^3$ has its probability mass function given by the Poisson distribution with mean $\lambda_\mu|\Omega|$, where $|\Omega|$ is the volume of Ω. Let the velocities of the heat bath particles be distributed according to the Maxwell-Boltzmann distribution

$$f_\mu(\mathbf{v}) = \frac{1}{\sigma_\mu^3(2\pi)^{3/2}}\exp\left[-\frac{v_1^2+v_2^2+v_3^2}{2\sigma_\mu^2}\right], \tag{2.15}$$

where $\mathbf{v} = [v_1, v_2, v_3]$ and

$$\sigma_\mu = \sqrt{(\mu+1)\,D\,\gamma}. \tag{2.16}$$

Then the monomer's behaviour is known to converge to the Langevin description [125, 127]. In particular, if we consider that each monomer has its own heat bath, we can show that the position and velocity of the monomers, $\mathbf{X}_i$ and $\mathbf{V}_i$, converge (for sufficiently large numbers of heat bath particles) to the solution of (2.1)–(2.4) in the limit $\mu \to \infty$.

In reality all beads representing a macromolecule exist within a single heat bath. Thus, we ask whether the correlations introduced by a bath of solvent which interacts with both monomers has a non-negligible affect on the equilibrium statistics of the dimer. Introducing such coupled heat baths for both short-range (in this section) and long-range (in Section 2.1.3) interactions we study whether there is a significant difference between the simulated one-bath and two-bath models as we vary ℓ_0, the separation distance, introduced in equation (1.3). In order to study this problem, we make use of multi-resolution modelling.

Multi-resolution model using a co-moving frame

The solvent in the short-range heat bath interacts with the monomers of the dimer through direct contact. In order to simulate the model for long times, i.e. where the dimer has undergone a large excursion, the simulated domain must be vast as will be the number of solvent particles that must be modelled. We present a multi-resolution approach where we only model the solvent that is within the close vicinity of the dimer. We consider a co-moving cubic frame of length L that is centred at $\mathbf{X}_{\mathrm{f}}(t)$, which we here identify with the centre of mass of the dimer at time t, i.e.

$$\mathbf{X}_{\mathrm{f}}(t) = \overline{\mathbf{X}}(t) = \frac{\mathbf{X}_1(t) + \mathbf{X}_2(t)}{2}. \tag{2.17}$$

Within this frame we explicitly model the heat bath with solvent particles, i.e. they are simulated in the cubic box

$$\mathbf{X}_{\mathrm{f}}(t) + \left[-\frac{L}{2}, \frac{L}{2}\right] \times \left[-\frac{L}{2}, \frac{L}{2}\right] \times \left[-\frac{L}{2}, \frac{L}{2}\right]. \tag{2.18}$$

Externally we model the heat bath as a continuum, where the particles are distributed according to the spatial Poisson process with density λ_μ given in (2.14) and the velocities are distributed according to $f_\mu(\mathbf{v})$ given in (2.15), see Figure 2.1(a) for a diagrammatic representation of the multi-resolution framework (drawn for clarity in two spatial dimensions, while all our simulations are three-dimensional). As the dimer moves around in $\mathbb{R}^3$ the frame will move with it. In order for the multi-resolution model to capture the full model where solvent

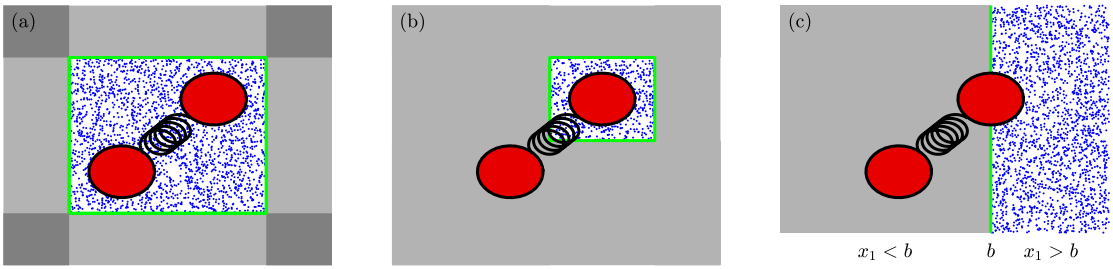


Figure 2.1: *A diagrammatic representation of multi-resolution approaches for a dimer in a heat bath with short-range interactions. While these are rendered in 2 dimensions for clarity, in practice simulations were carried out in 3 dimensions* (a) *Simulation of the whole dimer in a co-moving frame. The green box depicts the co-moving frame that is centred about the dimer. The blue dots correspond to solvent molecules that are explicitly modelled. Solvent molecules are not explicitly modelled in the external gray regions.* (b) *Simulation of one monomer in a co-moving frame.* (c) *Simulation with a fixed region of space where an MD model is explicitly used. A dimer molecule can move to the gray region where it is simulated using the Langevin description. Reproduced from our paper* [1] *with permission (Creative Commons attribution license).*

particles are distributed in the entire domain, $\mathbb{R}^3$, we need to introduce new solvent particles at the boundary of the frame.

Consider that time is discretised using small time step Δt, i.e. if the current time is t, we want to calculate the state of the system at time $t + \Delta t$. In our simulations of the multi-resolution model we need the probability of introducing a particle at a boundary of frame (2.18) in a timestep of length Δt and subsequently the distribution of the position $\mathbf{x}_{\text{new}}$ and velocity $\mathbf{v}_{\text{new}}$ of the new solvent particle. For simplicity we transform into the coordinate system of the co-moving frame which over an interval of length Δt has velocity

$$\mathbf{V}_{\text{f}} = \frac{\mathbf{X}_{\text{f}}(t + \Delta t) - \mathbf{X}_{\text{f}}(t)}{\Delta t}. \tag{2.19}$$

The frame is always translated to occupy the region $[0, L]^3$. Thus, the velocities for the solvent particles in the new reference frame are given by $\mathbf{w}^j = \mathbf{v}^j - \mathbf{V}_{\text{f}}$. We first calculate the density of particles that enter the frame via a particular boundary within a timestep of length Δt. Take, as an illustrative example, the boundary face corresponding to $\{x_1 = 0\}$. Consider particles which are in half-space $(-\infty, 0) \times \mathbb{R}^2 \in \mathbb{R}^3$ at time t. These particles have not yet been explicitly included in the simulation. Some of them will be in half-space $(0, \infty) \times \mathbb{R}^2 \in \mathbb{R}^3$ at time

$t + \Delta t$. Their density, $h(x_1)$, only depends on their first coordinate $x_1 \in (0, \infty)$. We can calculate $h(x_1)$ by integrating density (2.14)–(2.15) over solvent particles which are at $x_1' \in (-\infty, 0)$ at time t and have the appropriate velocity to reach $x_1 \in (0, \infty)$ at time $t + \Delta t$, namely as [127]

$$\begin{aligned} h(x_1) &= \int_{-\infty}^{0} \int_{\mathbb{R}^2} \lambda_\mu \, f_\mu \left(\frac{x_1 - x_1'}{\Delta t} + V_{\mathrm{f};1}, v_2, v_3 \right) \mathrm{d}v_2 \, \mathrm{d}v_3 \, \mathrm{d}x_1' \\ &= \frac{\lambda_\mu}{2} \operatorname{erfc}\left(\frac{x_1 + V_{\mathrm{f};1} \Delta t}{\sigma_\mu \Delta t \sqrt{2}} \right), \end{aligned} \tag{2.20}$$

where $V_{\mathrm{f};1}$ is the first component of the frame velocity and

$$\operatorname{erfc}(z) = 2/\sqrt{\pi} \int_z^\infty \exp\bigl(-s^2\bigr) \, \mathrm{d}s \tag{2.21}$$

is the complementary error function. Integrating (2.20) over the domain $(0, \infty) \times [0, L] \times [0, L] \in \mathbb{R}^3$ gives us the average number of particles that have entered the frame from the $\{x_1 = 0\}$ boundary in a time interval of length Δt as

$$\begin{aligned} p_{\mathrm{in}} &= \int_0^\infty \int_0^L \int_0^L h(x_1) \, \mathrm{d}x_3 \, \mathrm{d}x_2 \, \mathrm{d}x_1 \\ &= \lambda_\mu L^2 \Delta t \left(\frac{\sigma_\mu}{\sqrt{2\pi}} \exp\left[-\frac{V_{\mathrm{f};1}^2}{2\sigma_\mu^2} \right] - \frac{V_{\mathrm{f};1}}{2} \operatorname{erfc}\left[\frac{V_{\mathrm{f};1}}{\sigma_\mu \sqrt{2}} \right] \right). \end{aligned} \tag{2.22}$$

In our simulations we choose a timestep small enough that $p_{\mathrm{in}} \ll 1$, we can therefore use p_{in} as the probability of introducing a new solvent particle. Let $\mathbf{z} = [z_1; z_2; z_3]$ be the position of the new solvent particle in the coordinate system of the co-moving frame. Then coordinates z_2 and z_3 are uniformly distributed in $(0, L)$ and the first coordinate can be sampled from the error function distribution

$$C_1 \operatorname{erfc}\left[\frac{z_1 + V_{\mathrm{f};1} \Delta t}{\sigma_\mu \Delta t \sqrt{2}} \right], \qquad \text{for } \; z_1 \in (0, \infty), \tag{2.23}$$

where C_1 is a normalising constant. Then the position of the new solvent particle in the original coordinates is given by translating the sampled point relative to the centre of the frame, $\mathbf{x}_{\mathrm{new}} = \mathbf{z} + \mathbf{X}_{\mathrm{f}}(t + \Delta t) - [L/2, L/2, L/2]$. The velocity, $\mathbf{w}$, of the new particle in the co-moving frame must have a first coordinate exceeding $z_1/\Delta t$ in order to reach z_1 in a time interval of length Δt. Noting that $\mathbf{w} = \mathbf{v}_{\mathrm{new}} - \mathbf{V}_{\mathrm{f}}$

- Generate two random numbers η_1 and η_2 uniformly distributed in interval (0,1).
- Calculate $a_1(\beta)$ and $a_2(\beta)$ according to (2.27) and (2.28).
- Compute an exponentially distributed random number η_3 by $\eta_3 = -a_1(\beta)\,\log(\eta_1)$.
- If $\eta_1\,\eta_2 < a_2(\beta)\,\mathrm{erfc}(\eta_3 + \beta)$, then choose η_3 as a sample from the probability distribution (2.25). Otherwise, repeat the algorithm.

Table 2.1: *Acceptance-rejection algorithm for sampling random numbers according to the probability distribution $p(z;\beta)$ given by* (2.25).

can be derived from our expression for $\mathbf{x}_{\mathrm{new}}$, we write down the distribution of the velocity as the following truncated Gaussian distribution

$$C_2\,H(v_1\Delta t - (z_1 + V_{\mathrm{f};1}\Delta t))\,f_\mu(\mathbf{v}), \tag{2.24}$$

where C_2 is a normalising constant and $H(\cdot)$ is the Heaviside step function, satisfying $H(y) = 1$ for $y \in [0,\infty)$ and $H(y) = 0$ otherwise. The position and velocity of solvent particles introduced at the other five faces can be done by symmetric modifications of the above distributions.

Random numbers from distributions (2.23) and (2.24) can be efficiently sampled using acceptance-rejection algorithms. We use an acceptance-rejection method for the truncated normal distribution (2.24) presented in the literature [128], while we sample random numbers from the distribution (2.23) using the acceptance-rejection algorithm presented in Table 2.1. This is a generalisation of the acceptance-rejection algorithm for sampling random numbers according to the distribution $\sqrt{\pi}\,\mathrm{erfc}(z)$ previously used in simulations in the stationary frame [127]. In the case of the distribution (2.23), we need to sample random numbers according to the probability distribution

$$p(z;\beta) = C_3(\beta)\;\mathrm{erfc}(z+\beta), \tag{2.25}$$

where $\beta \in \mathbb{R}$ is a constant and $C_3(\beta)$ is the normalising constant given by

$$C_3(\beta) = \frac{\sqrt{\pi}}{\exp[-\beta^2] - \sqrt{\pi}\,\beta\,\mathrm{erfc}(\beta)}. \tag{2.26}$$

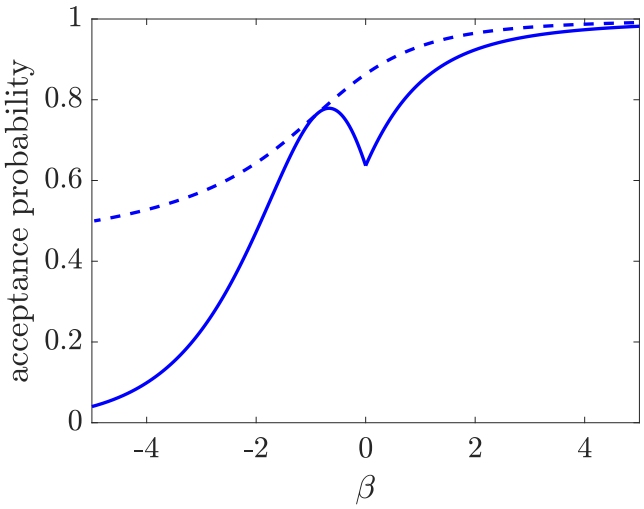


Figure 2.2: *Plot of the acceptance probability* (2.29) *of the algorithm presented in Table* 2.1 *for parameters given by* (2.27) *and* (2.28) *(solid line) compared with the acceptance probability* (2.29) *calculated for optimal choices of* $a_1(\beta)$ *and* $a_2(\beta)$ *for each parameter value* β *(dashed line). The kink in the acceptance probability is due to the discontinuity in* $a_2(\beta)$ *at* $\beta = 0$

The algorithm in Table 2.1 does this by generating an exponentially distributed random number η_3 with mean $a_1(\beta)$, where

$$a_1(\beta) = \frac{\sqrt{\pi}}{2} \times \begin{cases} \operatorname{erfc}(\beta)\exp(\beta^2), & \text{for } \beta \geq 0; \\ 1, & \text{for } \beta \leq 0. \end{cases} \tag{2.27}$$

To improve the acceptance probability (how often a sampled value is chosen to be used) of this algorithm, we choose its second parameter, $a_2(\beta)$, as

$$a_2(\beta) = \begin{cases} 1/\operatorname{erfc}(\beta), & \text{for } \beta \geq 0; \\ \exp\left(2\beta/\sqrt{\pi}\right), & \text{for } \beta \leq 0. \end{cases} \tag{2.28}$$

Then the acceptance probability, in terms of β, is defined as

$$\frac{a_2(\beta)}{a_1(\beta)\, C_3(\beta)}. \tag{2.29}$$

We plot the acceptance probability (2.29) in Figure 2.2 for our choices (2.27)-(2.28) of $a_1(\beta)$ and $a_2(\beta)$ as the solid line.

We observe that the acceptance probability (2.29) for $\beta = 0$ is equal to $2/\pi \approx 63.7\%$. This value can be improved [127] in the case of $\beta = 0$ to 86.3% provided that we choose $a_1 = 0.532$ and $a_2 = 0.814$. To obtain a similar improvement for all values of β, we could choose both $a_1(\beta)$ and $a_2(\beta)$ to maximise the acceptance probability

(2.29), rather than postulating that $a_1(\beta)$ is given by the piece-wise defined function (2.27) and optimising $a_2(\beta)$ only. The acceptance probability (2.29) of the resulting algorithm (which would have $a_1(\beta)$ and $a_2(\beta)$ given by a lookup table, rather than by using formulas (2.27)-(2.28)) is plotted in Figure 2.2 as the dashed line for comparison. However, in our illustrative simulations, we use the acceptance-rejection algorithm in Table 2.1 with the values of $a_1(\beta)$ and $a_2(\beta)$ given by (2.27)-(2.28).

Comparing equations (2.25) and (2.23), we observe that we can sample random numbers from the distribution (2.23) by sampling random numbers from the distribution $p(z; V_{\mathrm{f};1}\Delta t)$ (using the acceptance-rejection algorithm in Table 2.1 for $\beta = V_{\mathrm{f};1}\Delta t)$) and multiplying them by the factor $\sigma_\mu \Delta t\sqrt{2}$.

One iteration (i.e. an update of the state of the system from time t to time $t + \Delta t$) of the multi-resolution simulation algorithm in the co-moving frame is given as Algorithm [S1]–[S7] in Table 2.2. It evolves the positions and velocities of both monomers together with the positions and velocities of $N(t)$ solvent particles, where $N(t)$ does depend on time t. To formulate Algorithm [S1]–[S7], we assume that the timestep Δt is chosen small enough so that at most one collision happens per iteration.

We initialise the two monomers with a separation distance ℓ_0 and generate a Poisson number (with mean $\lambda_\mu\, L^3$) of solvent particles in our simulation domain, the cubic frame (2.18). The solvent particles are initially placed uniformly in the frame (2.18), where we remove particles overlapping with monomers (before we begin our simulation) to get the initial number, $N(0)$, of simulated solvent particles. Their initial velocities are drawn from the Maxwell-Boltzmann distribution (2.15).

In Step [S1], we update the system over the time interval $(t, t + \Delta t]$ using the "free-flight" positions for each monomer and solvent particle, namely we use

$$\widehat{\mathbf{X}}_i(t + \Delta t) = \mathbf{X}_i(t) + \mathbf{V}_i(t)\, \Delta t, \tag{2.30}$$

$$\widehat{\mathbf{x}}_i^j(t + \Delta t) = \mathbf{x}_i^j(t) + \mathbf{v}_i^j(t)\, \Delta t, \tag{2.31}$$

where $i = 1, 2$ is the monomer number and $j = 1, 2, \ldots, N(t)$, is the number of the heat bath particle and the velocities $\mathbf{V}_i$ and $\mathbf{v}_i^j$ are determined by previous particle

collisions, with $\mathbf{V}_i = 0$ if there are no collisions. Since Δt is chosen so small that only one collision happens during the time interval $[t, t+\Delta t)$, most of the "free-flight" positions of solvent particles are accepted in Step [S2] as their updated positions $\mathbf{x}_i^j(t+\Delta t)$ and only the solvent particle colliding with a monomer is further updated.

In Step [S3], we update the velocities of the monomers by solving (2.9)–(2.10) over one time step $[t, t+\Delta t]$. We discretise (2.9)–(2.10) using the forward Euler method as follows

$$\mathbf{V}_1(t+\Delta t) = \widetilde{\mathbf{V}}_1 + \frac{\Phi'(R)}{M}\frac{\mathbf{R}}{R}\,\Delta t, \tag{2.32}$$

$$\mathbf{V}_2(t+\Delta t) = \widetilde{\mathbf{V}}_2 - \frac{\Phi'(R)}{M}\frac{\mathbf{R}}{R}\,\Delta t, \tag{2.33}$$

where $\widetilde{\mathbf{V}}_i$, for $i = 1, 2$, is either the post collision velocity (if a collision happened in Step [S2]) or is equal to $\mathbf{V}_i(t)$. We choose this simple first order method as velocity changes only occur discretely on collision, making the model very stable. In Steps [S4]–[S5], we update the position and velocity of the frame. We remove solvent particles which leave our simulation domain in this step and update $N(t)$ accordingly, with particles only able to 're-enter' via the sampling in the next step.

In Step [S6], we use probability p_{in}, given by (2.22), to check whether any solvent particle entered the simulation domain during the time interval $(t, t+\Delta t]$. Since p_{in} is the probability of entering the domain through one of its six sides, we can, for time step Δt small enough that $6p_{\text{in}} \ll 1$, introduce at most one solvent particle through a randomly chosen side with probability $6p_{\text{in}}$. The initial position and velocity of the introduced solvent particle are sampled according to distributions (2.23) and (2.24) or their symmetric modifications, taking into account through which side of the cubic frame (2.18) the particle entered the frame.

There is one little caveat in our derivation of p_{in}. To derive equation ((2.22)) we integrated over the half-space $(-\infty, 0) \times \mathbb{R}^2$, meaning that once we consider all six faces of the cubic frame (2.18) we have over-counted twice at the edges and three times at the corners (as it is highlighted with darker grey shading in our illustrative diagram in Figure 2.1(a)). This will have negligible effect if we choose L sufficiently large. However, it can bias our simulation for values of L

comparable with the monomer size r_0 when Δt is not sufficiently small as boundary effects become more pronounced. To compensate for this effect, we consider the sampled position, $\mathbf{x}_{\text{new}}$ and velocity $\mathbf{v}_{\text{new}}$ of the new incoming particle at time $t+\Delta t$ and calculate its previous position at time t by

$$\mathbf{y} = \mathbf{x}_{\text{new}} - \mathbf{v}_{\text{new}}\,\Delta t.$$

If $\mathbf{y}$ is in the regions which were counted twice or three times in our derivation, we reject the proposed introduction of the new solvent particle with the corresponding probability. Namely, we use the acceptance probability in Step [S6] given by

$$h_{\text{acc}}(\mathbf{x}_{\text{new}}, \mathbf{v}_{\text{new}}) = \begin{cases} 1, & \text{for } \mathbf{y} - \mathbf{X}_{\text{f}}(t) \in \mathcal{Y}_1; \\ 1/2, & \text{for } \mathbf{y} - \mathbf{X}_{\text{f}}(t) \in \mathcal{Y}_2; \\ 1/3, & \text{for } \mathbf{y} - \mathbf{X}_{\text{f}}(t) \in \mathcal{Y}_3, \end{cases}$$

where $\mathcal{Y}_j \subset \mathbb{R}^3$ is the region of the space which consists of points which have exactly j of their coordinates outside of the interval $[-L/2, L/2]$. For example, in our two-dimensional diagrammatic representation in Figure 2.1(a), the lighter grey shading corresponds to region $\mathcal{Y}_1$ while the darker grey shading corresponds to region $\mathcal{Y}_2$.

In our illustrative simulations, we use algorithm [S1]–[S7] from Table 2.2 together with parameter values $r_0 = 0.08$, $\gamma = 10$, $D = 1$, $\mu = 10^3$, $k = 10^6$ and $L = 0.72$ for the one-bath case. These are chosen to ensure computational stability, along with convergence of simulation data to theoretically expected statistics. In Figure 2.2, we compare simulation results of the average length of the dimer at equilibrium, L_d, for the one-bath and two-bath models. Since the two-bath case uses uncoupled heat baths, we can further improve the efficiency of our algorithm by centering the co-moving frame corresponding to each heat bath on the corresponding monomer, i.e. we use $\mathbf{X}_{\text{f}}(t) = \mathbf{X}_i(t)$ for the heat bath corresponding to the i^{th} monomer in Step [S4] (instead of the centre of mass (2.17)) and choose smaller value of L in the two-bath case, namely $L = 0.32$. In both one-bath and two-bath models, the solvent particles are distributed according to the spatial Poisson process with density λ_μ given by (2.14). The velocities are distributed according to the Maxwell-Boltzmann distribution $f_\mu(\mathbf{v})$ given by (2.15). We note that in the two-bath case, our model converges to Langevin dynamics (2.1)–(2.4) as $\mu \to \infty$. This allows us to attribute

[S1] Update the positions of the solvent and the monomers by their "free-flight" positions (2.30)–(2.31).

[S2] If the "free-flight" position (2.31) of a solvent particle lies within the radius of either of the monomers, reverse the trajectories of the solvent and the monomer by time $\tau < \Delta t$ such that they are just touching. Calculate post-collision velocities by equations (2.12)–(2.13) and update their new positions by moving forward by time τ. Otherwise, each "free-flight" position is accepted as the particle's position at time $t + \Delta t$.

[S3] Update the velocities of the monomers by (2.32)–(2.33).

[S4] Calculate the new centre of the co-moving frame, $\mathbf{X}_{\mathrm{f}}(t + \Delta t)$, by (2.17). Update $N(t)$ by removing solvent particles which now lie outside of the frame (2.18) from the simulation.

[S5] Calculate the velocity of the frame, $\mathbf{V}_{\mathrm{f}}$, over the interval $[t, t + \Delta t]$ by equation (2.19).

[S6] Generate two random number r_1 and r_2 uniformly distributed in interval $(0, 1)$. If $r < 6p_{\mathrm{in}}$, then choose a side of the cube at random and generate proposed position $\mathbf{x}_{\mathrm{new}}$ and velocity $\mathbf{v}_{\mathrm{new}}$ of the new solvent particle according to distributions (2.23) and (2.24). If $r_2 < h_{\mathrm{acc}}(\mathbf{x}_{\mathrm{new}}, \mathbf{v}_{\mathrm{new}})$, then increase $N(t) = 1$ and initialise the new solvent particle at position $\mathbf{x}_{\mathrm{new}}$ with velocity $\mathbf{v}_{\mathrm{new}}$.

[S7] Continue with step [S1] using time $t = t + \Delta t$.

Table 2.2: *One iteration of the multi-resolution simulation algorithm of the dimer in a co-moving frame.*

any changes between the one-bath case and the Langevin model to the correlations induced by sharing a heat bath. The asymptotic analytic result obtained for the Langevin model, equation (2.8), is plotted as the black solid line for comparison.

In Figure 2.2, we set the separation distance to be $\ell_0 = \alpha\, r_0$ where $\alpha \geq 2$, to ensure that the monomers do not initially overlap. The plot shows the two-sided 99% confidence intervals for $(L_d - \ell_0)/\ell_0$ for $\alpha \in \{2.25, 2.5, 2.75, 3, 3.5, 4, 4.5, 5\}$. Firstly, we note that $L_d > \ell_0$ in each of the models as predicted in (2.8). There seems to be reasonable correspondence between the one- and two-bath models, with the confidence intervals overlapping. This suggests that the correlations we lose by approximating a larger co-moving frame around both monomers with two smaller dedicated frames around each monomer are negligible, allowing us to increase efficiency without biasing our overall results. In the next section, we build on this observation and present a multi-resolution framework which replaces

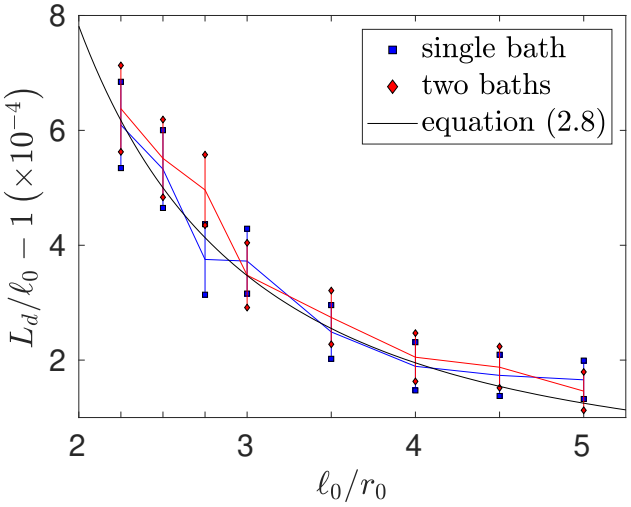


Figure 2.3: *The extension of the average length of a dimer from its separation distance* ℓ_0 *for short-range interaction heat bath models. The equilibrium data for each model was collected from a long-time simulation of length* 100 *dimensionless time units where* $\Delta t = 10^{-6}$ *and the monomers were initially placed with separation* ℓ_0*. The values of* $\alpha = \ell_0/r_0$ *presented are* $\{2.25, 2.5, 2.75, 3, 3.5, 4, 4.5, 5\}$*. The parameters used are* $r_0 = 0.08$*,* $\gamma = 10$*,* $D = 1$*,* $\mu = 10^3$*, and* $k = 10^6$*. Dimer initialised at origin at rest, with heat bath particle positions sampled uniformly according to equation 2.14 and velocities from distribution given by equation 2.15. In the one-bath case we use* $L = 0.72$ *for the frame* (2.18) *enclosing the whole dimer, and for the two-bath case, we use* $L = 0.32$ *for each monomer frame.*

one of the smaller dedicated frames by a coarser model of the heat bath, written in terms of Langevin dynamics.

Monomers with different resolution

As the length of a polymer (i.e. numbers of monomers) increases, a model incorporating solvent particles around each of the monomers becomes increasingly computationally expensive. However, a fully coarse-grained Langevin model of a polymer such as the Rouse model [95] can lack the required level of detail. Thus, some multi-resolution approaches for simulating macromolecules only model an important (small) part of a macromolecule using a detailed modelling approach [92–96]. In our case, we can mimic such methodologies by modelling the first monomer with explicit solvent with a heat bath of physical molecules, while the second monomer is modelled using the Langevin equations (2.3) and (2.4). Such a multi-resolution approach is schematically shown in Figure 2.1(b). To simulate this model we

use a co-moving frame, given by equation (2.18), which is centred around the first monomer, i.e. $\mathbf{X}_{\rm f}(t) = \mathbf{X}_1(t)$.

One iteration of the algorithm is presented as Algorithm [MR-S1]–[MR-S5] in Table 2.3. To begin, we initialise the particle positions and velocities in the similar way as in the case of Algorithm [S1]–[S7], with the only difference that the cubic frame (2.18) is now centred around the first monomer. Steps [MR-S1] and [MR-S2] are directly equivalent to steps [S1] and [S2]. In Step [MR-S3], we update the position and velocity of the second monomer by

$$\mathbf{V}_2(t+\Delta t) = \mathbf{V}_2(t) - \left(\frac{\Phi'(R)}{M} \frac{\mathbf{R}}{R} \ + \gamma \mathbf{V}_2(t) \right) \Delta t + \gamma \sqrt{2D\Delta t}\, \boldsymbol{\xi}, \tag{2.34}$$

where $\boldsymbol{\xi} \in \mathbb{R}^3$ is sampled from the normal distribution with zero mean and unit variance. That is, we have replaced the heat bath of the second monomer by solving the corresponding Langevin equation (2.1)–(2.4) using the standard Euler-Maruyama integrator. There have been other schemes developed in the literature for discretising the Langevin equation such as van Gunsteren and Berendsen [129] and the Langevin Impulse integrators, which capture the Langevin dynamics more accurately especially in the presence of forces, such as the spring force between the monomers [130]. Another option would be to consider the BBK integrator [131], which we use in Section 2.1.3, where we present a multi-resolution algorithm for the long-range interaction heat bath model and discretise the Langevin equation using a combination of the velocity Verlet and Euler-Maruyama (a first order stochastic scheme) integrators, see equations (2.47)-(2.51). An additional approach is the Verlet scheme [132] that approximates the velocity using a central difference discretisation rather than the forward difference approach used in the Euler-Maruyama method, or Runge-Kutta methods [133], which could further reduce the error of the multi-resolution simulations.

In order to compare simulations of the multi-resolution model with simulations of the Langevin model (2.1)–(2.4) we use the velocity autocorrelation function of the dimer, $C_d(\tau)$, given by equation (2.5). It has been analytically calculated for the Langevin description in equation (2.6). In Figure 2.3, we present numerical

[MR-S1]	Update the positions of the solvent and the monomers by their "free-flight" positions (2.30)–(2.31).
[MR-S2]	If the "free-flight" position (2.31) of a solvent particle lies within the radius of the first monomer, reverse the trajectories of the solvent and the monomer by time $\tau < \Delta t$ such that they are just touching. Calculate post-collision velocities by equations (2.12)–(2.13) for $i = 1$ and update their new positions by moving forward by time τ. Otherwise, each "free-flight" position is accepted as the particle's position at time $t + \Delta t$.
[MR-S3]	Update the velocity of the first monomer by (2.32) and the velocity of the second monomer by (2.34).
[MR-S4]	Calculate the new centre of the co-moving frame as $\mathbf{X}_{\mathrm{f}}(t+\Delta t) = \mathbf{X}_1(t)$. Update $N(t)$ by removing solvent particles which now lie outside of the frame (2.18) from the simulation. Use steps [S5]–[S6] from the algorithm in Table 2.2 to introduce new solvent particles into the co-moving frame (2.18).
[MR-S5]	Continue with step [MR-S1] using time $t = t + \Delta t$.

Table 2.3: *One iteration of the multi-resolution simulation algorithm of the dimer in the heat bath with short-range interactions, where the second monomer is simulated by the Langevin equation.*

estimates of the velocity autocorrelation function of the multi-resolution model from long time simulation data, using definition (2.5). To check simulations allowed for enough time for equilibrium to be each each, we calculated the evolution of the velocity autocorrelation and ensured it plateau around a constant value. In the long time simulation, we observed no noticeable growth in the errors over time, giving us confidence in the stability of our implementation.

Our results compare well with the theoretical result for the Langevin model, though it seems like there is a slightly raised value for $C_d(0)$. Using (2.7), we can estimate the diffusion constant of the dimer D_d by numerically integrating the velocity auto-correlation function in interval $[0, 1]$. We obtain $D_d \approx 0.529$, while its theoretical value for the dimer model is given in equation (2.7) as $D/2 = 0.5$. Another approach is to fit and exponential function, in the form seen in equation (2.6), to the computational result presented in Figure 2.3. In this way, the best fit values of both D and γ can be calculated simultaneously for our simulated data and compared to our input parameters. We found that $D \approx 1.0714$, which is

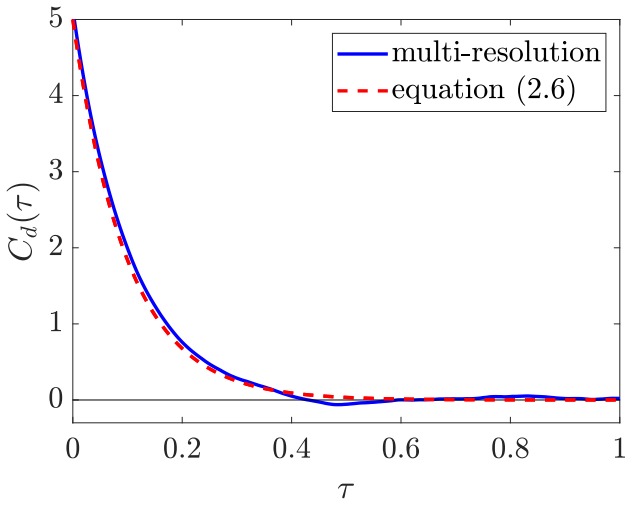


Figure 2.4: *The velocity autocorrelation function for the multi-resolution model (blue solid line) for short-range interactions. The function is estimated from long time simulation over dimensionless time of* 500 *time units. It is compared with the result for the Langevin description of the whole dimer, given by equation* (2.6) *(red dashed line). The parameters are* $r_0 = 0.08$, $\gamma = 10$, $D = 1$, $\mu = 10^3$, $k = 10^6$, $\ell_0 = 4r_0$, $\Delta t = 10^{-6}$ *and* $L = 0.32$. *Dimer initialised at origin at rest, with heat bath particle positions sampled uniformly according to equation 2.14 and velocities from distribution given by equation 2.15.*

higher than our parameter value $D = 1$, and $\gamma \approx 9.6064$, which is lower than $\gamma = 10$ used in our simulations. This could suggest that the value of λ_μ is too low or that of σ_μ is too high in our simulations. However, when these quantities are measured during the simulations we do not observe any deviation. This suggests that, rather than our sampling methods, there are small errors introduced by our implementation of the moving frame, or more profound boundary effects introduced by the small size of the frame. A potential problem in the implementation of the co-moving frame, is that solvent particles that leave the frame never return. For a stationary frame this is valid as the monomer cannot interact with a particle that leaves. However, for a co-moving small frame centred about the monomer, a solvent particle could leave the frame and return at a later time in the simulation. This is not taken into account in the presented algorithms, but could be addressed by introducing a buffer region around the co-moving frame. Particles could then be introduced in this volume rather than at the frame surface, and departing particles could return to the main frame so long as they stayed within the larger buffer boundaries.

2.1.3 Long-range interaction heat bath

Coarse-grained models of molecular systems can be written in terms of beads interacting through coarse-grained force fields. Each bead represents a collection of atoms and a coarse-grained potential energy can be constructed from detailed all-atom MD. Such an approach can usually provide a good description of equilibrium properties of molecular systems, but it does not necessarily lead to correct dynamics if the time evolution of the system is solely based on the Hamiltonian dynamics corresponding to the coarse-grained potential energy surface [134]. Dynamical behaviour can be corrected by introducing additional degrees for freedom (fictitious particles) interacting with each coarse-grained bead [134–136]. Fictitious particles can then be subject to suitable friction and noise terms to correct the dynamics.

Considering our dimer molecule model as an example of a coarse-grained molecule, written in terms of two coarse-grained beads (monomers) interacting through coarse-grained potential energy (1.3), then each monomer could be coupled with one or several fictitious particles interacting with the monomer through a suitable harmonic spring term [134, 135]. Our long-range interaction heat bath is based on this approach, by assuming that the i^{th} monomer, $i = 1, 2$, is coupled with N_i harmonic oscillators, in a manner similar to well known theoretical heat bath models [65, 67]. Then equations (2.9)–(2.10), expressing Newton's second law of motion, include additional terms as follows [65]

$$M\frac{\mathrm{d}\mathbf{V}_1}{\mathrm{d}t} = \Phi'(R)\,\frac{\mathbf{R}}{R} + \sum_{j=1}^{N_1} k_{1,j}\,\alpha_{1,j}\Big(\mathbf{x}_1^j - \alpha_{1,j}\mathbf{X}_1\Big), \tag{2.35}$$

$$M\frac{\mathrm{d}\mathbf{V}_2}{\mathrm{d}t} = -\Phi'(R)\,\frac{\mathbf{R}}{R} + \sum_{j=1}^{N_2} k_{2,j}\,\alpha_{2,j}\Big(\mathbf{x}_2^j - \alpha_{2,j}\mathbf{X}_2\Big), \tag{2.36}$$

where $\mathbf{x}_i^j$ is the position of the j^{th} solvent particle which interacts with the i^{th} monomer through a harmonic spring with spring constant $k_{i,j}$ and interaction constants $\alpha_{i,j}$, $j = 1, 2, \ldots, N_i$, $i = 1, 2$. Equations (2.35)–(2.36) are coupled with the evolution equations for solvent particles via these springs. We assume that $\mathbf{v}_i^j$ is the velocity of the j^{th} solvent particle interacting with the i^{th} monomer. Moreover,

we assume that all oscillators have the same mass, m. Using Newton's second law of motion, we get the following evolution equations for the heat bath oscillators

$$\frac{\mathrm{d}\mathbf{x}_i^j}{\mathrm{d}t} = \mathbf{v}_i^j, \tag{2.37}$$

$$m\frac{\mathrm{d}\mathbf{v}_i^j}{\mathrm{d}t} = -k_{i,j}\left(\mathbf{x}_i^j - \alpha_{i,j}\mathbf{X}_i\right), \tag{2.38}$$

for $j = 1, 2, \ldots, N_i$ and $i = 1, 2$. Unlike in some fictitious particle models [134–136], we do not include friction and random forces into equation (2.38) for solvent, because we assume that we explicitly model all solvent particles, i.e. N_1 and N_2 are considered to satisfy $N_1 \gg 1$ and $N_2 \gg 1$. We are therefore working 'close' to the limit $N_1 \to \infty$ and $N_2 \to \infty$, in which we can get the convergence of our long-range interaction heat bath to Langevin dynamics as discussed below. In practice, it is impossible to include all solvent molecules in simulations and friction and noise terms are still included to control temperature of the simulated system [33, 137]. We can solve the solvent equations of motion (2.37)–(2.38) to give [33, 64]

$$\mathbf{x}_i^j = \mathbf{x}_i^j(0)\cos\left(\omega_{i,j}t\right) + \frac{\mathbf{v}_i^j(0)}{\omega_{i,j}}\sin\left(\omega_{i,j}t\right) \; + \; \alpha_{i,j}\,\omega_{i,j}\int_0^t \sin\left(\omega_{i,j}(t-\tau)\right)\mathbf{X}_i(\tau)\,\mathrm{d}\tau$$

where $\mathbf{x}_i^j(0)$ is the initial position of the j^{th} heat bath particle corresponding to the i^{th} monomer, $\mathbf{v}_i^j(0)$ is its initial velocity and $\omega_{i,j} = (k_{i,j}/m)^{1/2}$ is its frequency. Substituting for $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$ in dimer's equations of motion (2.35)–(2.36), we obtain the following coupled system of generalised Langevin equations

$$M\frac{\mathrm{d}\mathbf{V}_1}{\mathrm{d}t} = \Phi'(R)\,\frac{\mathbf{R}}{R} - \int_0^t \kappa_1(\tau)\,\mathbf{V}_1(t-\tau)\,\mathrm{d}\tau + \boldsymbol{\xi}_1, \tag{2.39}$$

$$M\frac{\mathrm{d}\mathbf{V}_2}{\mathrm{d}t} = -\Phi'(R)\,\frac{\mathbf{R}}{R} - \int_0^t \kappa_2(\tau)\,\mathbf{V}_2(t-\tau)\,\mathrm{d}\tau + \boldsymbol{\xi}_2, \tag{2.40}$$

where the friction kernel $\kappa_i(\tau)$ and noise term $\boldsymbol{\xi}_i \equiv \boldsymbol{\xi}_i(t) = [\xi_{i;1}, \xi_{i;2}, \xi_{i;3}]$ are given by

$$\kappa_i(\tau) = m\sum_{j=1}^{N_i}\alpha_{i,j}^2\,\omega_{i,j}^2\cos\left(\omega_{i,j}\tau\right), \tag{2.41}$$

$$\boldsymbol{\xi}_i(t) = m\sum_{j=1}^{N_i}\mathbf{x}_i^j(0)\,\alpha_{i,j}\,\omega_{i,j}^2\cos\left(\omega_{i,j}t\right) + \mathbf{v}_i^j(0)\,\alpha_{i,j}\,\omega_{i,j}\sin\left(\omega_{i,j}t\right), \tag{2.42}$$

for $i = 1, 2$. We assume that initial positions and velocities of solvent oscillators, $\mathbf{x}_i^j(0)$ and $\mathbf{v}_i^j(0)$, are both independently sampled according to their equilibrium

distributions. Then noise autocorrelation function is given by the generalised fluctuation-dissipation theorem

$$\lim_{t\to\infty} \langle \xi_{i;j}(t)\, \xi_{i;n}(t-\tau)\rangle = D\gamma\, \delta_{j,n}\, \kappa_i(\tau),$$

where k_B is the Boltzmann constant and T is the absolute temperature. Next, we assume that the frequencies $\omega_{i,j}$ are sampled from a (continuous) exponential distribution with mean $\overline{\omega}$ and we set our interaction constants equal to

$$\alpha_{i,j} = \frac{1}{\omega_{i,j}} \sqrt{\frac{2\,\gamma\,\overline{\omega}}{N_i\, m\, \pi}}, \tag{2.43}$$

where $\gamma > 0$ is the friction constant used in equations (2.2) and (2.4). Then friction kernel (2.41) becomes

$$\kappa_i(\tau) = \frac{2\gamma\,\overline{\omega}}{\pi} \frac{1}{N_i} \sum_{j=1}^{N_i} \cos\left(\omega_{i,j}\tau\right).$$

Passing to the limit $N_i \to \infty$ allows us to consider the above summation as a continuous integral over the distribution of oscillator frequencies, with both friction kernels $\kappa_1(\tau)$ and $\kappa_2(\tau)$ converging to the same friction kernel [64]

$$\begin{aligned} \kappa(\tau) &= \frac{2\gamma}{\pi} \int_0^\infty \cos\left(\omega\tau\right) \exp\left(-\frac{\omega}{\overline{\omega}}\right) \mathrm{d}\omega \\ &= \frac{2\gamma}{\pi} \frac{\overline{\omega}}{\overline{\omega}^2\tau^2 + 1}. \end{aligned} \tag{2.44}$$

Then $\int_0^\infty \kappa(\tau)\,\mathrm{d}\tau = \gamma$. Moreover, we can define the limiting friction kernel by

$$\kappa_\infty(\tau) = \lim_{\overline{\omega}\to\infty} \kappa(\tau),$$

which, for our choice of oscillators' frequencies and interaction terms (2.43), satisfies $\kappa_\infty(\tau) = 0$ for $\tau > 0$ and $\kappa_\infty(0) = \infty$. Thus the limiting kernel is a multiple of the Dirac delta function. Therefore the position and velocity of the monomers, $\mathbf{X}_i$ and $\mathbf{V}_i$, converge to the solution of (2.1)–(2.4) in the limit $\overline{\omega} \to \infty$, provided that each monomer has its own separate heat bath. Moreover, we obtain the Einstein-Smoluchowski relation for the diffusion constant of the monomer as $D = k_B T/(\gamma\, M)$.

As in Section 2.1.2, we have explained our MD model of the dimer using the case where each monomer has its own heat bath. We now turn our attention to the

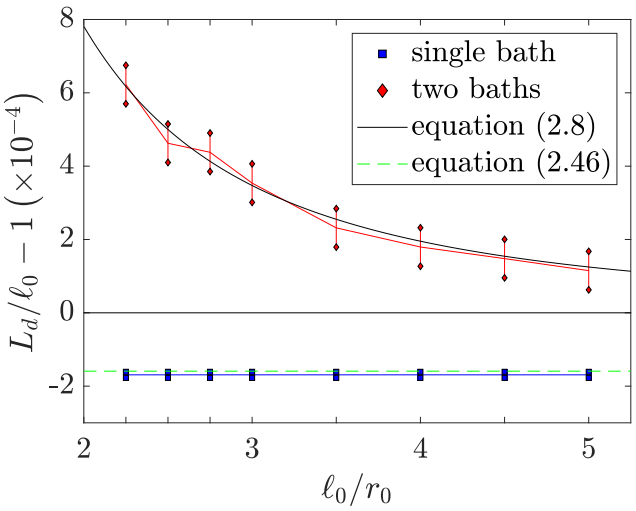


Figure 2.5: *The extension of the average length of a dimer from its separation distance ℓ_0 for long-range interaction heat bath models. The values of parameters are the same as in Figure* 2.2*, together with $\overline{\omega} = 100$, $N_1 = N_2 = N = 10^4$, $M = 1$ and $m = 10^{-3}$, which give the same value of $\mu = M/m$ as used in Figure* 2.2*. Dimer initialised at origin at rest, with initial heat bath particle positions and velocities sampled from equilibrium distributions. The simulations for the single heat bath case use parameter choice* (2.45) *with $\alpha_j^2 = \gamma\,\overline{\omega}/(N\,\pi\,k_j)$, $k_j = m\,\omega_j^2/2$, and ω_j sampled according to the exponential distribution with mean $\overline{\omega}$, confirming result given in equation* (2.46) *(green dashed line). The results for the two heat bath case are compared with the result obtained for the Langevin model in equation* (2.8) *(black solid line).*

case when monomers share their heat bath. This has been studied in the case of the short-range interaction MD model with the help of multi-resolution modelling in a co-moving frame, as schematically shown in Figure 2.1(a). In the case of long-range interactions, a co-moving frame is less straightforward to implement because we need to take into account that particles outside of the simulated box do exert (long-range) forces on particles in our simulation domain. Some multi-resolution techniques in the literature solve this problem by introducing suitable overlap (bridging, blending) regions [135, 138–140], where molecules which are near the simulation domain exert some partial forces on the simulated molecules.

In what follows, we do not truncate the simulated domain, but we consider a different multi-resolution approach. Before then we discuss results comparable to Figure 2.2, i.e. we compare simulations with a single heat bath and two heat baths for the case of our long-range interaction MD model. The results are presented

in Figure 2.5, where we use the same values of ℓ_0 as in Figure 2.2, expressed as α-multiples of r_0, although our long-range interaction model does not make use of parameter r_0. The value of L_d is for each value of ℓ_0 calculated from a long simulation over 200 dimensionless time units, where the first 100 time units are used to equilibrate the system, while the second half of each simulation is used to compute L_d. To initialise this model we start with monomers separated by the rest length ℓ_0 and sample oscillators' frequencies, $\omega_{i,j}$, according to the exponential distribution with mean $\overline{\omega}$. Their positions and velocities are sampled from the Maxwell-Boltzmann distribution. For the two-bath model, each dimer particle is separately initialised with its own set of oscillators around their respective positions in space.

In Figure 2.5, we observe that in the case of the two-bath model we obtain results which match well qualitatively with equation (2.8) for our parameter values. These results are also directly comparable with the results obtained for the two-bath case in Figure 2.2. The situation is more complicated in the case of simulations with a single heat bath with N oscillators. Then, using notation (2.11), we can rewrite (2.35)–(2.36) as

$$M\frac{\mathrm{d}\mathbf{V}_i}{\mathrm{d}t} = (-1)^{i+1}\Phi'(R)\,\frac{\mathbf{R}}{R} + \sum_{j=1}^{N} k_{i,j}\,\alpha_{i,j}\Big(\mathbf{x}^j - \alpha_{i,j}\mathbf{X}_i\Big),$$

for $i = 1, 2$, where the heat bath evolution equation (2.38) includes terms corresponding to both monomers

$$m\frac{\mathrm{d}\mathbf{v}^j}{\mathrm{d}t} = -k_{1,j}\left(\mathbf{x}^j - \alpha_{1,j}\mathbf{X}_1\right) - k_{2,j}\left(\mathbf{x}^j - \alpha_{2,j}\mathbf{X}_2\right),$$

for $j = 1, 2, \ldots, N$. Our results will then depend how we choose parameters $k_{i,j}$ and $\alpha_{i,j}$. For example, if we choose $k_{i,j}$ and $\alpha_{i,j}$ to be the same for both monomers, i.e.

$$k_{1,j} = k_{2,j} = k_j, \quad \text{and} \quad \alpha_{1,j} = \alpha_{2,j} = \alpha_j, \tag{2.45}$$

for $j = 1, 2, \ldots, N$, then the oscillating frequency of the j^{th} heat bath oscillator is $\omega_j = \sqrt{2k_j/m}$ and we can subtract the evolution equations for monomers to obtain

$$M\frac{\mathrm{d}^2\mathbf{R}}{\mathrm{d}t^2} = -2\Phi'(R)\,\frac{\mathbf{R}}{R} - \sum_{j=1}^{N} k_j\,\alpha_j^2\,\mathbf{R}.$$

This equation does not contain any heat bath variables. Using (2.43) to select α_j, i.e. using $\alpha_j^2 = 2\,\gamma\,\overline{\omega}/(N\,m\,\pi\,\omega_j^2) = \gamma\,\overline{\omega}/(N\,\pi\,k_j)$, we get

$$M\frac{\mathrm{d}^2\mathbf{R}}{\mathrm{d}t^2} = -2\Phi'(R)\,\frac{\mathbf{R}}{R} + \frac{\gamma\,\overline{\omega}}{\pi}\,\mathbf{R}.$$

Using potential (1.3), we conclude that we effectively obtain a shorter rest length of the spring which gives the following approximation

$$L_d \approx \frac{2\,k\,\pi\,\ell_0}{2\,k\,\pi + \gamma\,\overline{\omega}}. \tag{2.46}$$

This result is plotted in Figure 2.5 together with results obtained by illustrative simulations. We use a long-time simulation of length 200 dimensionless time units, with monomers initially placed at separation ℓ_0 and averaging over the second half of the simulation (of length 100 dimensionless time units) to obtain the presented values of dimer's expected length L_d.

In Figure 2.5, we observe that the average dimer length, L_d, during our single heat bath simulations is smaller than the natural length of the spring, ℓ_0. However, this conclusion is only a consequence of our choice of parameters (2.45). An opposite phenomenon can be observed in simulations for other parameter regimes. For example, if we divide our oscillators into two groups consisting of N_1 and N_2 oscillators, i.e. $N = N_1 + N_2$, and choose our parameters $k_{i,j}$ and $\alpha_{i,j}$ such that

$$\begin{aligned} k_{2,j} &= 0, \qquad && \text{for } j = 1, 2, \ldots, N_1, \\ k_{1,j} &= 0, \qquad && \text{for } j = N_1 + 1, N_1 + 2, \ldots, N, \end{aligned}$$

then our "one-bath" case is effectively equal to the two-bath case for which we have the result given in equation (2.8) presented in Figure 2.5. In particular, depending on our choices of $k_{i,j}$ and $\alpha_{i,j}$, the single heat bath case can both increase or decrease the average length of the dimer.

Multi-resolution modelling of dimer

In Figure 2.1(b), we use our dimer example to illustrate a multi-resolution approach which models a part of a molecule using a detailed MD approach, while using a

[MR-L1] Update velocities of the dimer and solvent particles for a half time step using (2.47).
[MR-L2] Update positions of the dimer and solvent particles using (2.48).
[MR-L3] Recalculate accelerations of each monomer and solvent oscillators by (2.50), (2.51) and (2.52).
[MR-L4] Update velocities of the dimer and solvent particles for a half time step using (2.49).
[MR-L5] Continue with step [MR-L1] using time $t = t + \Delta t$.

Table 2.4: *One iteration of the multi-resolution simulation algorithm of the dimer in the heat bath with long-range interactions, where the second monomer is simulated by the Langevin equation.*

coarser description of the rest of the molecule. Here, in the same manner as carried out for our short-range model, we illustrate such a multi-resolution approach using our long-range interaction MD model. We use the Langevin model (2.1)–(2.4) to coarse-grain one of the monomers, while the other monomer is modelled in detail using the MD model with its heat bath described by harmonic oscillators (2.37)–(2.38). As in Figure 2.3, we again calculate numerical estimates for the velocity autocorrelation function, $C_d(\tau)$ in equation (2.5), from long time simulations of the dimer after equilibrium has been reached.

The pseudo-code of one iteration our multi-resolution algorithm is presented as Algorithm [MR-L1]–[MR-L5] in Table 2.4. Algorithm [MR-L1]–[MR-L5] is based on the velocity Verlet integrator, where both monomers are updated by

$$\mathbf{V}_i\left(t + \tfrac{1}{2}\Delta t\right) = \mathbf{V}_i(t) + \frac{1}{2}\,\mathbf{A}_i(t)\,\Delta t, \tag{2.47}$$

$$\mathbf{X}_i(t + \Delta t) = \mathbf{X}_i(t) + \mathbf{V}_i(t + \tfrac{1}{2}\Delta t)\,\Delta t, \tag{2.48}$$

$$\mathbf{V}_i(t + \Delta t) = \mathbf{V}_i\left(t + \tfrac{1}{2}\Delta t\right) + \frac{1}{2}\,\mathbf{A}_i(t + \Delta t)\,\Delta t, \tag{2.49}$$

where $\mathbf{A}_i$, for $i = 1, 2$, is the acceleration of the corresponding monomer. For the first monomer, its acceleration $\mathbf{A}_1$ is defined as the right hand side of equation (2.35) divided by M, i.e.

$$\mathbf{A}_1 = \frac{\Phi'(R)}{M}\,\frac{\mathbf{R}}{R} + \frac{1}{M}\sum_{j=1}^{N_1} k_{1,j}\,\alpha_{1,j}\Big(\mathbf{x}_1^j - \alpha_{1,j}\mathbf{X}_1\Big). \tag{2.50}$$

For the second monomer, we use the BBK integrator [131], i.e. we define its acceleration as

$$\mathbf{A}_2 = -\frac{\Phi'(R)}{M}\,\frac{\mathbf{R}}{R} - \gamma \mathbf{V}_2 + \gamma\sqrt{\frac{2D}{\Delta t}}\,\boldsymbol{\xi}, \tag{2.51}$$

where each element of $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]$ is sampled from the normal distribution with zero mean and unit variance. The corresponding solvent oscillator integrator is identical to the scheme (2.47)–(2.49), with $\mathbf{X}_1$, $\mathbf{V}_1$ and $\mathbf{A}_1$ replaced by $\mathbf{x}^j$, $\mathbf{v}^j$ and $\mathbf{a}^j$, respectively, where acceleration $\mathbf{a}^j$ is defined as the right hand side of equation (2.38) divided by m, i.e.

$$\mathbf{a}^j = -\frac{k_{i,j}}{m}\left(\mathbf{x}_i^j - \alpha_{i,j}\mathbf{X}_i\right). \tag{2.52}$$

The results obtained by Algorithm [MR-L1]–[MR-L5] are compared with analytic results given by equation (2.6) for the Langevin model in Figure 2.6. We see that there is a good correspondence between these, suggesting that the value $\bar{\omega} = 100$ is large enough to create an accurate Dirac delta approximation from the kernel function (2.44), along with having a large enough number of oscillators, $N_1 = 10^5$, in our heat bath for our other approximations to hold. If these conditions did not hold, we would see that our kernel function has a different form (for example, decaying at a slower rate), and in this case we would have to use a generalised Langevin model as our coarse-graining approach in order to capture the dynamics of the dimer with sufficient accuracy.

The diffusion constant of the dimer, D_d, can again be estimated by numerically integrating the velocity auto-correlation function. Integrating our results from Figure 2.6 over interval $[0, 1]$, we obtain $D_d \approx 0.510$, which compares well with the theoretical value, $D/2 = 0.5$, given by equation (2.7).

2.2 Extension to generalised Langevin model

In the previous section. the long-range interaction model leads to the system of generalised Langevin equations, given by equations (2.39)–(2.40). Although we have worked in the parameter regime where the generalised Langevin equations

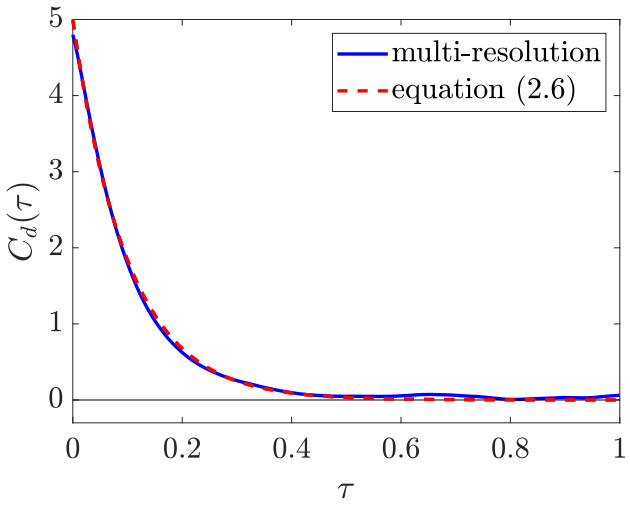


Figure 2.6: *The velocity autocorrelation function for the multi-resolution model (blue solid line) for long-range interactions, estimated from long time simulation over dimensionless time of* 10^3 *dimensionless time units. It is compared with the result for the Langevin description of the whole dimer, given by equation* (2.6) *(red dashed line). The parameters are the same as in Figure* 2.3*, namely* $\gamma = 10$*,* $D = 1$*,* $k = 10^6$*,* $M = 1$*,* $m = 10^{-3}$*,* $\ell_0 = 0.32$*, together with* $\varpi = 100$ *and* $N_1 = 10^5$*. Dimer initialised at origin at rest, with initial heat bath particle positions and velocities sampled from equilibrium distributions*

can be well approximated by the system of Langevin equations given by (2.1)–(2.4), this will not be the case in other parameter regimes and for more realistic solvent descriptions, especially when the friction kernel is estimated from MD simulations [70, 71], which we will now explore further.

The section is organised as follows. In Section 2.2.1 we examine the friction kernel properties for various choices of oscillator frequencies, with attention paid to case where choices of parameters weakens the Dirac delta approximation used to derive the parameter coupling for the Langevin model used in Section 2.1.3. With this in mind, Section 2.2.2 looks to solving the generalised Langevin equation, with a slow decaying friction kernel and more pronounced memory effects i.e. dynamics of the simulation not simply depending on the previous timestep, but the values of the noise and velocity at past times. Finally, in Section 2.2.3 we see how it is possible to directly estimate the form of a friction kernel from Hamiltonian MD simulation data. Correlation functions are used to numerically construct the kernel, resulting in a generalised Langevin model with matched dynamics to the

Hamiltonian model, allowing for a more sophisticated multi-resolution dimer to be formulated, generalising the work of Section 2.1.3.

2.2.1 Alternative oscillator frequencies and interaction constant parametrisations

In Section 2.1 we assumed that the oscillator frequencies ω_i were sampled from a continuous exponential distribution with mean $\overline{\omega}$, and interaction constants given by equation (2.43). In the limit $N \to \infty$ it was possible to find a closed analytic form for the friction kernel seen in equation (2.44). In this Section, we will restrict ourselves to a single particle case, rather than the dimer considered in the previous section, with almost identical equations of motion for the particle and oscillators

$$\begin{aligned} M\frac{\mathrm{d}\mathbf{V}}{\mathrm{d}t} &= \sum_{j=1}^{N} k_j\,\alpha_j\Big(\mathbf{x}^j - \alpha_j\mathbf{X}\Big), \\ \frac{\mathrm{d}\mathbf{x}_i^j}{\mathrm{d}t} &= \mathbf{v}^j, \\ m\frac{\mathrm{d}\mathbf{v}^j}{\mathrm{d}t} &= -k_j\left(\mathbf{x}^j - \alpha_j\mathbf{X}\right), \end{aligned} \tag{2.53}$$

and corresponding generalised Langevin equation form

$$M\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = -\int_0^t \kappa(\tau)\boldsymbol{V}(t-\tau)\,\mathrm{d}\tau + \boldsymbol{\xi}. \tag{2.54}$$

It is possible to go further and find an analytic solution to the fully equations of motion for the Hamiltonian model in the single parameter case, for which we give full details of in Appendix A. Taking the limit $\overline{\omega} \to \infty$ then enabled us to use the approximation $\kappa(\tau) \approx \gamma\delta(\tau)$ to derive the Langevin model

$$M\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = -\gamma\boldsymbol{V} + \boldsymbol{\xi}, \tag{2.55}$$

with solvent behaviour matched to the full Hamiltonian description. While this resulted in the good theoretical correspondence seen in Figure 2.6 between the respective models used for the multi-resolution dimer, the requirement to have a large value $\overline{\omega}$ is restrictive. In Figure 2.7 we see this demonstrated, with slower decay rate seen when $\overline{\omega}$ is high. We compared the time-averaged velocity fluctuations $\langle \boldsymbol{V}^2 \rangle$ for simulation trajectories and the analytic solutions the single particle Hamiltonian

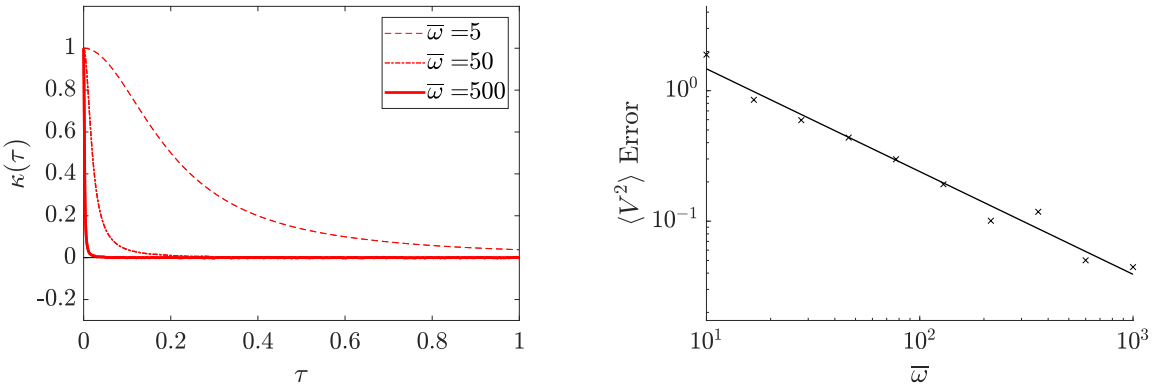


Figure 2.7: *Accuracy of Dirac delta function approximation of the friction kernel when sampling frequencies from exponential distribution. In the first panel we plot* $\kappa(\tau)$ *for 3 different values of* $\overline{\omega}$ *and normalise such that* $\kappa(0) = 1$*. In second panel, we plot error in* $\langle \boldsymbol{V}^2 \rangle$ *between analytic form in equation* (A.4) *and fully simulated Hamiltonian model in equation* (2.53) *over* 10 *dimensionless time units. Non-varied parameters used were* $\gamma = 10$*,* $D = 1$*,* $M = 1$*,* $m = 10^{-3}$*, and* $N = 10^5$*.*

system (see Appendix A) and found that accuracy improved as the value of $\overline{\omega}$ was increased as expected. Each cross in the second panel in Figure 2.6 represents a single set of sampled ω_i for a given ω_d and their corresponding α_i, with the error defined as the percentage difference in $\langle \boldsymbol{V}^2 \rangle$ found when calculating it from simulated trajectories using equation (2.53) as opposed to from equation (A.4).

This choice of oscillator frequencies is not unique, and indeed for different situations it maybe be necessary to sample from different distributions to obtain meaningful results. An alternative discussed in the literature [67] is the Debye distribution, which can also be used to reduce the friction kernel to the Dirac delta function, and is defined as follows:

$$f(\omega) = \frac{3\omega^2}{\omega_d^3} \ , \ 0 \leq \omega < \omega_d.$$

This time, a slightly different choice of interaction coefficient is made to account for the change in frequency distribution

$$\alpha_i = \frac{1}{\omega_i^2} \sqrt{\frac{2\,\gamma\,\omega_d^3}{3\,N\,m\,\pi}},$$

Again assuming that N is large enough to treat the summation as a continuous integral, we derive a new friction kernel

$$\kappa(\tau) = \frac{2\gamma}{\pi} \int_0^{\omega_d} \cos\left(\omega\tau\right) \, \mathrm{d}\omega = \frac{2\gamma}{\pi} \frac{\sin(\omega_d \tau)}{\tau}, \tag{2.56}$$

$$\int_0^\infty \kappa(\tau)\mathrm{d}\tau = \gamma, \quad \kappa(0) = \frac{2\gamma\omega_d}{\pi}.$$

As we did in Section 2.1.3, by considering the limiting behaviour of the friction kernel for large ω_d

$$\kappa_\infty(\tau) = \lim_{\omega_d \to \infty} \kappa(\tau),$$

for this choice of oscillators' frequencies and interaction terms (2.2.1), we see that it satisfies $\kappa_\infty(\tau) = 0$ for $\tau > 0$ and $\kappa_\infty(0) = \infty$. Thus the limiting kernel is a multiple of the Dirac delta function, $\kappa(\tau) \approx \gamma\delta(\tau)$. We can also generalise this to other distributions with a polynomial as a probability density function

$$f(\omega) = \frac{(n+1)\omega^n}{\omega_d^{n+1}} \ , \ 0 \leq \omega < \omega_d,$$

$$\alpha_i = \frac{1}{\omega_i^{(n/2+1)}} \sqrt{\frac{2\,\gamma\,\omega_d^{n+1}}{(n+1)\,N\,m\,\pi}}.$$

This will again lead to a form of the kernel seen in equation (2.56). Setting $n = 0$ leaves us with a simple uniform distribution for oscillator frequencies between 0 and ω_d, with the interaction term in this case the similar to equation (2.43) used for the exponential frequency case, with $\overline{\omega}$ changed to ω_d. In Figure 2.8 we see how $\kappa(\tau)$ behaves for this choice of interaction and oscillator frequencies, with a slower decay rate seen when the cutoff frequency ω_d is high, however and increased frequency of oscillation is also seen, which is reminiscent of the Gibbs phenomenon seen when trying to approximate a function that is not smooth with a Fourier series. If a similar principle holds here, we would expect the overshoot to be roughly

$$\left(\int_0^\pi \frac{\sin(\tau)}{\tau}\mathrm{d}\tau - \frac{\pi}{2}\right) / \left(\frac{\pi}{2}\right) \approx 17.898\%,$$

which is similar to what we observe. We also compared the error in $\langle \boldsymbol{V}^2 \rangle$ between the trajectories over a simulation to the analytic solutions for the single particle Hamiltonian system (see Appendix A) as we did for the exponential case and found that accuracy improved as the value of ω_d was increased as expected.

Another possible option would be to approximate the Dirac delta function with a convergent sum. While this is harder to justify in a physical setting, it provides

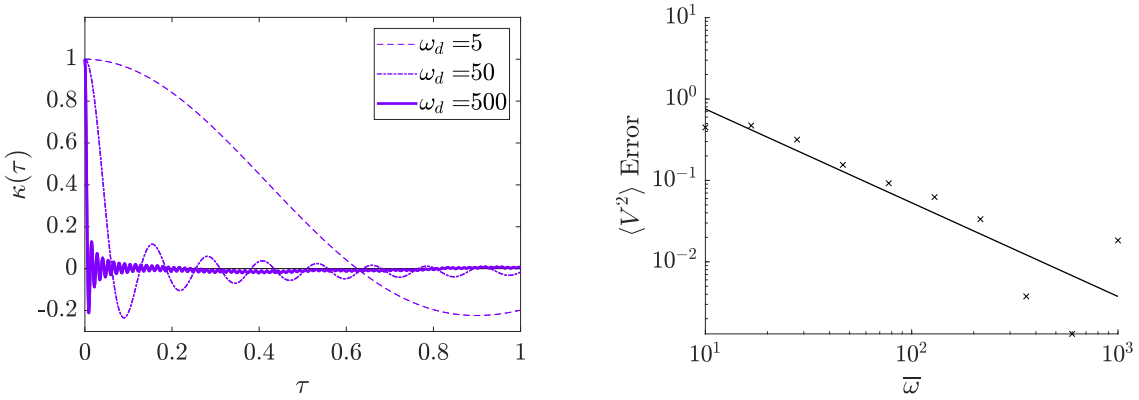


Figure 2.8: *Accuracy of Dirac delta function approximation of the friction kernel when sampling frequencies from uniform distribution. In the first panel we plot $\kappa(\tau)$ for 3 different values of ω_d and normalise such that $\kappa(0) = 1$. In second panel, we plot error in $\langle \boldsymbol{V}^2 \rangle$ between analytic form in equation (A.4) and fully simulated Hamiltonian model in equation (2.53) over 10 dimensionless time units. Non-varied parameters used were $\gamma = 10$, $D = 1$, $M = 1$, $m = 10^{-3}$, and $N = 10^5$.*

an opportunity to construct new forms of $\kappa(\tau)$ which have the limiting behaviour needed for the Dirac delta approximation. If we restrict ourselves to even functions which decay quickly, we can calculate their Fourier series in the domain $[-L, L]$

$$\kappa(\tau) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_i \cos\left(\frac{i\pi\tau}{L}\right),$$

$$a_0 = \frac{1}{L}\int_{-L}^{L} \kappa(\tau)\mathrm{d}\tau \quad , \quad a_i = \frac{1}{L}\int_{-L}^{L} \kappa(\tau)\cos\left(\frac{i\pi\tau}{L}\right)\mathrm{d}\tau. \tag{2.57}$$

An obvious initial choice would be the Fourier series for the Dirac delta function, where $a_i = 1/L, i = 0, 1, 2, ...$no matter what the choice of domain is

$$\delta(t) = \frac{1}{2L} + \frac{1}{L}\sum_{i=1}^{\infty} \cos\left(\frac{i\pi\tau}{L}\right). \tag{2.58}$$

To achieve this, we set the frequencies and interactions such that

$$\omega_i = \frac{i\pi}{L} \quad \text{for} \quad i = 1, 2, ...$$
$$\alpha_i = \frac{1}{\omega_i}\sqrt{\frac{\gamma}{mL}}.$$

This results in the following expression for $\kappa(\tau)$

$$\kappa(\tau) = \frac{\gamma}{L}\sum_{i=1}^{\infty} \cos\left(\frac{i\pi\tau}{L}\right) \approx \gamma\delta(\tau) - \frac{\gamma}{2L}. \tag{2.59}$$

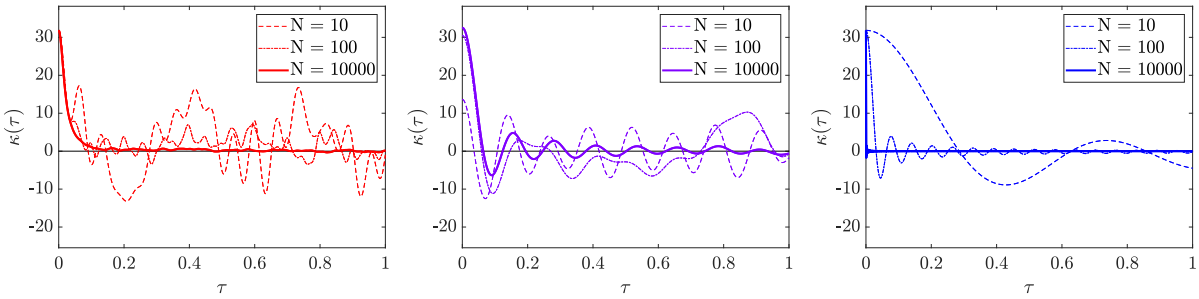


Figure 2.9: *Comparison of friction kernels for different choices of solvent particle oscillator numbers. In the first panel (red) frequencies are sampled from an exponential distribution with mean $\overline{\omega}$ and expected theoretical kernel given by equation* (2.44)*. In the second panel (purple) frequencies are sampled from a uniform distribution, with expected kernel given by equation* (2.56)*. In the third panel (blue) frequencies are taken from discrete series dictated by Fourier coefficient of the Dirac delta function, resulting in equation* (2.59)

Now, using equation (2.54) and substituting this in gives us

$$M\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = \boldsymbol{\xi}(t) - \gamma\boldsymbol{V}(t) - \frac{\gamma}{2L}\boldsymbol{X}(t). \tag{2.60}$$

If we keep things as they are, the final term essentially acts as a spring, bringing the target particle back to its initial position (in our derivation this is the origin). To avoid this we can set the value of L such that $\gamma \ll L$, allow us to neglect this term, returning the Langevin equation. We can generalise this to set $\kappa(\tau)$ as a function for which we can find a Fourier series over a given domain L, with Fourier coefficient from equation (2.57)

$$\omega_i = \frac{i\pi}{L} \quad \text{for} \quad i = 1, 2, ...$$
$$\alpha_i = \frac{1}{\omega_i}\sqrt{\frac{\gamma a_i}{m}},$$
$$\kappa(\tau) \approx \gamma\delta(\tau) - \frac{\gamma a_0}{2}.$$

where in this case we must have $\gamma a_0 \ll 1$ to be able to neglect the extra term. By following this procedure, it is possible to construct more theoretical Hamiltonian models which behave like a Langevin model (or a generalised Langevin model) when large enough numbers of oscillator particles are simulated. In Figure 2.9 we see examples of the rate of convergence as the number of solvent particles are increased, choosing timescales and parameters that allow easy comparison. It is clear, especially

from the exponential and uniform cases, that there is dramatically different memory effects via the friction kernel between these models at lower solvent numbers. Using the Langevin equation as a coarse-grained version of these models loses these subtleties, so in the coming sections we will instead solve equation (2.54) in full.

2.2.2 Non-Markovian random noise

Previously, we have exclusively considered cases where we had a fast decaying friction kernel, allowing us to use a Dirac delta approximation to simplify the friction integral term in equation (2.54) when moving from our detailed interaction models to the coarse-grained stochastic Langevin model. We turn to situations where this is not the case, and account for memory effects, i.e. having the friction term depend on past particle velocities, rather than simply the current velocity. The generalised Langevin model for our simple one particle system is given by

$$M\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = U(\boldsymbol{X}) + \boldsymbol{\xi}(t) - \int_0^t \kappa(\tau)\boldsymbol{V}(t-\tau)\mathrm{d}\tau,$$

with noise term $\boldsymbol{\xi}(t)$, friction kernel $\kappa(t)$, and potential energy $U(\boldsymbol{X})$, which we will initially set to zero. For a simple Langevin model we can easily simulate memory-less, Markovian random noise and use the Euler-Maruyama algorithm, or variants of it [131, 132], to solve our equations. However, for the generalised model we have two additional challenges:

- The friction integral $\int_0^t \kappa(\tau)\boldsymbol{V}(t-\tau)\mathrm{d}\tau$ must be explicitly calculated at each simulation timestep, as opposed to the Dirac delta approximation reducing it to the easily calculable $\gamma\boldsymbol{V}(t)$,
- The noise cannot simply be sampled from a Gaussian distribution, due to the more complicated form of the fluctuation-dissipation theorem.

When evaluating the friction integral, it is best to consider forms of $\kappa(\tau)$ which decay reasonably quickly. This still make physical sense as we would still generally

expect events long in the past to have less effect on dynamics than recent events. If there exists some τ_c such that $\kappa(\tau) \ll 1$ for $\tau > \tau_c$, we can cut off the integral

$$\int_0^t \kappa(\tau)\boldsymbol{V}(t-\tau)\mathrm{d}\tau \approx \int_0^{\tau_c} \kappa(\tau)\boldsymbol{V}(t-\tau)\mathrm{d}\tau.$$

By doing this we can still investigate cases with non trivially decaying kernels, with interesting consequences on the dynamics of a system, without having an algorithm which becomes more expensive to evaluate with time. We can find the numerical solution of the integral by using the trapezoidal rule, or a more accurate Newton-Cotes method (e.g Simpson's rule, Boole's rule). This increases the complexity of the simulations in a manner proportional to $\tau_c/\Delta t$, the number of past velocity terms considered when calculating the integral.

Having addressed the friction integral, we turn to the noise term $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]$, which is related to the friction kernel by a fluctuation-dissipation theorem

$$\lim_{t\to\infty} \langle \xi^i(t)\, \xi^j(t-\tau)\rangle = D\gamma\, \delta_{i,j}\, \kappa(\tau).$$

Previously, as $\kappa(\tau) = \gamma\delta(\tau)$, this relation resulted in Markovian (independence from previous noise in time) Gaussian noise. However, when we have a non-trivially decaying kernel, the noise becomes a time dependent quantity. If again we restrict ourselves to kernels which decay reasonably quickly, we can design a scheme where the noise at time t is dependent on the noise are previous times back to $t - \tau_c$.

Letting the noise at each timestep be defined as $\boldsymbol{\xi}_i = \boldsymbol{\xi}(i\Delta t)$, we can postulate that the following expressions hold for some constants c_i, with the cutoff time $\tau_c = I\Delta t$

$$\boldsymbol{\xi}_i = c_0\boldsymbol{\xi}\sqrt{D\gamma\kappa(0)} + \sum_{j=1}^{I} c_j\boldsymbol{\xi}_{i-j}, \tag{2.61}$$

$$\langle \boldsymbol{\xi}_{i-j}\rangle = 0 \quad \text{for,} \quad \langle \boldsymbol{\xi}_i\boldsymbol{\xi}_{i-j}\rangle = D\gamma\kappa(j\Delta t), \quad j = 0, ..., I, \tag{2.62}$$

In equation (2.61) we see how the noise for the current timestep depends on the previous noise values, along with newly sampled noise $\boldsymbol{\xi}$, whose components are normally distributed $\sim N(0, 1)$. Equation (2.62) gives the statistical properties of the noise, with zero mean and correlation determined by a fluctuation-dissipation

theorem. To simulate the correlated noise values for these coefficients must be found in so that the above expression hold, as is seen in [141], where the expressions for $\langle \boldsymbol{\xi}_i \boldsymbol{\xi}_{i-k} \rangle, k = 0, \ldots, I$ are used to find equations for c_j

$$A_{jk} = \kappa(|j-k|\Delta t), \quad b_k = \kappa(k\Delta t), \tag{2.63}$$

$$A_{kj}c_j = b_k, \quad j = 1, ..., I, \quad c_0 = \sqrt{1 - \frac{1}{b_0}\sum_{j=1}^{I} c_j b_j}. \tag{2.64}$$

The linear system can be solved efficiently to find the c_j for $j > 0$ using a number of different methods, such as Gaussian elimination or Cholseky decomposition (as the matrix A_{jk} is symmetric), with the first coefficient c_0 easily calculated afterwards. However, for some choices of $\kappa(\tau)$ the linear system for the coefficients can be hard to solve accurately, with the matrix A_{ij} often being close to singular even in cases where we have reasonable decay rates.

A more robust method of calculating correlated noise is presented in [142], and applied to the generalised Langevin model in [70, 143], where in a similar manner to the above, a set of coefficients c_i is calculated that determines the time dependence of the noise based on the form of the kernel. Using similar notation as previously, we let

$$\boldsymbol{\xi}_i = \sqrt{D\gamma} \sum_{j=-I}^{I} c_j \boldsymbol{\xi}'_{i-j}, \tag{2.65}$$

where $\boldsymbol{\xi}_i$, $i = 0, ...$ is our desired correlated noise at time $i\delta t$, and the components of $\boldsymbol{\xi}'_i$, $i = -I+1, ...$ are sequences of unit normally distributed random variables. For this expression to hold, the constraints must satisfy

$$\kappa_j = \sum_{k=-I}^{I} c_k c_{j-k}, \tag{2.66}$$

where $c_{j-k} = c_{j-k-I+1}$ if $j - k < 0$. To do this we take the discrete Fourier transform of the kernel sequence

$$\hat{\kappa}_k = \sum_{j=-I}^{I} \kappa_j e^{-2\pi i(jk/(2I+1))}. \tag{2.67}$$

Letting $\hat{c}_k = \sqrt{\hat{\kappa}_k}$ we can finally calculate the coefficients by taking the inverse discrete Fourier transform

$$c_j = \frac{1}{2I+1} \sum_{k=-I}^{I} \hat{c}_k e^{2\pi i(jk/(2I+1))}. \tag{2.68}$$

We can see that this is consistent by substituting our result for the coefficients back into equation (2.66) to recover κ_j

$$\begin{aligned}
\sum_{k=-I}^{I} c_k c_{j-k} &= \sum_{k=-I}^{I} \left(\frac{1}{2I+1} \sum_{m=-I}^{I} \hat{c}_m e^{2\pi i (mk/(2I+1))} \right) \left(\frac{1}{2I+1} \sum_{n=-I}^{I} \hat{c}_n e^{2\pi i (n(j-k)/(2I+1))} \right) \\
&= \frac{1}{(2I+1)^2} \sum_{n,m,k=-I}^{I} \sqrt{\hat{\kappa}_m \hat{\kappa}_n} e^{2\pi i ((nj+k(m-n))/(2I+1))} \\
&= \frac{1}{2I+1} \sum_{n,m=-I}^{I} \sqrt{\hat{\kappa}_m \hat{\kappa}_n} \delta_{nm} e^{2\pi i (nj/(2I+1))} \\
&= \frac{1}{2I+1} \sum_{n=-I}^{I} \hat{\kappa}_n e^{2\pi i (nj/(2I+1))} = \kappa_j.
\end{aligned}$$

In Figure 2.10 when the friction kernel decays more slowly, as a result of the fluctuation dissipation theorem, we see that the noise becomes less disordered, with clear, continuous paths for $\boldsymbol{\xi}$ appearing. For each of the plots the diffusion D is kept constant, and we see that to maintain the same kinetic energy at equilibrium the variance of the noise must become smaller as it becomes more correlated. By using methods to generate correlated noise determined by a time dependent kernel, in conjunction with method for numerically integrating friction term we can now solve the full generalised Langevin equation. While this is more computationally expensive than the simple Langevin model, these implementations are still less taxing than a full Hamiltonian simulation and can be seen as a middle ground between them. In the examples that we present in the rest of this section, we simulate the generalised Langevin using the method described in [70], which is based on the form for the basic Langevin model presented in [132], with correlated noise $\boldsymbol{\xi}(t)$ sampled by equations (2.65)-(2.68)

$$\begin{aligned}
\mathbf{X}(t+\Delta t) &= \mathbf{X}(t) + \frac{1}{1+\frac{\kappa(0)\Delta t}{4M}} \left(\mathbf{V}(t) - \frac{1}{2M}\mathbf{K}(t)\Delta t + \frac{1}{2M}\boldsymbol{\xi}(t)\sqrt{\Delta t} \right) \Delta t, \\
\mathbf{V}(t+\Delta t) &= \frac{1-\frac{\kappa(0)\Delta t}{4M}}{1+\frac{\kappa(0)\Delta t}{4M}} \mathbf{V}(t) + \frac{1}{1+\frac{\kappa(0)\Delta t}{4M}} \left(-\frac{1}{M}\mathbf{K}(t)\Delta t + \frac{1}{M}\boldsymbol{\xi}(t)\sqrt{\Delta t} \right), \qquad (2.69) \\
\mathbf{K}(t) &= \sum_{i=1}^{I} \kappa(i\Delta t) \left(\boldsymbol{X}(t-(i-1)\Delta t) - \boldsymbol{X}(t-i\Delta t) \right).
\end{aligned}$$

As we are currently purely considering generalised Langevin dynamics, this will offer good stability, with better energy conservation than the BBK integrator [131].

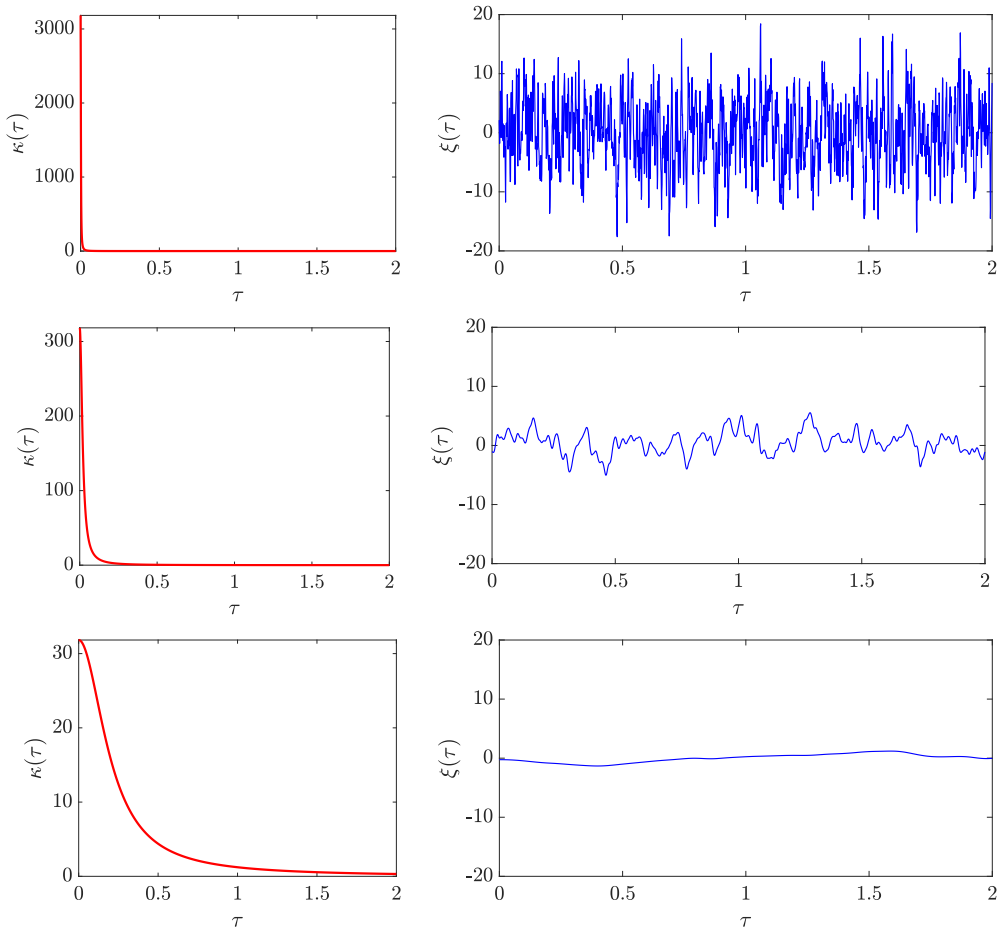


Figure 2.10: *Effect of rate of decay of friction kernel $\kappa(\tau)$ on properties of noise sampled using equation* (2.65) *in accordance with the fluctuation-dissipation theorem. Kernel used is from Section* 2.2.1 *derived from Hamiltonian model in Section* 2.1.3*, defined by equation* (2.44)*. Mean of exponential distribution is varied for the three rows, with $\overline{\omega} = 500, 50, 5$ reading down the figure. Shared parameters between all panels are $D = 1$, $\gamma = 10$, $I = 10^4$, $\Delta t = 10^{-3}$*

However, in Section 2.2.3 we will return to the issues that arise with this method when constructing a hybrid multi-resolution algorithm.

We will now turn to testing this method on the friction kernels for the exponential and uniform frequency distributions of oscillators and see if we can reproduce the correct long term statistics, such as the velocity autocorrelation function, for less restrictive choices of $\overline{\omega}$ and ω_d. For the simple Langevin model we have

$$C_m(\tau) = \lim_{t\to\infty} \frac{1}{3} \left\langle \mathbf{V}(t+\tau) \cdot \mathbf{V}(t) \right\rangle, = D\gamma \exp[-\gamma\tau]. \tag{2.70}$$

However, as we are now dealing with the generalised form we must instead

[GLE1]	Sample correlated non-Markovian noise $\boldsymbol{\xi}(t)$ using equations (2.65)-(2.66) with a cut off time of $\tau_c = I\Delta t$.
[GLE2]	Calculate friction integral $\boldsymbol{K}(t)$ specified in (2.69) using trapizoidal rule and past positions back to cut off time $\tau_c = I\Delta t$.
[GLE3]	Using these terms, calculate new positions and velocities using equation (2.69).
[GLE4]	Continue with step [GLE1] using time $t = t + \Delta t$.

Table 2.5: *One iteration of generalised Langevin model simulation algorithm for single monomer*

numerically calculate the velocity autocorrelation by solving

$$\frac{\mathrm{d}C_m(t)}{\mathrm{d}t} = -\int_0^\tau \kappa(\tau)C_m(t-\tau)\mathrm{d}\tau, \tag{2.71}$$

and use this to validate the results from our simulations, rather than use equation (2.70). This is known as a Volterra integro-differential equation of the first kind, and are known to be tricky to solve, both analytically and numerically with accuracy.

Beginning with the numerical route, we must make discretisation choices for both the differentiated term and the integral term. There are a multitude of schemes described in the literature [144, 145], with a simple option being to start with the trapezoidal rule, letting $C_{m;i} = C_m(i\Delta t)$ and $\kappa_i = \kappa(i\Delta t)$

$$C_{m;i+1} = C_{m;i}\left(1 - \frac{\kappa_0}{2}(\Delta t)^2\right) - \frac{1}{2}(\Delta t)^2\sum_{j=1}^{i}\kappa_j C_{m;i-j}. \tag{2.72}$$

Having set out a numerical scheme, we will now attempt to see if it is possible to find the analytic solution for comparison. One option is to take Laplace transforms [146] of both sides of the Volterra equation, with $\mathcal{L}_s[C_m(\tau)] = \hat{C}_m(s)$ and $\mathcal{L}_s[\kappa(\tau)] = \hat{\kappa}(s)$

$$s\hat{C}_m(s) - C_m(0) = -\hat{C}_m(s)\hat{\kappa}(s).$$

Rearranging this and taking the inverse Laplace transform gives us an expression for the velocity autocorrelation [69, 141]

$$C_m(\tau) = \mathcal{L}_\tau^{-1}\left[\frac{C_m(0)}{s + \hat{\kappa}(s)}\right]. \tag{2.73}$$

For our previous choices of kernel in Section 2.2.1 the expressions for the Laplace transform lead to a form equation (2.73) that is not easily invertible and would

require a further numerical method to solve the Laplace integral. Instead, if we consider a simple test case that decays exponentially, with an easily calculable Laplace transform, we can find a complete analytic solution.

$$\kappa(\tau) = \gamma\beta \exp(-\beta\tau) \quad , \quad \int_0^\infty \kappa(\tau)\mathrm{d}\tau = \gamma \quad , \quad \mathcal{L}_s[\kappa(\tau)] = \frac{\gamma\beta}{s+\beta}. \tag{2.74}$$

Substituting this into our expression for the velocity autocorrelation function gives

$$\begin{aligned} C_m(\tau) &= \mathcal{L}_\tau^{-1}\left[\frac{C_m(0)(s+\beta)}{s^2+\beta s+\gamma\beta}\right] \\ &= C_m(0)\mathcal{L}_\tau^{-1}\left[\frac{A^+}{s-s_+} + \frac{A^-}{s-s_-}\right] \\ &= C_m(0)\left[A^+\exp(s_+\tau) + A^-\exp(s_-\tau)\right], \end{aligned} \tag{2.75}$$

where s_+ and s_- are the roots of $s^2+\beta s+\gamma\beta = 0$ and A^+ and A^- are constants appearing from splitting the expression into partial fraction, defined below

$$\begin{aligned} s_+ &= \frac{-\beta}{2} + \frac{1}{2}\sqrt{\beta^2 - 4\beta\gamma}, \\ s_- &= \frac{-\beta}{2} - \frac{1}{2}\sqrt{\beta^2 - 4\beta\gamma}, \\ A^+ &= \frac{\beta+s_+}{s_+-s_-}, \\ A^- &= \frac{\beta+s_-}{s_--s_+}. \end{aligned}$$

This is an interesting test case as if $\beta^2 - 4\beta\gamma > 0$ then we have simple exponential decay of the velocity autocorrelation, whereas if $\beta^2 - 4\beta\gamma < 0$ as $\beta > 0$ we can find an oscillating, exponentially decaying form of $C_m(\tau)$, with closed form defined below

$$C_m(\tau) = C_m(0)\exp\left(\frac{-\beta\tau}{2}\right)\left[\frac{\beta}{\alpha}\sin\left(\frac{\alpha\tau}{2}\right) + \cos\left(\frac{\alpha\tau}{2}\right)\right],$$

where in this case $\alpha = \sqrt{4\beta\gamma - \beta^2}$ is real [141]. It is possible to extend this analysis further include a kernel that itself oscillates, which would have similar properties to equation (2.56), taking the form

$$\begin{aligned} \kappa(\tau) = \frac{\gamma(\beta_1^2+\beta_2^2)}{\beta_1}\exp(-\beta_1\tau)\cos(\beta_2\tau) \quad , \quad \int_0^\infty \kappa(\tau)\mathrm{d}\tau = \gamma \\ \mathcal{L}_s[\kappa(\tau)] = \frac{\gamma(\beta_1^2+\beta_2^2)}{\beta_1}\left(\frac{s+\beta_1}{(s+\beta_1)^2+\beta_2^2}\right), \end{aligned} \tag{2.76}$$

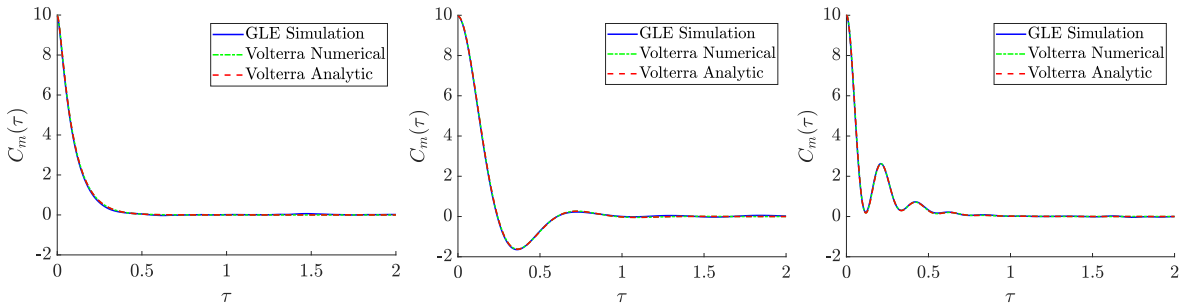


Figure 2.11: *Demonstration of efficacy of numerical scheme to solve Volterra integro-differential equations for the velocity auto-correlation function $C_m(\tau)$, performing well for a variety of auto-correlation function behaviours. Friction kernel $\kappa(\tau)$ considered is from equation* (2.76)*, with $(\beta_1, \beta_2) = (100, 0)$ for the first panel, $(10, 0)$ for the second, and $(10, 20)$ for the third. We calculated $C_m(\tau)$ from simulated Generalised Langevin equation data (blue), numerical method defined in equation* (2.72) *(green), and analytic solution in equation* (2.77)*. Shared parameters between two panels are simulation time of* 100 *dimensionless units,* 100 *repeats, $D = 1$, $\gamma = 10$, $I = 10^3$, $\Delta t = 10^{-3}$*

giving a less appealing, but still solvable, expression for the autocorrelation function Laplace transform by finding the general solution for the partial fractions that arise when we substitute this into equation (2.73).

$$\begin{aligned} C_m(\tau) &= \mathcal{L}_\tau^{-1}\left[\frac{C_m(0)(s^2 + 2\beta_1 s + (\beta_1^2 + \beta_2^2))}{s^3 + 2\beta_1 s^2 + (\beta_1^2 + \beta_2^2)(\gamma/\beta_1 + 1)s + \gamma(\beta_1^2 + \beta_2^2)}\right] \\ &= C_m(0)\mathcal{L}_\tau^{-1}\left[\frac{A_1}{s - s_1} + \frac{A_2}{s - s_2} + \frac{A_3}{s - s_3}\right] \\ &= C_m(0)\left[A_1 \exp(s_1\tau) + A_2 \exp(s_2\tau) + A_3 \exp(s_3\tau)\right]. \end{aligned} \tag{2.77}$$

This can be solved analytically by using a cubic formula (e.g. Cardano method) to find the roots s_j for $j = 1, 2, 3$ (though this time we will omit their exact forms in terms of β_1, β_2, and γ) and solving the linear system below to find A_j for $j = 1, 2, 3$

$$\begin{bmatrix} 1 & 1 & 1 \\ -(s_2 + s_3) & -(s_3 + s_1) & -(s_1 + s_2) \\ s_2 s_3 & s_3 s_1 & s_1 s_2 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2\beta_1 \\ \beta_1^2 + \beta_2^2 \end{bmatrix}$$

In Figure 2.11, we see that both the numerical and analytic solutions to the Volterra integro-differential equation compare well with the simulated correlation function $C_m(\tau)$ for both the simple and oscillating cases. In the first two panels $\beta_2 = 0$ and equation (2.76) simplifies down to equation (2.74), with $\beta_1 < 4\gamma$ in the first, and $\beta_1 > 4\gamma$ in the second, while the third is an example of $C_m(\tau)$ for an oscillating kernel. This gives us confidence to now apply the numerical solving

method to the friction kernels from Section 2.2.1. While these kernels seem to the naked eye very similar, both equation (2.44) and equation (2.56) decay to zero much slower than equation (2.76), due to the fact that they are power law expressions. The resulting long tail behaviour has significant implications on the the memory effects, meaning that a larger kernel window $\tau_c = I\Delta t$ has to be used in simulations to not underestimate either the noise and friction terms of the generalised Langevin equation.

Taking all of this into account, we can now simulate the generalised Langevin equation using the kernels from Section 2.2.1, avoiding restrictions on the choice of $\overline{\omega}$ for the exponentially sampled oscillator frequencies, and ω_d for the uniformly sampled frequencies. In Figure 2.12 we see that the simulated values for the velocity autocorrelation match up with our theoretical expectations. In the bottom panel, we show that the generalised Langevin model reduces to a simple Langevin model when $\omega_d, \overline{\omega} \to \infty$, reproducing the exact solution from equation (2.70). On top of this, we see that for various smaller values of $\omega_d, \overline{\omega}$ the generalised Langevin equation is still consistent with the equipartition theorem $C_m(0) = D\gamma$ and when $C_m(\tau)$ is fully calculated from simulation data it matches the results from the numerical Volterra solution to equation (2.71). There are some discrepancies seen in the second panel for larger values of τ, and this can be put down to the fact that the friction kernel for exponentially distribution oscillator frequencies decays $\mathrm{O}(1/\tau^2)$, while the kernel for the uniform frequencies decays $\mathrm{O}(1/\tau)$. This slower rate of decay (especially for small values of ω_d) means a very long cutoff time $\tau_c = I\Delta t$ would have to be used to capture the point when $\kappa(\tau) \approx 0$. We can now relax the restriction on choices of $\overline{\omega}$ during our analysis and use the generalised Langevin model to coarse-grain the Hamiltonian model. In the final section of this chapter we will look at how we can directly establish the parameters for the GLE model simply from data derived from the Hamiltonian model, opening the possibility for more general heat baths to be employed, without an underlying mathematical link at each stage of the analysis.

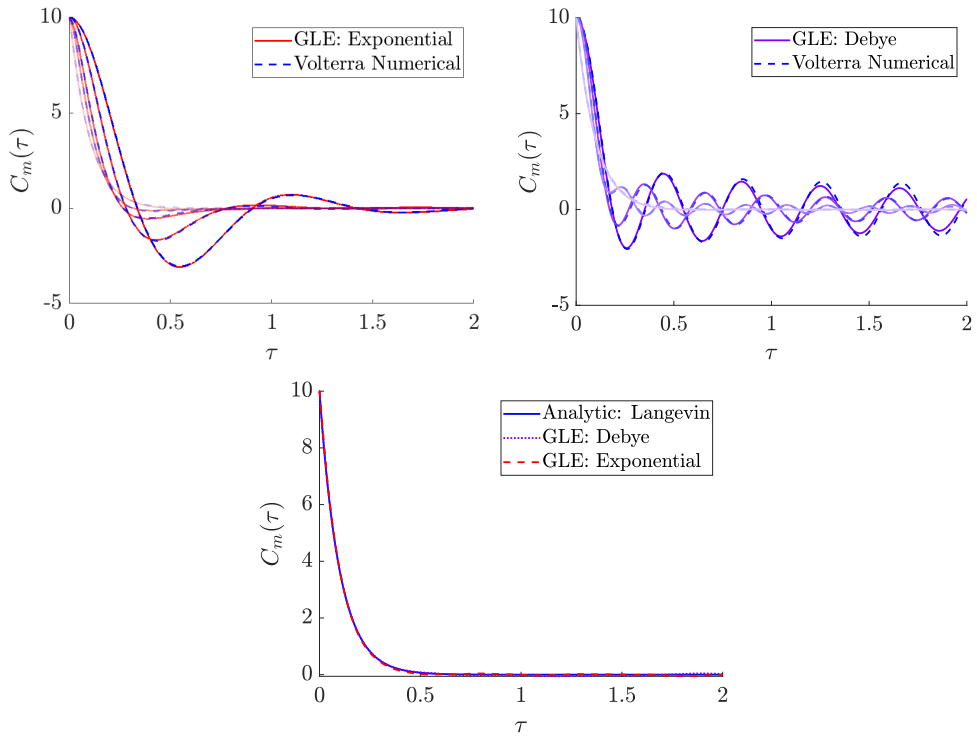


Figure 2.12: *Simulating generalised Langevin equation friction kernels derived from Hamiltonian dynamics. In first panel we see the velocity auto-correlation $C_m(\tau)$ when $\kappa(\tau)$ is from equation (2.44) for $\overline{\omega} = 5, 10, 25, 50, 500$. Solid blue lines are calculated from simulation data and dashed red lines from numerical Volterra method from equation (2.72), with higher transparency for increasing $\overline{\omega}$. Second panel is identical to first, except $\kappa(\tau)$ is from equation (2.56) for $\omega_d = 20, 30, 500$. Solid blue lines are calculated from simulation data and dashed purple lines from numerical Volterra method, with higher transparency for increasing ω_d. Bottom panel shows that generalised Langevin model reduces to simple Langevin expression for $C_m(\tau)$ (equation (2.70)) in Dirac delta limits, with $\overline{\omega} = 1000$, $\omega_d = 1000$ Shared parameters between three panels are simulation time of 100 dimensionless units, 100 repeats, $D = 1$, $\gamma = 10$, $I = 10^4$, $\Delta t = 10^{-3}$*

2.2.3 Kernel estimation

Now that we have established the efficacy of some solving methods for the generalised Langevin model, we now consider methods to estimate the friction kernel from MD simulation data rather than using the analytic links we have previously described. In the reference [71] the kernel and noise of a generalised Langevin model are estimated, using a numerical scheme, from the time-correlation functions calculated from simulation data. Beginning with the generalised Langevin equation in the form

$$M\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = \boldsymbol{\xi}(t) - \int_0^t \kappa(\tau)\boldsymbol{V}(t-\tau)\mathrm{d}\tau,$$

with noise term $\boldsymbol{\xi}(t)$ and kernel $\gamma(\tau)$, with the noise obeying the following fluctuation-dissipation theorem

$$\langle \boldsymbol{\xi}(t)\boldsymbol{\xi}(\tau)\rangle = D\gamma\kappa(t-\tau). \tag{2.78}$$

By considering the derivatives of the velocity auto-correlation function, it can be shown that [71]

$$C_m(t) = \langle \boldsymbol{V}(t)\boldsymbol{V}\rangle, \tag{2.79}$$

$$M\dot{C}_m(t) = -\int_0^t C_m(t-\tau)\kappa(\tau)\mathrm{d}\tau, \tag{2.80}$$

$$M\ddot{C}_m(t) = -C_m(0)\kappa(t) - \int_0^t \dot{C}_m(t-\tau)\kappa(\tau)\mathrm{d}\tau. \tag{2.81}$$

In Section 2.2.2 we considered the solutions of Volterra integro-differential equations of the first kind, but this time we would be solving for $\kappa(\tau)$, rather than $C_m(\tau)$, making it difficult to accurately numerically solve equation (2.80). For this reason, solving this Volterra integro-differential equation of the second kind, equation (2.81) kind becomes an interesting prospect. One option would be to calculate the velocity autocorrelation function, then numerically differentiate this twice, though this can introduce numerical errors so may not be appropriate in this case. Alternatively, we can calculate the force-velocity correlation and the force autocorrelation functions and note that

$$C_m(t) = \langle \boldsymbol{V}(t)\boldsymbol{V}\rangle = \lim_{\tau\to\infty}\frac{1}{3}\langle \mathbf{V}(t+\tau)\cdot\mathbf{V}(\tau)\rangle, \tag{2.82}$$

$$\dot{C}_m(t) = \langle \boldsymbol{F}(t)\boldsymbol{V}\rangle = \lim_{\tau\to\infty}\frac{1}{3}\langle \mathbf{F}(t+\tau)\cdot\mathbf{V}(\tau)\rangle, \tag{2.83}$$

$$\ddot{C}_m(t) = -\langle \boldsymbol{F}(t)\boldsymbol{F}\rangle = \lim_{\tau\to\infty}\frac{1}{3}\langle \mathbf{F}(t+\tau)\cdot\mathbf{F}(\tau)\rangle. \tag{2.84}$$

Substituting equations (2.82)-(2.84) into equation (2.81) then gives us

$$M\langle \boldsymbol{F}(t)\boldsymbol{F}\rangle = C_m(0)\kappa(t) + \int_0^t \langle \boldsymbol{F}(t-\tau)\boldsymbol{V}\rangle\kappa(\tau)\mathrm{d}\tau. \tag{2.85}$$

We can then discretise and invert this expression derived from the second order Volterra equation to find the value of the kernel at each timestep we have calculated the correlation functions

$$\kappa(i\Delta t) = \frac{M}{C_m(0)+\omega_i\langle \boldsymbol{F}\boldsymbol{V}\rangle\Delta t}\left[\langle \boldsymbol{F}(i\Delta t)\boldsymbol{F}\rangle - \frac{\Delta t}{M}\sum_{j=0}^{i-1}\omega_j\langle \boldsymbol{F}((i-j)\Delta t)\boldsymbol{V}\rangle\kappa(j\Delta t)\right],$$

where $\omega_j = 1/2$ for $j = 0, i$ and $\omega_j = 1$ otherwise, arising from use of the trapizoidal rule to evaluate the integral, and initial condition $\gamma(0) = \langle \boldsymbol{\xi}^2 \rangle / C(0)$.

We now consider a new higher order discretisation of this method constructed by using Simpson's rule for the integral term. We will have to be careful with the end cases when the number of steps taken is not odd. Using the trapezoidal rule for the final step would be a simple solution, but this would reduce the accuracy of the solving method, and instead it is better to use Simpson's 3/8 rule on the final 4 terms of the expression to maintain the order or accuracy. We have the following initial condition and special cases for first couple of timesteps, where κ_1 is given by the trapezoidal rule over a single interval Δt, κ_2 by Simpson's rule over interval $2\Delta t$, and κ_3 by Simpson's 3/8 rule over interval $3\Delta t$

$$\begin{aligned}
\kappa_0 &= \frac{M\langle \boldsymbol{F}^2 \rangle}{C_m(0)}, \\
\kappa_1 &= \frac{M}{C_m(0) + \frac{1}{2}\Delta t \langle \boldsymbol{FV} \rangle} \left[\boldsymbol{FF}_1 - \frac{\Delta t}{2M} \boldsymbol{FV}_1 \kappa_0 \right], \\
\kappa_2 &= \frac{M}{C_m(0) + \frac{1}{3}\Delta t \langle \boldsymbol{FV} \rangle} \left[\boldsymbol{FF}_1 - \frac{\Delta t}{2M} (4\boldsymbol{FV}_1 \kappa_1 + \boldsymbol{FV}_2 \kappa_0) \right], \\
\kappa_3 &= \frac{M}{C_m(0) + \frac{3}{8}\Delta t \langle \boldsymbol{FV} \rangle} \left[\boldsymbol{FF}_1 - \frac{3\Delta t}{8M} (3\boldsymbol{FV}_1 \kappa_2 + 3\boldsymbol{FV}_2 \kappa_1 + \boldsymbol{FV}_3 \kappa_0) \right].
\end{aligned} \tag{2.86}$$

where $\kappa_i = \kappa(i\Delta t)$, $\boldsymbol{FF}_i = \langle \boldsymbol{F}(i\Delta t)\boldsymbol{F} \rangle$, and $\boldsymbol{FV}_i = \langle \boldsymbol{F}(i\Delta t)\boldsymbol{V} \rangle$. After these special cases, we can write the general forms of the discretisation for the kernel for even and odd cases:

$$\begin{aligned}
\kappa_{2n} = &\frac{M}{C_m(0) + \frac{1}{3}\Delta t \langle \boldsymbol{FV} \rangle} \Bigg[\boldsymbol{FF}_{2n} \\
&- \frac{\Delta t}{3M} \left(4 \sum_{j=0}^{n-1} \boldsymbol{FV}_{2j+1} \kappa_{2(n-j)-1} + 2 \sum_{j=1}^{n-1} \boldsymbol{FV}_{2j} \kappa_{2(n-j)} + \boldsymbol{FV}_{2n} \kappa_0 \right) \Bigg],
\end{aligned} \tag{2.87}$$

$$\begin{aligned}
\kappa_{2n+1} = &\frac{M}{C_m(0) + \frac{1}{3}\Delta t \langle \boldsymbol{FV} \rangle} \Bigg[\boldsymbol{FF}_{2n+1} \\
&- \frac{\Delta t}{3M} \left(4 \sum_{j=0}^{n-2} \boldsymbol{FV}_{2j+1} \kappa_{2(n-j)} + 2 \sum_{j=1}^{n-2} \boldsymbol{FV}_{2j} \kappa_{2(n-j)+1} + \boldsymbol{FV}_{2n-2} \kappa_3 \right) \\
&- \frac{3\Delta t}{8M} \left(\boldsymbol{FV}_{2n-2} \kappa_3 + 3\boldsymbol{FV}_{2n-1} \kappa_2 + 3\boldsymbol{FV}_{2n} \kappa_1 + \boldsymbol{FV}_{2n+1} \kappa_0 \right) \Bigg].
\end{aligned} \tag{2.88}$$

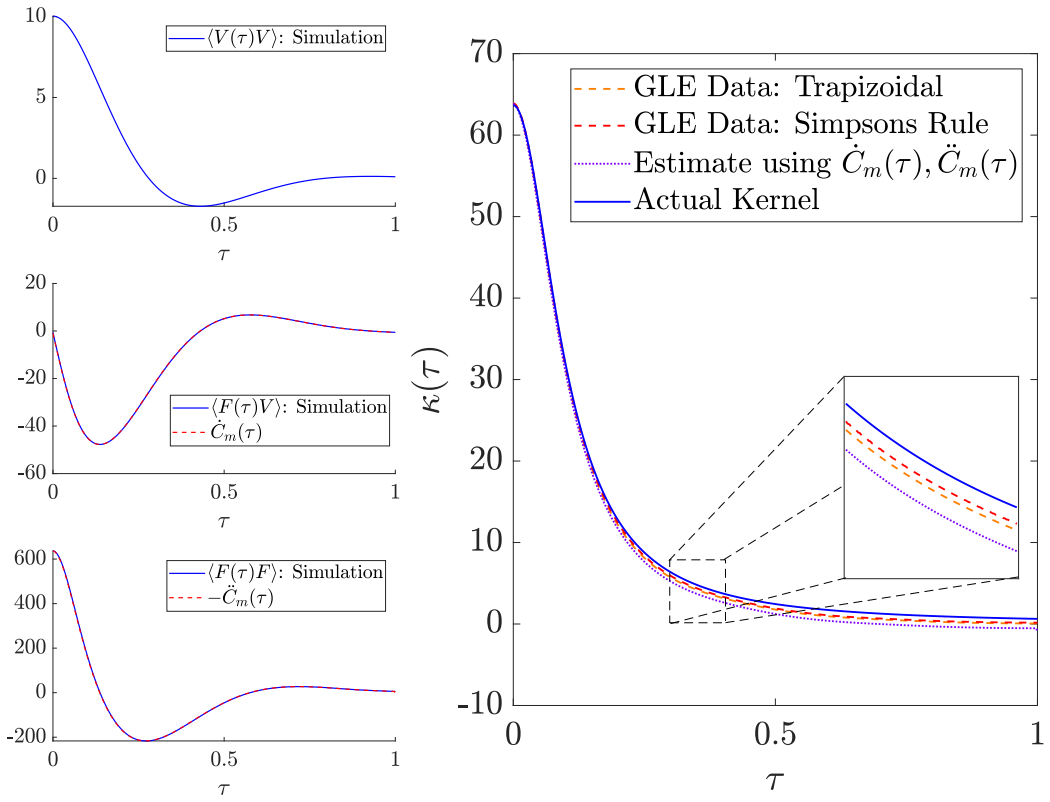


Figure 2.13: *Kernel estimation techniques used on generalised Langevin model simulation data. In the first column we have, from top to bottom, showing the velocity autocoreelation, force-velocity correlation, and the force autocorrelation functions. The blue lines in each are calculated from a GLE simulations using $\kappa(\tau)$ in equation* (2.44)*, and red lines in second and third graphs are found by numerically differentiating $C_m(\tau)$. Figure in second column shows $\kappa(\tau)$ used for GLE simulation and attempts to estimate it. Orange dashed line is estimation method when integral in equation* (2.85) *is discretised with trapezoidal method, while red dashed line uses Simpson's rule. Purple dotted line is when equation* (2.81) *is used, rather than correlation functions involving forces. Parameters used were simulation time of* 100 *dimensionless units,* 100 *repeats, $\varpi = 10$, $D = 1$, $\gamma = 10$, $I = 10^4$, $\Delta t = 10^{-3}$*

This means that for the first step we use the original method from [71], and thereafter we employ our new higher order method, which should improve the accuracy of the numerical estimation of the kernel without need to reduce size of the timestep. This could be useful in circumstances where a limited amount of data is available and there is no opportunity to repeat with smaller Δt. In Figure 2.13 we see how the simulation data for $\langle \boldsymbol{V}(\tau)\boldsymbol{V}\rangle$, $\langle \boldsymbol{F}(\tau)\boldsymbol{V}\rangle$, and, $\langle \boldsymbol{F}(\tau)\boldsymbol{F}\rangle$ from the generalised Langevin model simulated with friction kernel $\kappa(\tau)$ can be used to recover the same kernel by a variety of numerical methods. As expected, the higher order scheme

for estimating the integral term yields the most accurate results, with the least accurate results found when numerically differentiating the velocity autocorrelation function, rather than direct calculating of the force correlation functions.

Now that we have set up this framework, we can return to the Hamiltonian model from Section 2.1.2. By simulating and calculating the correlation functions directly from this data simulation data we can avoid making any approximations, or require any limits to hold, and reproduce the friction kernel $\kappa(\tau)$ to be used in the coarse-grained generalise Langevin model, rather than the simple Langevin model previously considered. In Section 2.2.2 we used an alternative (equation (2.69)) to the BBK integrator (equations (2.47)-(2.52)) to improve the accuracy of the generalised Langevin scheme. However, as we will be simultaneously integrating the Hamiltonian equations of motion it is advantageous to continue to use a Verlet style integrator for improved stability. As a compromise, we use a fusion of these schemes, with the integrator unchanged for the Hamiltonian model (equations (2.47)-(2.50),(2.52)). For the generalised Langevin bead of the dimer, we use the following velocity Verlet style method

$$\mathbf{V}_2\Big(t+\tfrac{1}{2}\Delta t\Big) = \mathbf{V}_2(t) + \frac{1}{2}\,\mathbf{A}_2(t) + \frac{1}{2}\,\mathbf{K}_+(t), \tag{2.89}$$

$$\mathbf{X}_2(t+\Delta t) = \mathbf{X}_2(t) + \mathbf{V}_2(t+\tfrac{1}{2}\Delta t)\,\Delta t, \tag{2.90}$$

$$\mathbf{V}_2(t+\Delta t) = \frac{1}{1+\frac{\kappa(0)\Delta t}{4M}}\left(\mathbf{V}_2\left(t+\tfrac{1}{2}\Delta t\right) + \frac{1}{2}\,\mathbf{A}_2(t+\Delta t) + \frac{1}{2}\,\mathbf{K}_-(t+\Delta t)\right), \tag{2.91}$$

where the forces for each stage of the discretisation defined below, with correlated noise $\boldsymbol{\xi}(t)$ sampled by equations (2.65)-(2.68)

$$\begin{aligned}
\Phi(R) &= \frac{k(R-\ell_0)^2}{2} \quad , \quad R = |\boldsymbol{R}| \quad , \quad \boldsymbol{R} = \boldsymbol{X}_2 - \boldsymbol{X}_1, \\
\mathbf{A}_2(t) &= \frac{\Phi'(R)}{M}\,\frac{\mathbf{R}}{R}\Delta t + \frac{1}{M}\boldsymbol{\xi}(t)\sqrt{\Delta t}, \\
\mathbf{K}_+(t) &= \frac{(\Delta t)^2}{M}\left(\frac{1}{2}\kappa(0)\mathbf{V}_2(t) + \sum_{i=1}^{I}\kappa(i\Delta t)\mathbf{V}_2(t-i\Delta t)\right), \\
\mathbf{K}_-(t) &= \frac{(\Delta t)^2}{M}\sum_{i=1}^{I}\kappa(i\Delta t)\mathbf{V}_2(t-i\Delta t).
\end{aligned} \tag{2.92}$$

The factor at the front of the second velocity timestep arises from the discrete representation of the integral being dependent on the velocity at $t+\Delta t$, with

[MR-GLE1]	Update velocity of dimer by a half timestep, using equation (2.47) for the first monomer, and equation (2.89) for the second monomer, with same values for $\boldsymbol{A}_i(t)$ as the previous timestep, and newly calculated $\boldsymbol{K}_+(t)$
[MR-GLE2]	Update position of dimer by a full timestep, using equation (2.48) for the first monomer, and equation (2.90) for the second monomer.
[MR-GLE3]	Update accelerations $\boldsymbol{A}_i(t+\Delta t)$ for new positions using equation (2.50) and equation (2.92). Sample correlated non-Markovian noise $\boldsymbol{\xi}(t+\Delta t)$ using equations (2.65)-(2.66) with a cut off time of $\tau_c = I\Delta t$.
[MR-GLE4]	Update velocity of dimer by second half timestep, using equation (2.49) for the first monomer, and equation (2.91) for the second monomer, using newly calculated $\boldsymbol{A}_i(t+\Delta t)$ and $\boldsymbol{K}_-(t)$.
[MR-GLE5]	Continue with step [MR-GLE1] using time $t = t + \Delta t$.

Table 2.6: *One iteration of the updated multi-resolution simulation algorithm of the dimer in the heat bath with long-range interactions, where the second monomer is now simulated by generalised Langevin dynamics.*

the extra factor of 1/2 from use of trapizoidal rule. While there is an extra function evaluation involved, with $\boldsymbol{K}_+$ and $\boldsymbol{K}_-$ needed for each timestep, this creates a negligible increase in complexity in the contest of the overall Hamiltonian-GLE multi-resolution model.

In Figure 2.14 we see the numerically calculated values for the correlation function lead to an estimate for $\kappa(\tau)$ which matches equation (2.44), as we would expect when sampling the oscillator frequencies from an exponential distribution of mean $\overline{\omega}$. In the final panel we implement a new multi-resolution dimer model, replacing the Langevin model used in Section 2.1.2 with the generalised Langevin model for coarse-graining. Despite using lower values for the mean of the oscillator frequencies, we find the velocity autocorrelation for this new model matches the theoretically expected forms calculated by solving equation (2.71) using the numerical scheme in equation (2.72).

2.3 Discussion

In Section 2.1, we used two theoretical heat baths. Although these heat baths are based on qualitatively different descriptions of solvent-dimer interactions, they both

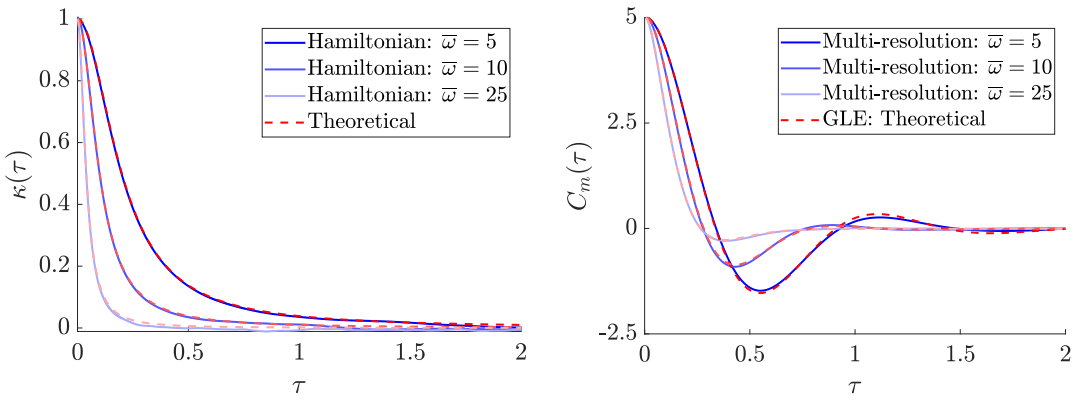


Figure 2.14: *Estimating friction kernel directly from Hamiltonian simulation data, resulting in updated Hamiltonian-GLE multi-resolution model. First panel shows kernel estimated using equations* (2.86)-(2.88) *from single particle Hamiltonian model simulation data (blue lines), compared against expected form of* $\kappa(\tau)$ *(red dashed lines, equation* (2.44)*) which have been normalised to* $\kappa(0) = 1$ *for clarity. Second panel shows velocity autocorrelation of multi-resolution dimer model (blue lines) with parameters for the spring coupling set as* $k = 10^6$,$\ell_0 = 0.32$. *Dimer initialised at origin at rest, with initial heat bath particle positions and velocities sampled from equilibrium distributions. These were compared against numerical solutions to equation* (2.71). *Shared parameters between panels were simulation time of* 100 *dimensionless units,* 100 *repeats,* $\overline{\omega} = 5, 10, 25$, $N = 10^5$, $D = 1$, $\gamma = 10$, $m = 10^-3$, $M = 1$, $I = 10^4$, $\Delta t = 10^{-3}$

lead to the Langevin description, given in equations (2.1)–(2.4), in a certain limit. In particular, we can use this limiting process to coarse-grain a part of the simulated dimer molecule, while use a detailed MD model to describe the rest of the molecule. Such a multi-resolution approach has potential to significantly speed up computer simulations of dynamics of macromolecules [92–94, 96], provided that it is combined with additional multiscale and multi-resolution methodologies, discussed below.

One possible strategy in this case is to couple a detailed MD model with a stochastic coarse-grained model which is written with the help of additional variables [134–136]. To improve the efficiency of simulations further, one can then coarse-grain such a generalised Langevin description using a Brownian dynamics approach [127, 147]. Brownian dynamics modelling can be further coupled with stochastic reaction-diffusion modelling based on lattice-based (compartment-based) methods [76]. Lattice-based models are very attractive for simulations of intracellular processes, because they enable modelling of spatio-temporal processes in the whole cell or its significant part [148]. Coupling Brownian dynamics with compartment-

based approaches has been used in a number of applications, including multi-resolution modelling of actin dynamics in filopodia [118, 149] or for modelling intracellular calcium dynamics [150].

In this chapter, we have investigated multi-resolution approaches, schematically described in Figure 2.1(a) and 2.1(b). Another class of multi-resolution approaches in the literature considers a fixed subdomain of the computational domain where a detailed modelling approach is used, which is coupled with a coarser model in the rest of the simulation domain [75, 76]. Such an approach is useful, for example, when modelling intracellular ion dynamics. Ions pass through an ion channel in single file and an MD model has to be used to accurately compute the discrete, stochastic, current in the channel [151, 152], while the details of the behaviour of individual ions are less important away from the channel where copy numbers may be very large. Thus, we can improve efficiency of our simulations if we allow ions to pass between regions with an explicitly modelled heat bath and a region where their trajectories are described by coarser stochastic models [135].

A similar multi-resolution approach can also be designed for our illustrative dimer model. It is schematically shown in Figure 2.1(c), where we identify the region with explicitly simulated heat bath as $\{x_1 > b\} = (b, \infty) \times \mathbb{R}^2$, where b is the fixed position of the boundary. We are again interested in the behaviour of the dimer in the MD model which would be considered in the full space, $\mathbb{R}^3$. However, we now want to replace solvent particles which are in $\{x_1 < b\} = (-\infty, b) \times \mathbb{R}^2$ by a coarser Langevin description (2.1)–(2.4). To do that, we have to carefully consider how we handle the transfer of monomers between $\{x_1 > b\}$ and $\{x_1 < b\}$. In Figure 2.1(c), we present a two-dimensional illustration of a monomer when it intersects the interface, $\{x_1 = b\}$. Such a monomer is subject to the collisions with heat bath particles on the part of its surface which lies in $\{x_1 > b\}$. This has to be compensated by using a suitable random force from $\{x_1 < b\}$, so that the overall model is equivalent to (2.1)–(2.4) in the Langevin limit. Such correction terms can be derived analytically for the case of a spherical monomer in our short-range interaction heat bath and are presented in References [64, 127]. They can be used to couple

the MD model with its corresponding Langevin description, which can be further coupled with Brownian dynamics, simulated using a much larger time step [127].

Mathematical analysis of multi-resolution methodologies can make use of the analysis of the model behaviour close to the boundaries of the computational domain. For example, derivations of reactive (Robin) boundary conditions of macroscopic models from their corresponding microscopic descriptions [153–155] can be generalised to the analysis of behaviour of molecules close to hybrid interfaces in multi-resolution schemes [75, 86, 87]. Analysis of open boundaries of MD schemes (i.e. boundaries which can transfer mass, momentum and energy) can lead to further understanding of multi-resolution schemes such as AdResS and hybrid continuum-particle dynamics [156], which enable efficient simulation of biomolecules at realistic physiological conditions [157].

Equations for coupled detailed/coarse-grained models can be systematically derived using Zwanzig's projection method, which has been used to address co-existence of atoms and beads (larger coarse-grained units) in the same dynamic simulations [158, 159]. The equations of motion take the form of dissipative particle dynamics, which have been coupled with atomistic water simulations to design multi-resolution schemes in the literature [83]. Other multi-resolution methods couple atomistic water with specially designed coarse-grained water models [160] or with a continuum approach [91]. Coupling discrete and continuum approaches can also be done for different molecular species present in the system and our choice of a modelling approach for each species can be based on its relative abundance [90, 161, 162].

An important point which have been left out from our discussion is the discretisation of time. Although our illustrative simulations use the same time step for both the MD model and the Langevin description, this is not the most efficient or desirable strategy, because the MD model requires much smaller time step than the corresponding Langevin equation. There is potential to design more efficient schemes by updating the coarser description only at certain multiples of the time step which is used in the most detailed model [95]. We will look at this idea

further in Chapter 3, where two coarse models of differing spatial resolution are coupled and evaluated simultaneously. There is a significant difference in stability in these models and the coupling equations when they are numerically integrated, and thus scope to evaluate the coarser model less often. This is also the case when a modeller further coarse-grains the Langevin description into a Brownian dynamics model which uses even large timesteps [127].

3

Multi-Resolution Model of an Actin Filament

In this chapter we introduce a multi-resolution model of F-actin, based on my work submitted for publication as R. S. Gunaratne et al. "On Stretching, Bending, Shearing and Twisting of Actin Filaments II: Multi-Resolution Modelling" [2], in which all figures featured in this chapter appear, with reference to accompanying paper C. Floyd et al. "On Stretching, Bending, Shearing and Twisting of Actin Filaments I: Variational Models" [3]. In Section 3.1 we reintroduce the models of F-actin discussed in Section 1.4: (i) a rod-based model of an actin filament, utilising a Cosserat description developed by Gazzola et al. [98]. (ii) a monomer-based model, with actin described using bound ellipsoids, developed by De La Cruz et al. [97]. These can both account for the torsional, bending, shear and compression properties seen in F-actin filaments, though the Ellipsoid model is able to describe filament behaviour with a higher level of detail, but less efficiently. For the ellipsoid model, we use parameters derived in papers subsequent to De La Cruz et al. [49, 119], along with giving explicit details of the methods we used to initialise and simulate the model in Chapter 4. We demonstrate how these models can be used in a multi-resolution framework, where the monomer-based description is substituted in parts of the filament by the rod-based description to

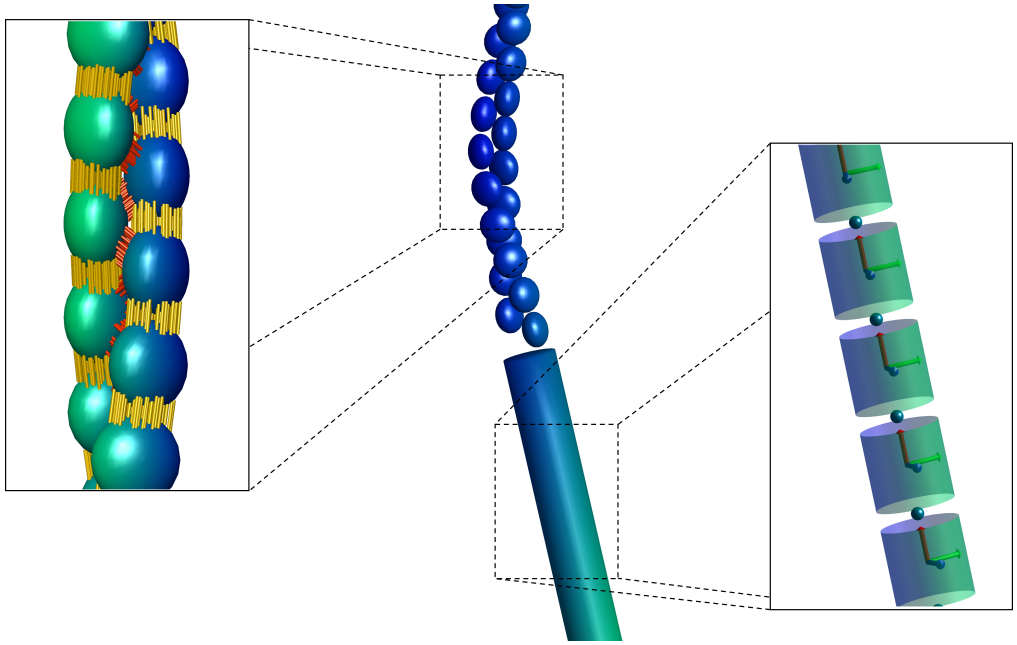


Figure 3.1: *A schematic of the Ellipsoid model (left panel) and the Cosserat model (right panel) which are coupled at the multi-resolution coupling to form a single, multi-resolution model of F-actin (middle panel).*

improve overall simulation efficiency, with exact details of simulations timings seen in Table C.6 in the appendices. Our multi-resolution coupling strategy is explained in Section 3.2, which is followed by details of the parametrisation of the models and extensions to the Cosserat model presented by Gazzola et al. [98] which are used in the multi-resolution framework in Section 3.3. We conclude in Section 3.4 by discussing further extensions to the Cosserat model to improve the performance of the model when applied to studies of actin, and other potential applications.

3.1 Ellipsoid and Cosserat models of actin filaments

We introduced two models of actin in Section 1.4 that can be used to describe F-actin filaments behaviour; a high resolution, monomer-based model and a low-resolution, rod-based model. These will be combined, as seen in Figure 3.1 to form a multi-resolution model, with both models updated and discretised in a consistent manner. The linear and angular acceleration for the Cosserat model is seen in

equations (1.9)–(1.10), while that of the Ellipsoid model is given in equations (1.13)–(1.14). These models are largely similar to those outlined in Gazzola et al. [98] and De La Cruz et al. [97] and subsequent extensions [49, 119], though we make some small amendments to improve performance.

As previously mentioned, we include a rigid body term in the Ellipsoid model angular velocity acceleration term in equation (1.14), ensuring it is consistent with equation (1.10), though this term only has a bearing on dynamics and will not affect system equilibria. The orientation of the i^{th} monomer is characterised by the matrix $\boldsymbol{Q}_i \in \mathbb{R}^{3\times3}$, for $i = 1, 2, \ldots, N$, whose columns are three orthonormal vectors $\boldsymbol{d}_{1,i}$, $\boldsymbol{d}_{2,i}$, $\boldsymbol{d}_{3,i} \in \mathbb{R}^3$, where $\boldsymbol{d}_{3,i}$ is the direction of the long filament axis. Rather than expressing the harmonic bond endpoints as individual coordinates, we can instead express their location based on their initial positions, along with the position and orientation of the monomer at time t

$$\boldsymbol{r}_i^k(t) = \boldsymbol{Q}_i(t)\, \boldsymbol{Q}_i^T(0) \left(\boldsymbol{r}_i^k(0) - \bar{\boldsymbol{R}}_i(0) \right) + \bar{\boldsymbol{R}}_i(t)\,.$$

The time evolution of $\boldsymbol{d}_{j,i}$ is for $j = 1, 2, 3$ given by

$$\frac{\mathrm{d}\boldsymbol{d}_{j,i}}{\mathrm{d}t} = (\boldsymbol{Q}_i\, \boldsymbol{\omega}_{i,\mathcal{L}}) \times \boldsymbol{d}_{j,i}\,,$$

where each respective element of $\boldsymbol{\omega}_{i,\mathcal{L}} \in \mathbb{R}^3$ is the angular velocity around the $\boldsymbol{d}_{j,i}$ vectors that make up the orientation basis $\boldsymbol{Q}_i$. This is useful when defining our multi-resolution coupling in Section 3.2. We make two amendments to the Cosserat model, which will be discussed in detail in Section 3.3 where we try to improve on the performance of the standard model to better capture the properties of F-actin. The first is to allow twisting rigidity $B_3(\kappa_{\mathcal{L},3})$ to be curvature dependent, which is studied in Section 3.3.3, alongside twist-stretch coupled behaviour, detailed in Section 3.3.4. To integrate both of these models (in addition to the coupling equations seen in Section 3.2) we apply the 4^{th} order symplectic method presented by Omelyan, Mryglod and Folk [163], which is described in detail in Section 4.1.

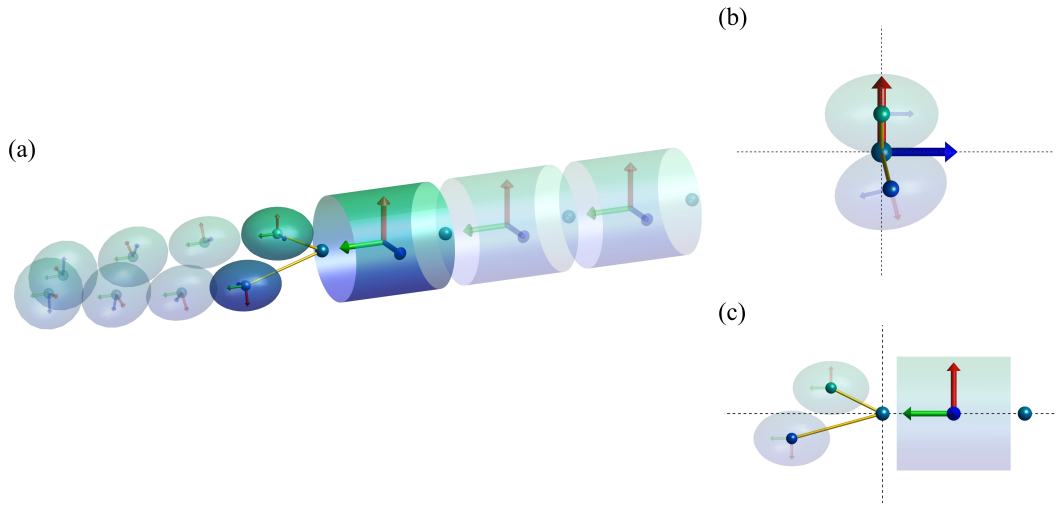


Figure 3.2: (a) *Illustration of multi-resolution coupling region. The final Cosserat model segment is bound to the final two monomers in the Ellipsoid model. This is achieved by attaching the final segment to the ellipsoid centres with bonds (yellow) that have the same material properties dictated by the model parameters of the Cosserat model. While only one coupling region is illustrated here, a single filament may have multiple couplings across its length, switching between the two models when required.*
(b) *View along the centre line.* (c) *Side view.*

3.2 Multi-resolution coupling region

To couple the Ellipsoid model of actin filament to the Cosserat model, we must find a way to mediate transfer of forces between models at boundary regions. This involves projecting forces resulting from each of the harmonic bond anchor sites on the monomers in the coupling region to forces acting solely on the angular and translational components of one (or more) rod segments of the Cosserat model. Both the Ellipsoid and Cosserat models have orientation matrices for monomers and rods, respectively, with their angular velocities defined in their local frame. To capture twisting behaviour, we must define forces that are dependent on orientation in addition to the endpoint and ellipsoid centre positions. Below we present a method to couple the models in a manner consistent with the equations of motion given in Section 1.4. While the coupling may seem complex, it requires the creation of only one new set of coordinates, describing the orientation of the coupling region. Because of its accuracy, we utilise it in Chapter 4 when presenting examples cases for our multi-resolution model, with alternate methods of coupling the Ellipsoid

and Cosserat model given in Appendix B.

The rod-based part of the filament is described by the centre-line $\boldsymbol{r}(s,t)$, where $s \in [0,1]$, and we identify $s = 0$ with the endpoint, which is attached to the monomer-based Ellipsoid model, i.e. $\boldsymbol{r}(0,t)$ is the endpoint of the filament for the Cosserat model and $\boldsymbol{Q}(0,t)$ is the corresponding orientation basis. However, in our discretisation $\boldsymbol{Q}(0,t)$ is not defined at the endpoint, so as a compromise we must use $\boldsymbol{Q}(|\boldsymbol{l}_1|/2,t)$, corresponding to the orientation of the final discrete segment of length $|\boldsymbol{l}_1|$. Labelling the monomer closest to $\boldsymbol{r}(0,t)$ as the monomer 1, we intend to attach the rod-based part of the filament to the monomer with its centre denoted by $\bar{\boldsymbol{R}}_1$ and its orientation basis by $\boldsymbol{Q}_1$. To do this, we create a new bond between $\boldsymbol{r}(0,t)$ and $\bar{\boldsymbol{R}}_1$, with quantities labelled with subscripts MR, Cos, and Ell corresponding to the multi-resolution coupling region, Cosserat model, and Ellipsoid model respectively [2]. We define an orientation basis for the coupling $\boldsymbol{Q}_{\mathrm{MR}}$, along with

$$\boldsymbol{l}_{\mathrm{MR}} = \bar{\boldsymbol{R}}_1 - \boldsymbol{r}(0,t)\,, \qquad e_{\mathrm{MR}} = \frac{|\boldsymbol{l}_{\mathrm{MR}}|}{|\hat{\boldsymbol{l}}_{\mathrm{MR}}|}\,, \qquad \boldsymbol{t}_{\mathrm{MR}} = \frac{\boldsymbol{l}_{\mathrm{MR}}}{|\boldsymbol{l}_{\mathrm{MR}}|}\,,$$

$$\boldsymbol{\sigma}_{\mathrm{MR}} = \boldsymbol{Q}_{\mathrm{MR}}^T \left(e_{\mathrm{MR}}\boldsymbol{t}_{\mathrm{MR}} - \boldsymbol{Q}_{\mathrm{MR}}\boldsymbol{d}_{\mathrm{MR}}\right)\,,$$

where the coupling direction vector, extension, tangent, and strain respectively are formulated in a manner consistent with the Cosserat model. The key difference occurs in the definition of the strain, where, rather than using the basis vector which is parallel to the backbone $\boldsymbol{d}_3$, we instead use the rest vector direction between $\boldsymbol{r}(0,t)$ and $\bar{\boldsymbol{R}}_1$

$$\boldsymbol{d}_{\mathrm{MR}} = \frac{\hat{\boldsymbol{l}}_{\mathrm{MR}}}{|\hat{\boldsymbol{l}}_{\mathrm{MR}}|}\,,$$

and rotate this vector to the current orientation of the coupling bond. This allows us to initialise the model such that the orientation basis is aligned with the filament direction, even though the coupling bond is not, so in the case of a straight, untwisted filament

$$\boldsymbol{Q}(|\boldsymbol{l}_1|/2,t) = \boldsymbol{Q}_1 = \boldsymbol{Q}_{\mathrm{MR}}.$$

The advantage of proceeding in this manner is that it allows us to define a strain matrix $\boldsymbol{S}$ in a manner consistent with the Cosserat model, with the correct strains both axially and radially. First, we define the linear force experienced by the final segment endpoint and ellipsoid centre as a result of the coupling's shearing and stretching properties

$$\boldsymbol{F}_{\text{Cos}} = \frac{\boldsymbol{Q}_{\text{MR}}\hat{\boldsymbol{S}}\boldsymbol{\sigma}_{\text{MR}}}{e_{\text{MR}}} \quad , \quad \boldsymbol{F}_{\text{Ell}} = -\boldsymbol{F}_{\text{Cos}}. \tag{3.1}$$

While this force keeps the segment and ellipsoids in the coupling in the correct spatial positions, we introduce an angular constraint to ensure the ellipsoid monomers maintain the correct orientation relative to the final Cosserat segment. First we define the rest lengths of the Voronoi regions and curvatures of the Cosserat segment relative to the coupling, and the coupling relatives to the ellipsoid

$$\begin{aligned}
\hat{D}_{\text{Ell}} &= \frac{|\hat{\boldsymbol{d}}_3 \cdot \boldsymbol{Q}_{\text{MR}}^T \hat{\boldsymbol{l}}_{\text{MR}}|}{2}, & \hat{D}_{\text{Cos}} &= \frac{|\hat{\boldsymbol{l}}_1|}{2} + \frac{|\hat{\boldsymbol{d}}_3 \cdot \boldsymbol{Q}_{\text{MR}}^T \hat{\boldsymbol{l}}_{\text{MR}}|}{2}, \\
\mathcal{E}_{\text{Ell}} &= \frac{D_{\text{Ell}}}{\hat{D}_{\text{Ell}}}, & \mathcal{E}_{\text{Cos}} &= \frac{D_{\text{Cos}}}{\hat{D}_{\text{Cos}}}, \\
\hat{\boldsymbol{\kappa}}_{\text{Ell}} &= \frac{\log\left(\boldsymbol{Q}_1^T \boldsymbol{Q}_{\text{MR}}\right)}{\hat{D}_{\text{Ell}}}, & \hat{\boldsymbol{\kappa}}_{\text{Cos}} &= \frac{\log\left(\boldsymbol{Q}_{\text{MR}}^T \boldsymbol{Q}(|\boldsymbol{l}_1|/2, t)\right)}{\hat{D}_{\text{Cos}}},
\end{aligned}$$

where $|\hat{\boldsymbol{l}}_i|$ is the rest length of a Cosserat segment. By defining length along the filament centre-line for the Voronoi regions, rather than through the coupling bond, we define curvature consistently with the Cosserat model, which has a significant effect when the coupling region is very short and $|\hat{\boldsymbol{d}}_3 \cdot \boldsymbol{Q}_{\text{MR}}^T \hat{\boldsymbol{l}}_{\text{MR}}| \ll |\hat{\boldsymbol{l}}_{\text{MR}}|$. Having introduced these curvatures, we can define bending and twisting torque forces

$$\begin{aligned}
\boldsymbol{\tau}_{\text{Ell}} &= -\frac{\hat{\boldsymbol{B}}\hat{\boldsymbol{\kappa}}_{\text{Ell}}}{\mathcal{E}_{\text{Ell}}^3} + \frac{\hat{\boldsymbol{\kappa}}_{\text{Ell}} \times \hat{\boldsymbol{B}}\hat{\boldsymbol{\kappa}}_{\text{Ell}}}{2\mathcal{E}_{\text{Ell}}^3}\hat{D}_{\text{Ell}}, \\
\boldsymbol{\tau}_{\text{Cos}} &= \frac{\hat{\boldsymbol{B}}\hat{\boldsymbol{\kappa}}_{\text{Cos}}}{\mathcal{E}_{\text{Cos}}^3} + \frac{\hat{\boldsymbol{\kappa}}_{\text{Cos}} \times \hat{\boldsymbol{B}}\hat{\boldsymbol{\kappa}}_{\text{Cos}}}{2\mathcal{E}_{\text{Cos}}^3}\hat{D}_{\text{Cos}},
\end{aligned} \tag{3.2}$$

where the matrix logarithm is calculated using Rodrigues' formula [164] (see Section 4.1.2), in a manner identical to that used in the Cosserat model. All that is left to consider is the evolution of the orientation basis of the coupling, which we will consider to have angular velocity $\boldsymbol{\omega}_{\text{MR}}$ defined with respect to $\boldsymbol{Q}_{\text{MR}}$.

The resulting force takes with a similar form to that for the angular velocity for the Cosserat model [2]

$$\boldsymbol{\tau}_{\mathrm{MR}} = +\frac{\hat{\boldsymbol{B}}\hat{\boldsymbol{\kappa}}_{\mathrm{Ell}}}{\mathcal{E}^3_{\mathrm{Ell}}} - \frac{\hat{\boldsymbol{B}}\hat{\boldsymbol{\kappa}}_{\mathrm{Cos}}}{\mathcal{E}^3_{\mathrm{Cos}}} + \frac{\hat{\boldsymbol{\kappa}}_{\mathrm{Ell}} \times \hat{\boldsymbol{B}}\hat{\boldsymbol{\kappa}}_{\mathrm{Ell}}}{2\mathcal{E}^3_{\mathrm{Ell}}} D_{\mathrm{Ell}} + \frac{\hat{\boldsymbol{\kappa}}_{\mathrm{Cos}} \times \hat{\boldsymbol{B}}\hat{\boldsymbol{\kappa}}_{\mathrm{Cos}}}{2\mathcal{E}^3_{\mathrm{Cos}}} D_{\mathrm{Cos}} \tag{3.3}$$
$$+ \left(\boldsymbol{Q}^T_{\mathrm{MR}}\boldsymbol{t}_{\mathrm{MR}} \times \hat{\boldsymbol{S}}\boldsymbol{\sigma}_{\mathrm{MR}}\right)|\hat{\boldsymbol{l}}_{\mathrm{MR}}| + \frac{\hat{\boldsymbol{I}}\boldsymbol{\omega}_{\mathrm{MR}}}{e^2_{\mathrm{MR}}}\frac{\partial e_{\mathrm{MR}}}{\partial t} + \left(\frac{\hat{\boldsymbol{I}}\boldsymbol{\omega}_{\mathrm{MR}}}{e_{\mathrm{MR}}}\right) \times \boldsymbol{\omega}_{\mathrm{MR}}\,,$$

By proceeding in this manner we are able to fully capture the shear, stretch, bend, and twist of the coupling region in a way that is intuitively consistent with both of the models we are coupling, while introducing the bare minimum in new parameters and coordinates [2].

3.3 Matching the Ellipsoid and Cosserat models

Having established the equations of motion for our multi-resolution model of F-actin, we must now determine parameters such that we have consistent macroscopic behaviour between areas of differing resolution. Firstly, we lay out the methodology used for the individual Ellipsoid and Cosserat models, with parametrisation informed by both simulation and experimental data. We then make parametrisation compromises, along with extensions to the Cosserat model where there is disagreement with the Ellipsoid model. In Section 3.3.1 we consider the parametrisation of the Ellipsoid model [97] by inference from MD simulation data [49, 119] and solvent accessible surface area calculations. We also explicitly describe the the full details of system initialisation used when simulating the model for example cases in Chapter 4, including methods for uniform sampling on the surface of an ellipsoid and drawing of bonds between pairs of ellipsoid monomers. In Section 3.3.2 we turn our attention to parametrisation of the Cosserat model [98] based on the material properties of uniform filaments of circular cross-section. We include parameter justifications used in Floyd et al. [3] which are based on physical properties of actin from experimental data. However, these are not used in our final multi-resolution model, and in Section 3.3.3 we discuss how the parameters of the Cosserat model are matched to Ellipsoid model simulation data to ensure correct macroscopic filament

behaviour across the length of a multi-resolution filament, with particular focus on bending and twisting rigidities. We draw comparisons between the inferred parameters and those calculated based on Young's and shear moduli. The Cosserat model is amended to allow for variable twisting rigidity, then further extended in Section 3.3.4 to account for the twist-stretch behaviour of the F-actin. This extended Cosserat model is better able to mimic the macroscopic behaviour of the Ellipsoid model than the version presented in Gazzola et al. [98], which will be demonstrated by simulations in Chapter 4.

3.3.1 Ellipsoid model parametrisation

The Ellipsoid model was first described in [97], and later extended to cofilin decorated actin filaments [49, 119]. The parameters for the Ellipsoid model are inferred from all atom MD simulations and crystal structure data [165]. The initial positions of the centres of the ellipsoid monomers are set to the centres of mass of the monomers in the F-actin chain, with their dimensions approximately matching those of actin. There are two different sets of harmonic interactions between monomers; lateral (between monomers in adjacent helical chains) and longitudinal (between monomers in the same helical chain). In the original Ellipsoid model papers [49, 119] points on the surface where these bonds are attached to are uniformly sampled across surface areas determined by solvent accessible surface area calculations [166]. While this gives sensibly sized interface areas, it is not a perfect option; the true shape of G-actin monomers is irregular and the interaction surfaces in reality will be very close together when compared with the modelled distance between ellipsoid surfaces. In this work we chose to match the lateral and longitudinal surface areas, resulting in increased torsional rigidity relative to bending rigidity. Whatever the choice of this tunable parameter, it is possible to match the dynamics of the Ellipsoid and Cosserat models, as will be discussed in Section 3.3.3. The bonds are parallel to the line between the centres of the two ellipsoids they connect in the initial configuration of the model. The lateral bonds all have the same rigidity, as do the longitudinal bonds. This means that even though we could have several thousand

harmonic bonds in any given model, only two effective parameters need to be fitted, differing from previous approaches seen in the literature [60, 106] where a multitude site and bond lengths and rigidities are described, with a resulting loss of clarity and danger of over-fitting.

In this work we set the harmonic bonds and their anchor points using the following method [2]. If we have two adjacent ellipsoid monomers, we uniformly sample a point on the surface of the first one. This process starts by considering an ellipsoid centred at the origin, and sampling the elements of a vector $\boldsymbol{n} = [n_1, n_2, n_3]$ from a unit normal distribution, *i.e.* $n_k \sim N(0,1)\,, \quad k = 1, 2, 3$. With the ellipsoid principle radial lengths a, b, and c oriented in the x, y, and z directions, respectively, we define a normalised vector $\boldsymbol{q} = \boldsymbol{n}/|\boldsymbol{n}|$ and we accept point $\boldsymbol{r} = [aq_1, bq_2, cq_3]$ with probability

$$c\sqrt{\left(\frac{q_1}{a}\right)^2 + \left(\frac{q_2}{b}\right)^2 + \left(\frac{q_3}{c}\right)^2}.$$

This acceptance-rejection step is necessary as otherwise we would have non-uniform sampling, most noticeable at the poles, and this acts as a scaling which depends on the relative lengths of a, b, and c. This general point can then be translated and rotated to the correct position on the Ellipsoid surface. For example, if this is the anchor point for the k$^{\text{th}}$ bond on the i^{th} monomer then

$$\boldsymbol{r}_i^k = \bar{\boldsymbol{R}}_i + \boldsymbol{Q}_i^T \boldsymbol{r},$$

where $\bar{\boldsymbol{R}}_i \in \mathbb{R}^3$ and $\boldsymbol{Q}_i \in \mathbb{R}^{3\times3}$ are the centre of mass and orientation. Having set a point on the i^{th} monomer, we now connect this to the surface of the adjacent j^{th} monomer. It is possible to write a vector equation of motion for points $\boldsymbol{r}$ on the ellipsoid surface

$$(\boldsymbol{r} - \bar{\boldsymbol{R}}_j)^T \boldsymbol{Q}_j^T \boldsymbol{D} \boldsymbol{Q}_j (\boldsymbol{r} - \bar{\boldsymbol{R}}_j) = 1,$$

where $\boldsymbol{D} = \text{diag}(1/a^2, 1/b^2, 1/c^2) \in \mathbb{R}^{3\times3}$. By definition, the line connecting the monomers is parallel to the vector connecting their centres and passes through the first anchor point, so its equation can be written as

$$\boldsymbol{r} = \boldsymbol{r}_i^k + \lambda \boldsymbol{l}\,, \quad \boldsymbol{l} = \frac{\bar{\boldsymbol{R}}_i - \bar{\boldsymbol{R}}_j}{|\bar{\boldsymbol{R}}_i - \bar{\boldsymbol{R}}_j|}.$$

Substituting this into the equation for the ellipsoid gives a quadratic equation

$$\alpha\lambda^2 + \beta\lambda + \gamma = 0,$$
$$\alpha = \boldsymbol{l}^T\boldsymbol{Q}_j^T\boldsymbol{D}\boldsymbol{Q}_j\boldsymbol{l}\,, \quad \beta = 2\boldsymbol{l}^T\boldsymbol{Q}_j^T\boldsymbol{D}\boldsymbol{Q}_j(\boldsymbol{r}_i^k - \bar{\boldsymbol{R}}_j)\,,$$
$$\gamma = (\boldsymbol{r}_i^k - \bar{\boldsymbol{R}}_j)^T\boldsymbol{Q}_j^T\boldsymbol{D}\boldsymbol{Q}_j(\boldsymbol{r}_i^k - \bar{\boldsymbol{R}}_j) - 1.$$

which can be solved in λ giving one or two real solutions if the line touches or intersects with the ellipsoid and no real solutions if it doesn't intersect, with the point on the surface of the j^{th} set as

$$\boldsymbol{r}_j^k = \boldsymbol{r}_i^k + \lambda\boldsymbol{l}$$

with the root chosen to minimise the bond length $|\boldsymbol{r}_j - \boldsymbol{r}_i|$. We can then impose an additional acceptance-rejection constraint on the maximum tolerated bond length to enforce sampling over a specified interface surface area. We verified the tolerance length was correct for each interface by sampling 10^5 bonds and carrying out a Delaunay triangulation [167] on the bond anchor points to create a 2-dimensional mesh representation of the surface (roughly an ellipsoidal dome) and numerically calculated the area of each triangle in this discrete surface.

To determine the bond rigidity parameters a fluctuation matching technique is employed [59, 119]. First, MD simulations are carried out for a test length of F-actin (usually 75 nm, or 26 actin monomers, a full period for the helix) and allowed to settle for a set length of time. After settling, the simulations are run for a further period of time to generate elastic network model data, with the movement of the centres of masses of the actin monomers being recorded. Using this data, the spring constants of the harmonic bonds in the Ellipsoid model can then be iteratively adjusted until eventually the the fluctuations of the coarse grain model match that of the data taken from the atomistic model [59, 60]. For this work, we do not explicitly carry out these MD simulations or iterative matching, instead making use of the values from original papers for the Ellipsoid model [49, 119], which are given in Table 3.1, along with our chosen values for the maximum interface bond length. While this "bottom up" approach can be expected to give an accurate picture of the relative inter-monomer forces, the macroscopic behaviour does not

Parameter	Value	Reference
Actin dimensions	$5.4 \times 5.4 \times 3.8$ nm	[165]
Filament period	71.2 nm	[168]
Monomers per filament period	26	[168]
Rise per monomer	5.52 nm	[168]
Effective interaction radius	1.8 nm	[168]
Lateral interface rigidity	392.007 $k_BT\,\mathrm{nm}^{-2}$	[165]
Longitudinal interface rigidity	582.361 $k_BT\ \mathrm{nm}^{-2}$	[165]
Lateral interface maximum bond length	1.6 nm	[2]
Longitudinal interface maximum bond length	2.2 nm	[2]

Table 3.1: *Parameters used to calibrate the Ellipsoid model used in Schramm et al.* [119]*, along with maximum interface bond lengths.*

exactly match up with experimental measures [49, 119], with lower filament rigidity observed under bending and twisting. We later address this discrepancy when coupling the parameters of the models in Section 3.3.3

3.3.2 Cosserat model parametrisation

It is possible to set the model parameters for the Cosserat model (describing shear, stretch, bending, and twisting rigidities) using experimentally measured values, approximating actin as having a cylindrical cross-section for comparison with other variational models. This is how we proceeded in Floyd et al. [3], which accompanies the work carried out in this chapter. We will outline the methodology used as a means of comparison for our choices in Section 3.3.3.

The diameter of an actin filament is in the range $5 - 7$ nm [169], giving it a cross-sectional area A (assuming roughly circular) in the range 40-75 nm^2. The second moment of inertia tensor is given by

$$\hat{\boldsymbol{I}} = \frac{mA^2}{4\pi} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

for circular cross-sectioned filaments. For an isotropic material, the matrix $\hat{\boldsymbol{B}}$ is diagonal and made up of the bending (B_1, B_2) and the twisting (B_3) rigidities, given by

$$B_{1,2} = EI_{1,2} \qquad B_3 = GI_3$$

where E and G are the Young's and shear moduli respectively. The Young's modulus has been estimated as $E \approx 2$ GPa [170]. To estimate the shear modulus of actin, we considered the formula [114]

$$E = 2G(1 + \nu)$$

where ν is Poisson's ratio. For actin, ν has been estimated at around 0.4 [170, 171], giving us $G \approx 0.7$ GPa. The matrix $\hat{\boldsymbol{S}}$ is also diagonal, and comprises of shearing (S_1, S_2) and stretching (S_3) rigidities of an actin filament, given by

$$S_{1,2} = \alpha_c GA \qquad S_3 = EA$$

where $\alpha_c = \frac{4}{3}$ for circular cross-sections [98]. All of the above parameter justifications and selections are summarised in Table 3.2. The Cosserat model can now be used

Parameter		Value	Reference
Filament cross-sectional area	A	$40 - 75$ nm^2	[169]
Young's modulus	E	2000 pN nm^{-2}	[170]
Poisson's ratio	ν	~ 0.4	[170, 171]
Shear modulus	G	~ 700 pN nm^{-2}	[114]
Second moment of inertia	$I_{1,2}$	$A^2/4\pi$	[98]
	I_3	$A^2/2\pi$	
Shearing rigidity	$S_{1,2}$	$\alpha_c GA$	[98]
	S_3	EA	
Bending rigidity	$B_{1,2}$	$EI_{1,2}$	[98]
	B_3	GI_3	

Table 3.2: *Parameters of the Cosserat model for F-actin used in Floyd et al.* [3].

to model F-actin filaments, with this continuum description coarsely capturing its macroscopic properties. However, despite using reasonable estimates from the literature, we noted in that this representation of actin had much higher twisting and bending rigidity than that observed in the Ellipsoid model. We will address this when coupling the parameters of the two models.

3.3.3 Matching Ellipsoid and Cosserat parameters

While it seems it should be a relatively simple task to match the properties of two models of the same material, due to the different assumptions and limitations

in the methods used to collect the data for parametrisation this is non-trivial. We proceed by taking "physical" measurements from the Ellipsoid model, and use equations governing the physical properties of filaments to match the model parameters of the Cosserat model, starting with the bending rigidity. The following relation for the deflection of a filament ΔL_{D} under a load force F and clamped at the opposite end holds for elastic materials

$$B_1 = B_2 = \frac{FL^3}{3\Delta L_{\mathrm{D}}}$$

where L is the length of the filament. This is derived from the solution to the cantilever equation for a uniform rod [172], with deflection along length of filament given by

$$\Delta L_{\mathrm{D}}(s) = \frac{Fs^2(3L - s)}{6B_1},$$

where s is the position along the filament centreline where the force is applied relative to the anchor point, so in our case $s = L$. These rigidities B_1, B_2 are the first and second diagonal element of the bending rigidity matrix $\boldsymbol{B}$ for the Cosserat model. Applying the relations defined in Section 3.3.2 we know that $B_1 = EI_1$ where E is the Young's modulus, and $I_1 = A^2/4\pi$. We are principally concerned with investigating the relative behaviour of the Ellipsoid and Cosserat models, so ensuring correct scaling between them is sufficient. However, in the interest of having rough correspondence between the results in Chapter 4 and physical reality, the rigidities used in the Ellipsoid model are scaled up empirically. In Schramm et al. [119], it was stated that the simulated persistence length of the Ellipsoid model is underestimated by around 40%, with a value of 7.0 μm measured, rather than an expected 9.8 μm, so our scale factor will be ~ 1.4. The decision to scale parameters is backed by experimental data for F-actin [173], with measurements giving $E = 330$ pN nm^{-2}, using the assumption that the actin cross-section is set to $A = 38.5\,\mathrm{nm}^2$, and resulting in bending inertia $I_1 = 120\,\mathrm{nm}^4$. This scaling is further justified by other studies of F-actin [174, 175], where the persistence length is stated to be in the range 8–20 μm. The overall bending rigidity used in the multi-resolution

model is $B_1 = 3.9 \times 10^4$ pN nm^2, as seen in Table 3.4. With these parameter choices, we maintain the relative microscopic monomer level behaviour of the filament, while still being able to match the observed macroscopic physical properties taken from experimental data, ensuring simulations are more physically meaningful.

Having accounted for bending behaviour, we now turn our focus to the twisting rigidities, which can be matched using the relation

$$B_3 = \frac{\tau L}{\Delta\theta}$$

where L is the filament length and $\Delta\theta$ is the twist angle under a torsional force τ. This was done by placing the filament of length $\sim 75\,\text{nm}$ (i.e the actin pitch length) under a fixed twist in one of the end segments and measured the torque force after the system relaxed. In Table 3.3 the value for the twisting rigidity is given for various levels of filament twisting, with the line above 200° used to indicate the point up to which values can be considered reliable (for large twists the Ellipsoid model is either completely untwisted or beginning to coil out of plane and will not be physically realistic). With the new scaling used, this resulted in maximum values of $B_3 \approx 9.6 \times 10^3$ pN nm^2 for over-twist and $B_3 \approx 1.6 \times 10^4$ pN nm^2 for under-twist. A simple compromise would be to take the average of these values, giving a value of $B_3 \approx 1.3 \times 10^4$ pN nm^2 (as used in Sections 4.3.1–4.3.2). Alternatively, the twisting rigidity can be allowed to vary in a manner consistent with the Ellipsoid model. We use a twist rigidity $B_3(\kappa_{\mathcal{L},3})$ which is curvature dependent to account for the variable twist rigidity of actin. We formulate an arbitrary function

$$\begin{aligned} B_3(\kappa_{\mathcal{L},3}) = &\left(\frac{B_{\max} + B_{\min}}{2}\right) + \left(\frac{B_{\max} - B_{\min}}{2}\right) \tanh\left(\frac{\kappa_{\mathcal{L},3} + \lambda_1}{\lambda_2}\right) \\ &+ \lambda_3 \left(\frac{B_{\max} + B_{\min}}{2}\right) \left(\tanh\left(\frac{\kappa_{\mathcal{L},3} - \lambda_4}{\lambda_5}\right) + \tanh\left(\frac{-\kappa_{\mathcal{L},3} - \lambda_6}{\lambda_7}\right)\right) \end{aligned} \tag{3.4}$$

and vary the values of λ_i to fit the data in Table 3.3. The first line of the equation tends to $B_{\min}$ and $B_{\max}$ when $\kappa_{\mathcal{L},3} \to -\infty$ and $\kappa_{\mathcal{L},3} \to \infty$ respectively, while the second line of the equation tends to 0 when $\kappa_{\mathcal{L},3} \to \pm\infty$. These properties ensure a

Under-twist			Over-twist		
ϕ	$\kappa_{\mathcal{L},3}$	B_3	ϕ	$\kappa_{\mathcal{L},3}$	B_3
240°	0.0555	9.6×10^3 pN	240°	−0.0555	1.2×10^4 pN
220°	0.0508	9.6×10^3 pN	220°	−0.0508	1.4×10^4 pN
200°	0.0462	9.4×10^3 pN	200°	−0.0462	1.6×10^4 pN
180°	0.0416	9.1×10^3 pN	180°	−0.0416	1.6×10^4 pN
160°	0.0370	8.6×10^3 pN	160°	−0.0370	1.5×10^4 pN
140°	0.0324	8.1×10^3 pN	140°	−0.0324	1.4×10^4 pN
120°	0.0277	7.4×10^3 pN	120°	−0.0277	1.4×10^4 pN
100°	0.0231	6.8×10^3 pN	100°	−0.0231	1.2×10^4 pN
80°	0.0185	5.8×10^3 pN	80°	−0.0185	1.0×10^4 pN
60°	0.0139	4.8×10^3 pN	60°	−0.0139	7.8×10^3 pN
40°	0.0092	3.5×10^3 pN	40°	−0.0092	5.0×10^3 pN
20°	0.0047	1.6×10^3 pN	20°	−0.0047	2.1×10^3 pN
10°	0.0023	6.6×10^2 pN	10°	−0.0023	8.2×10^2 pN

Table 3.3: *Twisting rigidity of ellipsoidal actin filament at various curvatures calculated by inference from simulation data [2].*

good fit, with computationally simple inference, giving final parameters

$$\begin{aligned} \lambda_1 &= 1.5928 \times 10^{-2} \\ \lambda_2 &= 7.2455 \times 10^{-3} \\ \lambda_3 &= 4.5982 \\ \lambda_4 &= 1.0000 \times 10^{-6} \\ \lambda_5 &= 1.8196 \times 10^{-2} \\ \lambda_6 &= 4.5935 \times 10^{-3} \\ \lambda_7 &= 1.7203 \times 10^{-2} \end{aligned} \tag{3.5}$$

along with $B_{\max} = 1.6 \times 10^4$ pN nm^2 and $B_{\min} = 9.6 \times 10^3$ pN nm^2 chosen as the rigidity limits that we reach for large curvature, with data and fitted curve seen in Figure 3.3. Interestingly, if we compare the average value of B_3 with the expected value using the assumptions for a rigid uniform filament seen in Section 3.3.2 we would expect $B_3 = GI_3 \approx 2.8 \times 10^4$ pN nm^2 which is 2.2 times larger than observed, suggesting that the double helical structure of actin allows for increased twisting relative to bending, when compared with a uniform Cosserat description.

We also set the extension/compression rigidity by extending an ellipsoidal

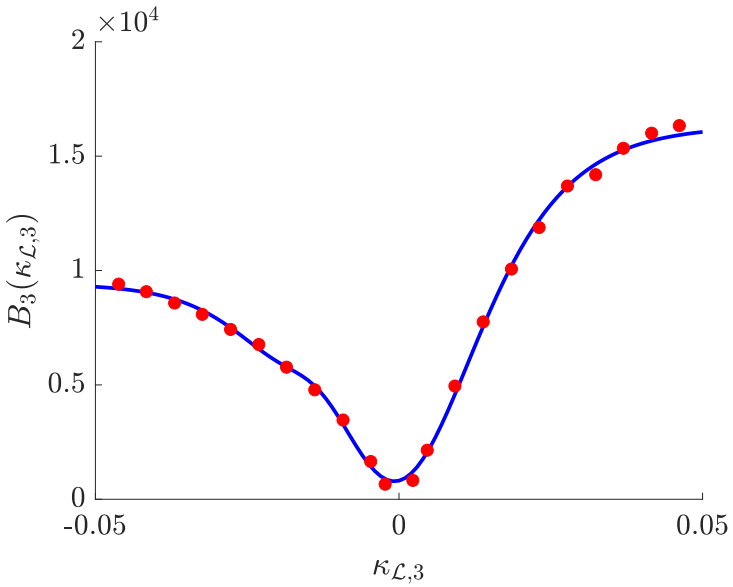


Figure 3.3: *Twisting rigidity of ellipsoidal actin filament for varying curvature. We over-twist and under-twist a* $\sim 75\,\mathrm{nm}$ *long, straight filament and measure the twisting rigidity for these points (circular markers). A continuous function for* $B_3(\kappa_{\mathcal{L},3})$ *fitted to this data.*

filament, fixing one end and applying a force on the other, and using the formula

$$S_3 = \frac{FL}{\Delta L}$$

where L is the filament length and ΔL is the length the filament extends under force F. This was done using the same set up as the torsion test above, with segments having their rotational orientation fixed. We also set a fixed extension in one of the end segments and measured the force after the system relaxed as an equivalent check. This rigidity S_3 is the third diagonal element of the bending rigidity matrix $\boldsymbol{S}$ for the Cosserat model. Interestingly, this was measured at $S_3 = 3.4 \times 10^4$ pN which, when compared to the relation $S_3 = EA \approx 1.3 \times 10^4$ pN, suggests that the observed Young's modulus or filament cross-section are 2.7 time larger than expected. This implies that the structure of actin allows for greater bending relative to compression and stretching the uniform rods described in Section 3.3.2. Setting the final component of the shear matrix S_2 and S_3 is less straightforward from direct measurement, so we assume that the factor of 2.7 stretch scaling translates directly into $S_1 = S_2 = \alpha GA \approx 1.6 \times 10^4$ pN. The final set of rigidity used in

the model for the Cosserat model are given in Table 3.4, in addition to the scaled interface rigidities for the Ellipsoid model. For the purposes of the example cases

Parameter		Value
Cosserat shearing rigidity	$S_{1,2}$	1.6×10^4 pN
Cosserat stretching rigidity	S_3	3.4×10^4 pN
Cosserat bending rigidity	$B_{1,2}$	3.9×10^4 pN nm^2
Cosserat twisting rigidity	B_3	$1.0 - 1.6 \times 10^4$ pN nm^2
Ellipsoid lateral interface rigidity	$S_\perp$	2.3×10^3 pN
Ellipsoid longitudinal interface rigidity	$S_\parallel$	3.4×10^3 pN

Table 3.4: *Parameters used for the Ellipsoid and Cosserat model for multi-resolution model, calculated by inference from simulation data, comparison to experimental data, and physical theory [2].*

tested in Section 4.3, this is enough to ensure reasonable correspondence between regions of the model of varying resolution. However, this does not line up well with estimates given in Section 3.3.2, which are set using a value for the Young's modulus of $E = 2000$ pN nm^{-2} calculated from experimental data [170]. This discrepancy could be down to various factors, such as measurement of E from extension/compression data as opposed to bending rigidity, along with varying actin cross-sectional filament area A used in calculations. This seems to be partly confirmed from Ellipsoid model measurements, with the observed value of S_3 stiffer than would be expected for a uniform filament with circular cross-section. To investigate further, we turn to the expressions in Table 3.2, for which we have measured values for the Ellipsoid model above

$$B_1 = EI_1 = \frac{EA^2}{4\pi}\,, \quad B_3 = GI_3 = \frac{GA^2}{2\pi}\,, \quad S_3 = EA\,.$$

These form a set of 3 equations with 3 unknowns and, for our parameter choices, we find $E = 2360$ pN nm^{-2}, $G = 390$ pN nm^{-2}, and $A = 14.4\,\text{nm}^2$. However, this would give us an effective cross-sectional radius of 2.14 nm which is not realistic when compared with common representation of actin in the literature. This suggests the relation between stretch and bending rigidity is not consistently captured by the standard Cosserat model. We can also calculate the Poisson ratio as $\nu \approx 2$, which is not a value seen in standard materials, for which $-1 < \nu < 1/2$ tends to hold,

but has been observed in theoretical settings [176]. This all suggests that further work may be required to fully evaluate the accuracy of models approximating actin as a simple uniform material of circular cross-section.

3.3.4 Cosserat twist-stretch coupling

While our the Cosserat model is able to capture the bending dynamics displayed by the Ellipsoid model, we observed that it performs less well under torsional forces, with variable twisting rigidity. F-actin also displays variable twist-stretch behaviour, which we will investigate in this section.

When measuring the bending rigidity of the Ellipsoid model, we noticed that when a cantilever filament had a downward force applied to its end, the filament stayed within the plane of the force. This suggests the double helical nature of the filament does not cause torsion under simple bending and stretching. However, when making our measurements in Section 3.3.3, we noted the filament extended and contracted under twists of varying magnitude. This suggests that rather than acting with forces, we may be able to instead directly relate the curvature to the rest length; if $\hat{\kappa}_{\mathcal{L},3}$ is large, the filament length would naturally be smaller or larger by some factor. In practice, this can be achieved in our discretisation by first calculating the curvature using the standard rest length of the Voronoi region

$$\hat{\boldsymbol{\kappa}}^i_{\mathcal{L}} = \frac{\log\left(\boldsymbol{Q}^T_{i+1}\boldsymbol{Q}_i\right)}{\hat{\mathcal{D}}_i}.$$

We then define an arbitrary function that varies in the twist curvature

$$\xi(\hat{\kappa}_{\mathcal{L},3}) = \sum_{i=0}^{6} p_i \hat{\kappa}^i_{\mathcal{L},3} \tag{3.6}$$

where p_i are parameters to be inferred. This is the scale factor for twist induced filament stretching, e.g. if a filament has length L and a twist of $\hat{\kappa}_{\mathcal{L},3}$ causes an extension of ΔL, the scaling would be

$$\xi(\hat{\kappa}_{\mathcal{L},3}) = \frac{L + \Delta L}{L}.$$

We use this function to scale the extension quantities of the model related to the segments and Voronoi regions, labelled e_i and $\mathcal{E}_i$ respectively. As we only know

$\hat{\kappa}_{\mathcal{L},3}$ at discrete points (i.e. $[\hat{\kappa}_{\mathcal{L},3}]_j$ for $j = 1, 2, \ldots, n-1$) over the course of the filament, we estimate the curvature of the segments as

$$\hat{\kappa}_i^{\text{seg}} = \mathcal{A}_i(\{[\hat{\kappa}_{\mathcal{L},3}]_j\}) \qquad \text{for } i = 1, 2, \ldots, n\,,$$

where $\mathcal{A}_i(\cdot)$ is the trapezoidal quadrature operator defined in Section 1.4. We apply this to the extensions

$$e_i = \frac{1}{\xi(\hat{\kappa}_i^{\text{seg}})} \cdot \frac{|\boldsymbol{l}_i|}{|\hat{\boldsymbol{l}}_i|}\,, \qquad \mathcal{E}_i = \frac{e_i + e_{i+1}}{2}, \qquad \text{for } i = 1, 2, \ldots, n\,, \tag{3.7}$$

and these quantities feed in directly to the normal acceleration calculations, meaning the Cosserat model will now stretch when twisted. We now need to establish parameters for equation (3.6) based on the behaviour of the Ellipsoid model under torsion. In Table 3.5 we present data showing the change in filament length of the

Under-twist			Over-twist		
ϕ	$\kappa_{\mathcal{L},3}$	$\Delta L/L$	ϕ	$\kappa_{\mathcal{L},3}$	$\Delta L/L$
240°	0.0555	−0.0126	240°	−0.0555	−0.0434
220°	0.0508	−0.0108	220°	−0.0508	−0.0378
200°	0.0462	−0.0090	200°	−0.0462	−0.0341
180°	0.0416	−0.0075	180°	−0.0416	−0.0284
160°	0.0370	−0.0057	160°	−0.0370	−0.0256
140°	0.0324	−0.0045	140°	−0.0324	−0.0219
120°	0.0277	−0.0032	120°	−0.0277	−0.0186
100°	0.0231	−0.0020	100°	−0.0231	−0.0156
80°	0.0185	−0.0009	80°	−0.0185	−0.0121
60°	0.0139	−0.0002	60°	−0.0139	−0.0086
40°	0.0092	0.0003	40°	−0.0092	−0.0053
20°	0.0047	0.0006	20°	−0.0047	−0.0022
10°	0.0023	0.0004	10°	−0.0023	−0.0009

Table 3.5: *Compression/extension of an ellipsoidal actin filament at various curvatures calculated by inference from simulation data [2].*

Ellipsoid model under torsion, and observe it is not symmetric. The more a filament is over-twisted, the shorter it gets. However, when it is under-twisted it initially lengthens, before slightly contracting, though at a smaller magnitude than when over-

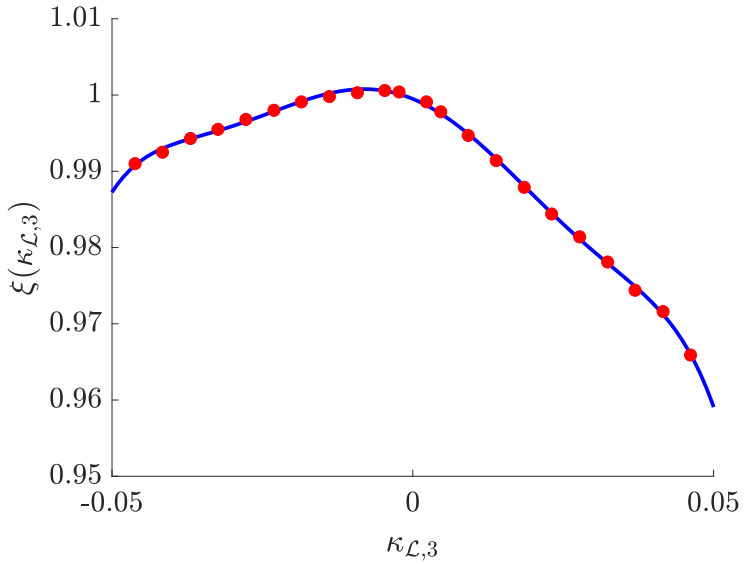


Figure 3.4: *Stretching of ellipsoidal actin filament for varying curvature. We over-twist and under-twist a* $\sim 75\,\mathrm{nm}$ *long, straight filament and measure the extension and compression of this filament under same conditions, with continuous function* $\xi(\kappa_{\mathcal{L},3})$ *fitted to data.*

twisted. We use this data to infer the correct form of $\xi(\hat{\kappa}^i_{\mathcal{L},3})$, with final parameters

$$\begin{aligned} p_6 &= -2.6662 \times 10^{6} \\ p_5 &= -2.4927 \times 10^{4} \\ p_4 &= 1.0660 \times 10^{4} \\ p_3 &= 7.9376 \times 10^{1} \\ p_2 &= -2.0508 \times 10^{1} \\ p_1 &= -3.2416 \times 10^{-1} \\ p_0 &= -5.0666 \times 10^{-4}, \end{aligned} \tag{3.8}$$

with data and fitted curve for $\xi(\hat{\kappa}^i_{\mathcal{L},3})$ presented in Figure 3.4. It is this form of twist-stretch coupling that we employ for our final adapted twist example.

Alternatively, we can consider a force based method, again depending on the curvature and extension of the Voronoi regions in which $\hat{\boldsymbol{\kappa}}^i_{\mathcal{L}}$ is defined. It possible to act on the midpoints with forces along the segment tangent, and assume the extension/compression is roughly constant over the length of a single segment. This

gives the following discretised linear and angular forces

$$\boldsymbol{l}_i = \boldsymbol{r}_{i+1} - \boldsymbol{r}_i\,, \qquad \mathcal{D}_i = \frac{|\boldsymbol{l}_{i+1}| + |\boldsymbol{l}_i|}{2} \qquad \text{and} \qquad \mathcal{E}_i = \frac{\mathcal{D}_i}{\hat{\mathcal{D}}_i},$$

$$\boldsymbol{F}_C^i = \mathcal{B}_i\left(\{\boldsymbol{Q}_j\} \cup \left\{\left[0, 0, \frac{C_3 \hat{\boldsymbol{\kappa}}_{\mathcal{L},3}^j}{\mathcal{E}_j}\right]\right\}\right),$$

$$\boldsymbol{\tau}_C^i = \Delta_i\left(\left\{\left[0, 0, \frac{-C_3(\mathcal{E}_j - 1)}{\mathcal{E}_j^3}\right]\right\}\right),$$

where C_3 is the rigidity of the twist-stretch coupling, $\Delta_i(\{\boldsymbol{x}_j\})$ is the difference operator introduced in Section 1.4, and operator

$$\mathcal{B}_i(\{\boldsymbol{Q}_j\} \cup \{\boldsymbol{x}_j\}) : \mathbb{R}^{(k-1)\times 3\times 3} \cup \mathbb{R}^{(k-2)\times 3} \to \mathbb{R}^3$$

acting on the sets $\{\boldsymbol{Q}_j\} \equiv \{\boldsymbol{Q}_1, \boldsymbol{Q}_2, \ldots, \boldsymbol{Q}_{k-1}\}$ and $\{\boldsymbol{x}_j\} \equiv \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{k-2}\}$ is defined as

$$\mathcal{B}_i(\{\boldsymbol{Q}_j\} \cup \{\boldsymbol{x}_j\}) = \begin{cases} \boldsymbol{Q}_1\boldsymbol{x}_1/2, & \text{if } i = 1, \\ \boldsymbol{Q}_1\boldsymbol{x}_1/2 + \boldsymbol{Q}_2(\boldsymbol{x}_2 - \boldsymbol{x}_1)/2, & \text{if } i = 2, \\ \boldsymbol{Q}_{i-1}(\boldsymbol{x}_{i-1} - \boldsymbol{x}_{i-2})/2 + \boldsymbol{Q}_i(\boldsymbol{x}_i - \boldsymbol{x}_{i-1})/2, & \text{if } 2 < i < k-1, \\ \boldsymbol{Q}_{k-2}(\boldsymbol{x}_{k-2} - \boldsymbol{x}_{k-3})/2 - \boldsymbol{Q}_{k-1}\boldsymbol{x}_{k-2}/2, & \text{if } i = k-1, \\ -\boldsymbol{Q}_{k-1}\boldsymbol{x}_{k-2}/2, & \text{if } i = k. \end{cases}$$

However, based on the non-symmetric over-twist/under-twist behaviour seen in Table 3.5 we can reasonably conclude that it is possible that the rigidity of the twist-stretch coupling varies with the magnitude of the twist $C_3(\kappa_{\mathcal{L},3}, \sigma_{\mathcal{L},3})$, or may even require all components to be coupled $\boldsymbol{C}(\boldsymbol{\kappa}_{\mathcal{L}}, \boldsymbol{\sigma}_{\mathcal{L}})$ to fully capture the behaviour of the Ellipsoid model. Additionally, the inclusion of the twist-stretch coupling affects the twisting rigidity $B_3(\kappa_{\mathcal{L},3})$, and cannot therefore not be parametrised independently. There is a potential future area of study by incorporating accurate estimates for $\boldsymbol{C}(\boldsymbol{\kappa}_{\mathcal{L}}, \boldsymbol{\sigma}_{\mathcal{L}})$ and $\boldsymbol{B}(\boldsymbol{\kappa}_{\mathcal{L}})$ into dynamical rod models of actin filaments, though for the examples presented later in Chapter 4 we keep $C_3(\kappa_{\mathcal{L},3}) = 0$.

3.4 Discussion

We have presented a multi-resolution model for actin filaments, using a rod-based Cosserat model [98] for low resolution regions and a monomer-based Ellipsoid

model [97, 119] for high resolution regions. The coupling region is defined such that two monomers at the end of the ellipsoidal filament are attached to the final segment of a Cosserat filament. The forces at this interface constrain linear and angular movement of the models in a manner consistent with the Cosserat formulation. The parameters for the standard Cosserat model are inferred directly from test simulations of the Ellipsoid model. This allows us to match both the dynamics and equilibrium structures for filaments of differing resolutions, along with having closely matching energies.

However, when placing the filaments under torsion, the standard Cosserat model is unable to capture the anisotropies of the Ellipsoid model. To address this, we present an augmented Cosserat model where the twisting rigidity $B_3(\kappa_{\mathcal{L},3})$ is dependent on the twist curvature $\kappa_{\mathcal{L},3}$. This allows us to account for the fact that it is easier to under-twist actin than it is to over-twist it. In addition, we act on the length of the filament via a newly introduced scaling $\xi(\kappa_{\mathcal{L},3})$, again motivated by the monomer-based representation of actin, where twist-stretch behaviour is observed. We present functions for both $B_3(\kappa_{\mathcal{L},3})$ and $\xi(\kappa_{\mathcal{L},3})$, with their parameters inferred from data collected by placing the Ellipsoid model under twists of varying magnitude.

If we employ these extensions, the original energy expression 1.11 for the Cosserat model no longer hold [98]. In effect, we are letting the twisting rigidity $B_3(\kappa_{\mathcal{L},3})$ and strain $\boldsymbol{\sigma}_{\mathcal{L}}(\kappa_{\mathcal{L},3})$, via the scaling $\xi(\kappa_{\mathcal{L},3})$, vary in twist curvature, resulting in the updated energy for the augmented Cosserat model

$$E = \frac{1}{2}\int_0^L \boldsymbol{\kappa}_{\mathcal{L}}^T \boldsymbol{B}(\kappa_{\mathcal{L},3})\boldsymbol{\kappa}_{\mathcal{L}}\,\mathrm{d}s + \frac{1}{2}\int_0^L \boldsymbol{\sigma}_{\mathcal{L}}(\kappa_{\mathcal{L},3})^T \boldsymbol{S}\boldsymbol{\sigma}_{\mathcal{L}}(\kappa_{\mathcal{L},3})\,\mathrm{d}s.$$

We can still gain clear insight into the bend and twist properties of the filament via $\boldsymbol{B}(\kappa_{\mathcal{L},3})$. However, it is less simple to separate the general shear and stretch properties (determined by $\boldsymbol{S}$) from the twist-stretch coupling behaviour introduced by using the non-linear $\boldsymbol{\sigma}_{\mathcal{L}}(\kappa_{\mathcal{L},3})$ strain term. Rather than having curvature dependent strain, it is also possible to create a twist-stretch coupling by introducing an extra energy term $C_3(\kappa_{\mathcal{L},3}, \sigma_{\mathcal{L},3})\,\sigma_{\mathcal{L},3}\,\kappa_{\mathcal{L},3}$, which would allow its effect to be quantified separately via the coupling rigidity $C_3(\kappa_{\mathcal{L},3}, \sigma_{\mathcal{L},3})$. In Section 3.3.4 we

include a basic dynamic implementation, with forces based on this energy term. However, it is a non-trivial task to infer the form of $C_3(\kappa_{\mathcal{L},3}, \sigma_{\mathcal{L},3})$ to accurately match the Cosserat model to the Ellipsoid model; the inclusion of this twist-stretch coupling affects the inference of the twisting rigidity $B_3(\kappa_{\mathcal{L},3})$, and these functional forms must be parametrised together. While this was beyond the scope of this work, a systematic study to determine $\boldsymbol{B}(\boldsymbol{\kappa}_{\mathcal{L}})$, $\boldsymbol{S}(\boldsymbol{\sigma}_{\mathcal{L}})$, and $\boldsymbol{C}(\boldsymbol{\kappa}_{\mathcal{L}}, \boldsymbol{\sigma}_{\mathcal{L}})$ by direct measurement from the Ellipsoid model would enable the formulation of a class of further augmented Cosserat models with energy

$$E = \frac{1}{2}\int_0^L \boldsymbol{\kappa}_{\mathcal{L}}^T \boldsymbol{B}(\boldsymbol{\kappa}_{\mathcal{L}})\boldsymbol{\kappa}_{\mathcal{L}}\,\mathrm{d}s + \frac{1}{2}\int_0^L \boldsymbol{\sigma}_{\mathcal{L}}^T \boldsymbol{S}(\boldsymbol{\sigma}_{\mathcal{L}})\boldsymbol{\sigma}_{\mathcal{L}}\,\mathrm{d}s + \int_0^L \boldsymbol{\kappa}_{\mathcal{L}}^T \boldsymbol{C}(\boldsymbol{\kappa}_{\mathcal{L}}, \boldsymbol{\sigma}_{\mathcal{L}})\boldsymbol{\sigma}_{\mathcal{L}}\,\mathrm{d}s,$$

which incorporate the curvature and strain dependence of the material properties, along with complex coupled filament behaviour. This more generalised form could better capture the anisotropies of F-actin, at a fraction of the computational expense of the Ellipsoid model.

Additionally, this could be extended to studies of other bio-filaments with non-trivial physical properties like filopodia [177], or even more flexible filaments such as hair fibres [113, 117]. Another approach to actin filament modelling is presented by Yamaoka et al. [112] where, rather than varying the twist rigidity $B_3(\kappa_{\mathcal{L},3})$, the intrinsic twist curvature of actin is incorporated into the formulation of a Cosserat model by considering the mismatch between the curve passing through monomer centres of mass and the filament centre-line. Alternatively, work has been done modelling cytoskeleton filaments using inextensible rod models [178] while accounting for hydrodynamics. We did not explore inextensible interactions in our examples cases, though using the constraint algorithms presented in Appendix B.2, we could enforce $e_i = 1$ for all segment extensions.

4

Multi-Resolution Actin Modelling Applications and Extensions

In this chapter we return to the multi-resolution model of F-actin introduced in Chapter 3, based on my work submitted for publication as R. S. Gunaratne et al. "On Stretching, Bending, Shearing and Twisting of Actin Filaments II: Multi-Resolution Modelling" [2], in which all featured figures appear, with reference to accompanying paper C. Floyd et al. "On Stretching, Bending, Shearing and Twisting of Actin Filaments I: Variational Models" [3]. This chapter will outline how to simulate the multi-resolution model such that we have consistent filament dynamic properties, along with presenting a set of example cases assess the performance of the model. In Section 4.1 we present symplectic integrators schemes used to solve the model, along with other numerical methods used in simulations. We then consider the effects of mass, inertia, and dissipative forces on filament dynamics in Section 4.2. Having established our numerical solving methodology, we present the simulated results for a set of example scenarios of biological relevance in Section 4.3. This allows us to assess the performance of the multi-resolution model, with a full discussion in Section 4.4 on how extensions to the Cosserat model seen in Section 3.3 improve model performance. In addition, we consider further

extensions to the multi-resolution model, such as an adaptive resolution scheme and possible biological applications.

4.1 Integration of equations of motion for multi-resolution model

We have a reasonable sense of the accuracy and stability of the integration methods used in the literature for the Ellipsoid and Cosserat models, so by introducing coupling forces with rigidities which are consistent with both the Cosserat and Ellipsoid models, we would expect to maintain accuracy so long as the integrating schemes are effective for each (single resolution) model. In Section 4.1.1 we give a number of symplectic integrators which may be used to solve the equations of motion presented in Section 1.4. Symplectic integrators conserve energy for Hamiltonian systems, and are therefore widely used in MD simulations. Other numerical methods used are given in Section 4.1.2.

4.1.1 Symplectic integrators

A sensible starting point is a position Verlet scheme, with discretisation for the Cosserat model given by

$$\boldsymbol{r}_i\left(t+\tfrac{1}{2}\Delta t\right)=\boldsymbol{r}_i(t)+\tfrac{1}{2}\boldsymbol{v}_i(t)\Delta t, \tag{4.1}$$

$$\boldsymbol{Q}_i\left(t+\tfrac{1}{2}\Delta t\right)=\exp\left(\tfrac{1}{2}\boldsymbol{\omega}^i_{\mathcal{L}}(t)\Delta t\right)\boldsymbol{Q}_i(t), \tag{4.2}$$

$$\boldsymbol{v}_i\left(t+\Delta t\right)=\boldsymbol{v}_i(t)+\frac{1}{m_i}\left[\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}\left(t+\tfrac{1}{2}\Delta t\right)\right]\Delta t, \tag{4.3}$$

$$\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\Delta t\right)=\boldsymbol{\omega}^i_{\mathcal{L}}\left(t\right)+\boldsymbol{I}_i^{-1}\left[\frac{\mathrm{d}\boldsymbol{\omega}^i_{\mathcal{L}}}{\mathrm{d}t}\left(t+\tfrac{1}{2}\Delta t\right)\right]\Delta t, \tag{4.4}$$

$$\boldsymbol{r}_i\left(t+\Delta t\right)=\boldsymbol{r}_i\left(t+\tfrac{1}{2}\Delta t\right)+\tfrac{1}{2}\boldsymbol{v}_i\left(t+\Delta t\right)\Delta t, \tag{4.5}$$

$$\boldsymbol{Q}_i\left(t+\Delta t\right)=\exp\left(\tfrac{1}{2}\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\Delta t\right)\Delta t\right)\boldsymbol{Q}_i\left(t+\tfrac{1}{2}\Delta t\right). \tag{4.6}$$

When calculating the change in velocity in equation (4.3) and equation (4.4) we use the linear and angular derivatives of velocity in time given in equation (1.9) and equation (1.10) respectively. The orientation bases are updated in equation (4.2) and

equation (4.6), using the Rodrigues formula[164] to calculate the matrix exponential for each timestep, which we will set out in Section 4.1.2. When simulating our examples in Section 4.3, we set the position of the harmonic bond anchor points using equation (3.1), based on a single rotation and translation of their positions at $t = 0$, rather than explicitly acting on each point individually to improve numerical accuracy. This means that equations (4.1)–(4.6) can also be used to integrate the fully multi-resolution model, using accelerations in equation (1.14).

We can also use a velocity Verlet scheme, allowing us to carry out a single update of the positions and orientation matrices $\boldsymbol{Q}_i$ per timestep:

$$\boldsymbol{v}_i\left(t+\tfrac{1}{2}\Delta t\right) = \boldsymbol{v}_i(t) + \frac{1}{2m_i}\left[\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}(t)\right]\Delta t,$$

$$\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\tfrac{1}{2}\Delta t\right) = \boldsymbol{\omega}^i_{\mathcal{L}}(t) + \frac{1}{2}\boldsymbol{I}_i^{-1}\left[\frac{\mathrm{d}\boldsymbol{\omega}^i_{\mathcal{L}}}{\mathrm{d}t}(t)\right]\Delta t,$$

$$\boldsymbol{r}_i(t+\Delta t) = \boldsymbol{r}_i(t) + \boldsymbol{v}_i\left(t+\tfrac{1}{2}\Delta t\right)\Delta t,$$

$$\boldsymbol{Q}_i(t+\Delta t) = \exp\left(\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\tfrac{1}{2}\Delta t\right)\Delta t\right)\boldsymbol{Q}_i(t),$$

$$\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i\left(t+\tfrac{1}{2m_i}\Delta t\right) + \frac{1}{2}\left[\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}(t+\Delta t)\right]\Delta t,$$

$$\boldsymbol{\omega}^i_{\mathcal{L}}(t+\Delta t) = \boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\tfrac{1}{2}\Delta t\right) + \frac{1}{2}\boldsymbol{I}_i^{-1}\left[\frac{\mathrm{d}\boldsymbol{\omega}^i_{\mathcal{L}}}{\mathrm{d}t}(t+\Delta t)\right]\Delta t.$$

In both of these cases we run into difficulties when updating the angular acceleration for the Cosserat rod model, as they are not solely dependent on position, and also vary in angular and linear velocity. One way around this is to rely on estimations of the velocities. In the case of the position Verlet scheme, we must estimate the velocities as follows:

$$\boldsymbol{v}_i\left(t+\tfrac{1}{2}\Delta t\right) = \boldsymbol{v}_i(t) + \frac{1}{2m_i}\left[\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}(t)\right]\Delta t,$$

$$\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\tfrac{1}{2}\Delta t\right) = \boldsymbol{\omega}^i_{\mathcal{L}}(t) + \frac{1}{2}\boldsymbol{I}_i^{-1}\left[\frac{\mathrm{d}\boldsymbol{\omega}^i_{\mathcal{L}}}{\mathrm{d}t}(t)\right]\Delta t.$$

Similarly, for the velocity Verlet scheme:

$$\boldsymbol{v}_i(t+\Delta t) = \boldsymbol{v}_i(t) + \frac{1}{m_i}\left[\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}(t)\right]\Delta t,$$

$$\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\Delta t\right)=\boldsymbol{\omega}^i_{\mathcal{L}}\left(t\right)+\boldsymbol{I}_i^{-1}\left[\frac{\mathrm{d}\boldsymbol{\omega}^i_{\mathcal{L}}}{\mathrm{d}t}\left(t\right)\right]\Delta t.$$

It may therefore more desirable to use the position Verlet, as the estimation timestep is half the size as that required in the velocity Verlet scheme, which should result in better accuracy in this situation. However, this requires two rotation calculations for the basis matrices every timestep, which is reasonably expensive, and may therefore negate this benefit. We do not encounter these problems for the Ellipsoid model as the force expression only depends on position.

It is possible to further increase stability by employing higher order symplectic integrators. For each stage of a symplectic integrator scheme we take the following steps in time

$$\boldsymbol{v}_i\left(t+\Delta t\sum_{k=1}^{j}a_k\right)=\boldsymbol{v}_i\left(t+\Delta t\sum_{k=1}^{j-1}a_k\right)+\frac{a_j}{m_i}\left[\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t}\left(t+\Delta t\sum_{k=1}^{j-1}b_k\right)\right]\Delta t \tag{4.7}$$

$$\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\Delta t\sum_{k=1}^{j}a_k\right)=\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\Delta t\sum_{k=1}^{j-1}a_k\right)+a_j\boldsymbol{I}_i^{-1}\left[\frac{\mathrm{d}\boldsymbol{\omega}^i_{\mathcal{L}}}{\mathrm{d}t}\left(t+\Delta t\sum_{k=1}^{j-1}b_k\right)\right]\Delta t \tag{4.8}$$

$$\boldsymbol{r}_i\left(t+\Delta t\sum_{k=1}^{j}b_k\right)=\boldsymbol{r}_i\left(t+\Delta t\sum_{k=1}^{j-1}b_k\right)+b_j\,\boldsymbol{v}_i\left(t+\Delta t\sum_{k=1}^{j}a_k\right)\Delta t \tag{4.9}$$

$$\boldsymbol{Q}_i\left(t+\Delta t\sum_{k=1}^{j}b_k\right)=\exp\left(b_j\,\boldsymbol{\omega}^i_{\mathcal{L}}\left(t+\Delta t\sum_{k=1}^{j}a_k\right)\Delta t\right)\boldsymbol{Q}_i\left(t+\Delta t\sum_{k=1}^{j-1}b_k\right) \tag{4.10}$$

for some set of coefficients a_j, b_j, with $j=1,...,n$, where n is the number of stages for the integrator (which generally corresponds directly with the order of the scheme). In the case of the Verlet integrators above, we carry out $n=2$ repeats of the above for each timestep, with $a_1=a_2=1/2, b_1=1, b_2=0$ for the velocity Verlet scheme, and $b_1=b_2=1/2, a_1=0, a_2=1$ for the position Verlet scheme. We see that following the condition holds for these cases

$$\sum_j a_j=\sum_j b_j=1$$

meaning that at the end of each round the algorithm takes a full timestep. These coefficients can be calculated by solving systems of constraint equations, including the above, to find consistent choices for a_j and b_j. In this work we use a 4^{th}

order symplectic integrator presented by Omelyan, Mryglod and Folk [163], with integrator coefficients given by

$$\begin{aligned} a_i &= [\xi, \chi, 1-2(\xi+\chi), \chi, \xi] \\ b_i &= [(1-2\lambda)/2, \lambda, \lambda, (1-2\lambda)/2] \\ \xi &= 0.1786178958448091 \\ \lambda &= -0.2123418310626054 \\ \chi &= -0.06626458266981849 \end{aligned}$$

These coefficient are the solution to a complex set of equations which ensure the local truncation error term is $\mathcal{O}(\Delta t^5)$, along with the global error of $\mathcal{O}(\Delta t^4)$. This can be verified by considering a simple harmonic oscillator example where $\ddot{X} = k^2 X$, and letting $X_i = X(t + \Delta t \sum_{j=1}^{i} b_j)$, $V_i = V(t + \Delta t \sum_{j=1}^{i} a_j)$, we substitute the position steps of the scheme to get

$$X(t+\Delta t) = X(t) + \Delta t \sum_{i=1}^{5} a_i V_{i-1}.$$

Then, by taking the velocity steps we simplify this as

$$X(t+\Delta t) = X(t) + V(t)\Delta t + (\Delta t)^2 \left(\sum_{i=1}^{4} \sum_{j=1}^{5-i} a_{6-j} b_i A_i \right),$$

where $A_i = A(X_i)$ is the acceleration. Substituting in the coefficients a_i, b_i this becomes

$$\begin{aligned} X(t+\Delta t) &= X(t) + V(t)\Delta t \\ &\quad + (\Delta t)^2 \Big(\xi(\tfrac{1}{2} - \lambda)(A_4 - A_1) + \lambda(\chi+\xi)(A_3 - A_2) + \lambda(A_2 - A_1) + \tfrac{1}{2} A_1 \Big), \end{aligned}$$

For our example $A_i = k^2 X_i$, so can make a substitution to simplify the expression

$$\begin{aligned} X(t+\Delta t) &= X(t) + V(t)\Delta t + \frac{k^2}{2} X(t)(\Delta t)^2 \\ &\quad + k^2 (\Delta t)^3 \Big(\xi(\tfrac{1}{2} - \lambda)(V_1 \chi + V_2(1 - 2(\chi+\xi)) + V_3 \chi) \\ &\quad + V_2 \lambda(\chi+\xi)(1-2(\chi+\xi)) + V_1 \chi \lambda + \tfrac{1}{2} V_0 \xi \Big). \end{aligned}$$

For these choices of ξ, λ and χ we calculate that

$$\xi + 2\chi\lambda - \xi^2 - 2\chi^2\lambda - 4\xi\lambda\chi = \frac{1}{6},$$

and can therefore further reduce the equations beyond the $(\Delta t)^3$ term to get

$$\begin{aligned} X(t+\Delta t) = X(t) + V(t)\Delta t + \frac{k^2}{2}X(t)(\Delta t)^2 + \frac{k^4}{6}X(t)(\Delta t)^3 \\ + k^4(\Delta t)^4\Big(X_1(\tfrac{1}{2}-\lambda)(\tfrac{1}{2}\xi + 2\chi\lambda - \xi^2 - 2\chi^2\lambda - 4\xi\lambda\chi) \\ + X_2\lambda(\tfrac{1}{2}\xi - \xi^2 - \tfrac{1}{2}\chi\xi - 2\chi^2\lambda - 3\xi\chi\lambda + \chi\lambda) \\ + X_3\lambda(\xi\chi(\tfrac{1}{2}-\lambda))\Big). \end{aligned}$$

Similarly, for these choices of ξ, λ and χ we calculate that

$$\frac{1}{4}\xi + \chi\lambda - \frac{1}{2}\xi^2 - 2\chi^2\lambda - 2\xi\chi\lambda = \frac{1}{24},$$

finally giving us confirmation of the stability of the integrator

$$\begin{aligned} X(t+\Delta t) = X(t) + V(t)\Delta t + \frac{k^2}{2}X(t)(\Delta t)^2 \\ + \frac{k^4}{6}V(t)(\Delta t)^3 + \frac{k^4}{24}X(t)(\Delta t)^4 + \mathcal{O}((\Delta t)^5) \end{aligned}$$

It is also possible to calculate coefficients for 3$^{\text{rd}}$ order cases or even high order schemes depending on the forces involved in the situation investigated [179]. For the multi-resolution model we found that choosing a 4$^{\text{th}}$ order scheme provided good efficiency and stability, with the extra steps not adversely affecting the simulation speed and allowing for larger timesteps to be used while maintaining stability. We summarise the full multi-resolution algorithm for simulating F-actin in Table 4.1.

4.1.2 Other numerical methods

There are additional numerical methods that are used in the course of simulating the multi-resolution filaments. For the coupling region, Ellipsoid, and Cosserat models we must calculate the matrix exponential when rotating their orientation matrices $\boldsymbol{Q} \in \mathbb{R}^{3\times3}$ at each time step as seen in, for example, equation (4.2). We use its inverse, the matrix logarithm, to calculate the discrete curvature $\boldsymbol{\kappa}_{\mathcal{L}}$ for the Cosserat model. We use Rodrigues' formula [164] for both of these cases, as described in Gazzola et al. [98]. The matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ representing a rotation of θ around a unit vector $\boldsymbol{u}$ is given by the matrix exponential, which we define as

$$\boldsymbol{R} = \exp(\theta\boldsymbol{U}) = \mathbf{1} + \sin\theta\boldsymbol{U} + (1-\cos\theta)\boldsymbol{U}\boldsymbol{U},$$

[MR-A1]	Commence stage j of symplectic integrator scheme with a velocity step of size a_j using equation (4.7) for the linear velocity and equation (4.8) for the angular velocity.
[MR-A2]	Continue stage j of symplectic integrator scheme with a position step of size b_j using equation (4.9) for the linear position and equation (4.10) for the orientation.
[MR-A3]	Calculate new acceleration using position and velocity coordinates calculated in [MR-A1] and [MR-A2]. For the Ellipsoid model, we use equation (1.14), the Cosserat model equations (1.9)–(1.10), and finally for the coupling region coordinates we use equations (3.1)–(3.3).
[MR-A4]	Return to step [MR-A1] for next stage of symplectic integrator scheme, with $j = j + 1$. If full round of algorithm is complete, i.e. j is greater than number of velocity steps in scheme, continue to step [MR-A5].
[MR-A5]	Return to step [MR-A1] using time $t = t + \Delta t$ and $j = 1$.

Table 4.1: *One iteration of the multi-resolution simulation algorithm for F-actin filaments, describing the velocity step, position step, and acceleration update for all elements of the model.*

where $\boldsymbol{U} = \mathrm{ax}(\boldsymbol{u})$. Inverting this, we can find a consistent expression for the matrix logarithm

$$\log(\boldsymbol{R}) = \frac{\theta}{2\sin\theta}\mathrm{skew}(\boldsymbol{R} - \boldsymbol{R}^T)\,, \quad \theta = \arccos\left(\frac{\mathrm{tr}(\boldsymbol{R}) - 1}{2}\right)\,,$$

To avoid numerical errors close to $\theta = 0$, we use the following Taylor expansion

$$\frac{\theta}{2\sin\theta} \approx \frac{1}{2} + \frac{1}{12}\theta^2 + \frac{7}{720}\theta^4 + \frac{31}{30240}\theta^6\,.$$

We also periodically check to see if our orientation bases still satisfy the condition for orthogonality $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I} \in \mathbb{R}^{3\times3}$. If they have drifted due to numerical errors in integration, it is possible to calculate the nearest orthogonal matrix

$$\boldsymbol{Q}_{\mathrm{new}} = \min_{\boldsymbol{Q}^T\boldsymbol{Q}=\boldsymbol{I}_{3\times3}} ||\boldsymbol{Q}_{\mathrm{old}} - \boldsymbol{Q}||_F$$

where $||\cdot||_F$ is the Frobenius norm, defined as

$$||\boldsymbol{A}||_F = \sqrt{\sum_i \sum_j |A_{ij}|^2}.$$

There are a number of ways of achieving this, but the simplest is via calculation of the singular value decomposition, with the new orthogonal matrix given by

$$\boldsymbol{Q}_{\mathrm{old}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \Rightarrow \boldsymbol{Q}_{\mathrm{new}} = \boldsymbol{U}\boldsymbol{V}^T$$

In our implementation we use the in-built MATLAB [180] function to carry out the decomposition.

4.2 Effect on dynamics due to dissipation, mass, and inertia

Having set out numerical methods to accurately integrate the equations of motion for the multi-resolution model, we must now ensure that the macroscopic dynamic behaviour of the filaments is consistent. F-actin in the cell cytoskeleton do not move freely and experience solvent interactions. A simple method to partially account for this in our simulations is to include dissipative forces dependent on the angular and linear velocities

$$\boldsymbol{F}_{\mathrm{v}} = -\gamma_v \boldsymbol{v}\,, \quad \boldsymbol{F}_{\omega} = -\gamma_w \boldsymbol{\omega}\,,$$

This can be extended to include a stochastic element, as seen in Chapter 2, by using the Langevin equation

$$\boldsymbol{F}_{\mathrm{v}} = -\gamma_v \boldsymbol{v} + \gamma_v \sqrt{2D}\boldsymbol{W}$$

where $\boldsymbol{W}$ denotes a Wiener process. This better mimics the random solvent interactions, though further improvements could be made by accounting for the hydrodynamics of the system. In this work, we chose to use simple dissipation, with $\gamma_v = \gamma_\omega = \gamma$. This is as we are principally concerned with the relative dynamics of the Ellipsoid and Cosserat models, though we will discuss this further in Section 4.4.

In Figure 4.1 we consider a cantilever filament represented using the Cosserat model. Figure 4.1(a) shows that as the dissipation constant is increased the amount of oscillation decreases, with no oscillation visible when $\gamma = 0.5$. For cases where oscillation occur, i.e with low dissipation, the system can be said to be underdamped. When the dissipation is high the system is overdamped, with no oscillations occurring. When γ is just high enough for oscillations to stop the system is said to be critically damped. For our example cases we chose values of γ which resulted in both over and underdamped systems. We found that choosing higher damping allowed us to

reach equilibrium, while also improving stability (due to lower max velocities, etc.) meaning larger timesteps could be used. However, if the dissipative force is too strong it can result in the system getting stuck in local minima, e.g. a compressed rod becoming very kinked, rather than simple buckling. While this may result in various different equilibria being reached for complex systems of actin filaments, we found that for simple examples (like in Section 4.3) this had little bearing on the final equilibrium coordinates. In Figure 4.1(b), we see the filament orientation

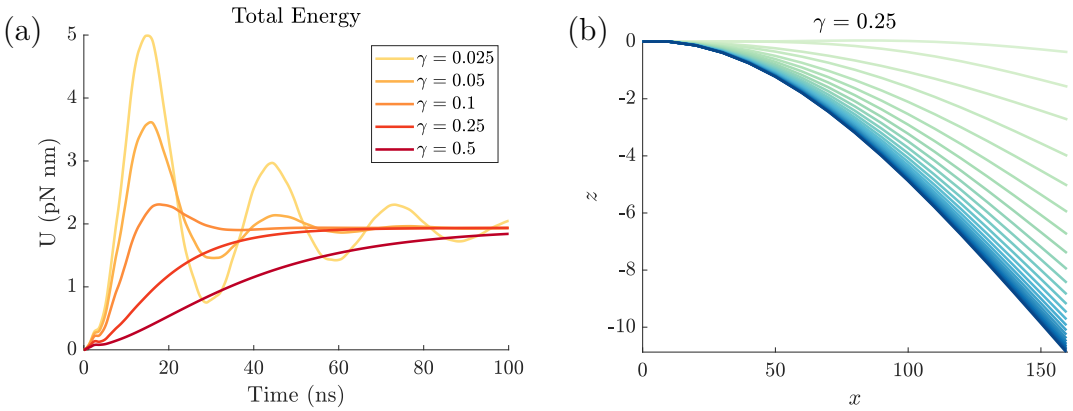


Figure 4.1: *(a) Filament energy changing over time for a cantilever filament described using the Cosserat model for various values of dissipation constant γ.*
(b) Case where $\gamma = 0.25$, with the filament orientation plotted across 20 equally spaced intervals of 5 ns*, with the equilibrium shown in dark blue.*

changing over time for the same cantilever example, with $\gamma = 0.25$. The lightest coloured curves signify the filament position in the early part of the simulation, with the darkest curves being when the filament is reaching equilibrium. The gaps between the curves are largest (i.e. the velocity is fastest) early on in the simulation, and slows as the forces involved become more balanced near equilibrium.

The other main factors influencing the dynamics of the simulations are the masses and inertias of the Cosserat segments and ellipsoid monomers. Though varying these will not generally affect the final equilibrium coordinates in simple filament studies, they do have a bearing on the rate at which they are reached. In the case of the multi-resolution model, this could mean one section of a filament could move faster than the rest. The Cosserat segment and Ellipsoid monomer

masses are, respectively, given by

$$m = \rho A|\hat{\boldsymbol{l}}_i|\,, \quad M = \frac{2.76}{|\hat{\boldsymbol{l}}_i|}m$$

with the ellipsoid mass set using the length increase for each monomer added. This ensures the filament sections of differing resolution have the same mass per unit length. The speed of simulations can be toggled by setting the mass density, and we used arbitrary choices $\rho = 10^{-2}$ with timestep $\Delta t = 2 \times 10^{-2}$ to ensure stability (roughly twice as large as that required when using the velocity Verlet scheme, compared to [163]). We then scale the simulation time units using the actual value of the density to infer the physically relevant timescale. The mass of a G-actin monomer is approximately 42 kDa, and factoring this into calculations for the segment density ρ tells us that our simulation timestep is approximately 1 ps when no dissipative force is employed. This seems reasonable for a dynamic model, as this is much larger than the timesteps required for stability for all atom simulations, which are in the 1–10 fs range. The length of the simulations for our investigations in Section 4.3 is 100 ns, so it seems with further optimisation (e.g. coding in a faster programming language like `C`/`C++`, and parallelisation for multi-filament situations) these investigations could feasibly be extended to μs timescale simulations.

The inertia for the model is set by defining the inertia for the Cosserat model as described in Section 3.3.2 and scaling this by $2.76/|\hat{\boldsymbol{l}}_i|$ to give the inertia for monomers in the Ellipsoid model, rather than explicitly calculate this using the spatial properties of an ellipsoid. This allows the multi-resolution dynamics to be matched and is fairly realistic as we can assume the choice of ellipsoid shape was arbitrary, and can be thought to have a non-uniform mass distribution. The linear and angular dissipation terms for the models are also scaled in this manner, which in a sense means we are scaling the velocities of the coordinates at each timestep, rather than their momenta. We will see in the coming examples in Section 4.3 that this leads to well matched simulation dynamics.

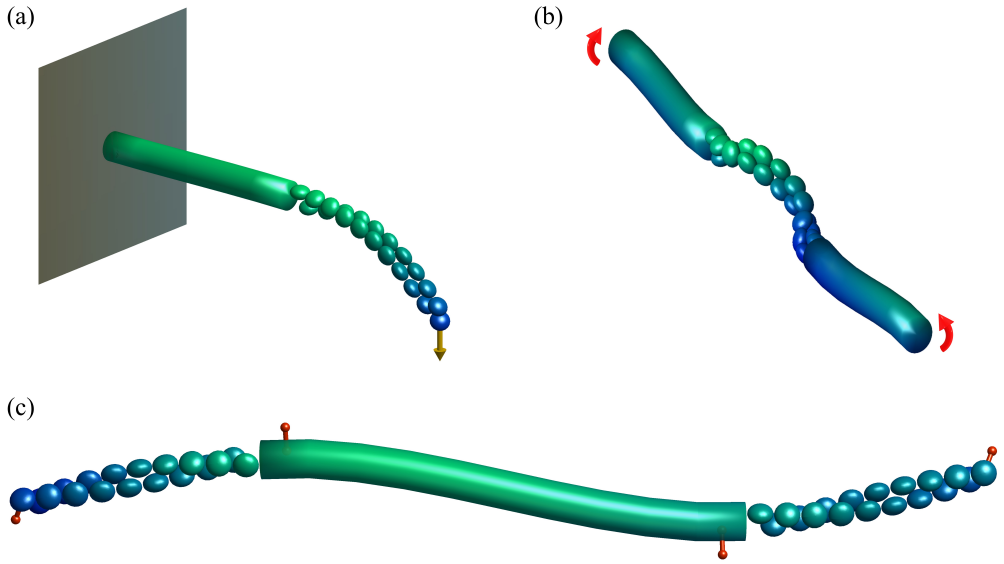


Figure 4.2: *Illustration of example cases for multi-resolution actin filaments.*
(a) *A cantilever filament attached to a wall (grey) at one end with a downward force at the other, causing it to bend.*
(b) *A filament under compression and twisting forces, causing a helical buckling.*
(c) *A filament attached to anchor points (red dots) by external linker bonds.*

4.3 Example cases

Having established how to numerically integrate the equations of motion for our multi-resolution model, we present a set of examples which allow us to investigating the bending, twisting, shearing, and stretching properties of the filaments. In particular, we assess whether using the coupling formulated in Section 3.2 adversely affects the dynamic properties of our filaments, along with studying the effect of varying resolution. In Figure 4.2 we see illustrative representations of the multi-resolution model of F-actin, showing a filament attached to a wall at one end under a cantilever force, one under compression and twisting to cause helical buckling, and finally one acted on by external linker forces. For each case we study the dynamics and static equilibrium (where dissipation is employed) of F-actin filaments resolved in varying degrees of detail. We use the parameters and methods outlined in Section 3.3.3 for the simulations in Section 4.3.1 and Section 4.3.2. In addition, methods from Section 3.3.4 i.e. the extensions to the Cosserat model for variable twisting rigidity and twist-stretch coupling, are used in Section 4.3.3. We

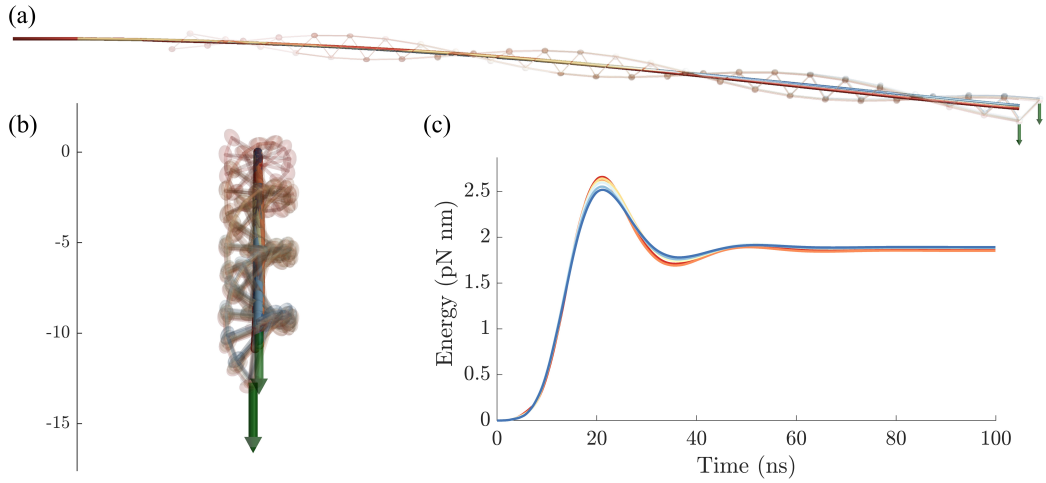


Figure 4.3: (a) *Multi-resolution actin filament under a cantilever force. We overlay the results of* 7 *simulations, with the Cosserat model used to resolve the filament for between* 20–140 nm*, with the remainder resolved using the Ellipsoid model, giving a* 160 nm *long filament in each case. The colours vary through red-yellow-blue, with red almost entirely monomer-based and blue almost entirely rod-based. Filaments are initialised at rest in a straight configuration and allowed to reach equilibrium under the external forces.*
(b) *View along the filament from the right of panel* (a).
(c) *The total filament energy* (*in* pN nm) *varying with time* (*in* ns) *as the system approaches the equilibrium.*

implemented our numerical scheme in MATLAB [180] to simulate these example cases. We utilised very few in-built functions, making these methods should be easily transferable to other programming languages.

4.3.1 Cantilever filament

We begin with a simple cantilever filament case, where F-actin is attached to a wall at one end, see Figure 4.2(a). This is achieved by setting both the angular and linear velocity at the wall to zero throughout the simulation. At the other end, a downward (z-direction) force is applied, causing the filament to bend. This allows us to establish the behaviour of the multi-resolution model under a simple bending deformation, and also resembles situations where actin filaments are attached to the cell membrane.

In Figure 4.3 we see filaments of length $L = 160$ nm under a cantilever force of magnitude $F = 0.35$ pN, reaching an equilibrium position with the filament bent downwards. The final spatial coordinates are visualised in Figure 4.3(a), with the results from 7 simulations of varying resolution overlayed. For the Cosserat model we

do not visualise full cylinders, but only their centre-line. For the Ellipsoid model we plot the monomer centres of mass, connected by thin cylinders representing lateral and longitudinal interactions. We also fit a curve to approximate the ellipsoidal filament centre-line to allow easier comparison between resolutions. We see that the results from these simulations overlap, indicating the accuracy of the multi-resolution methodology. This is confirmed in Figure 4.3(b) where the same results are visualised from a different viewpoint. Finally Figure 4.3(c) illustrates that the dynamical properties are also matched well for all cases. For all levels of resolution, we have agreement with the theoretically expected deflection

$$\Delta L_{\mathrm{D}} = \frac{FL^3}{3EI} \approx 10\,\mathrm{nm}$$

where $E = 330\,\mathrm{pN\,nm^{-2}}$ is the Young's modulus, and $I = A^2/4\pi \approx 120\,\mathrm{nm}^4$ is the relevant area moment of inertia (which corresponds to the first and second diagonal elements of the bending rigidity matrix $\boldsymbol{B}$ of the Cosserat model) which confirms that the bending parameters for the two models are correctly coupled.

4.3.2 External linker forces

Having investigated the effects of a force on a filament anchored to a wall, we now consider a free filament attached to multiple external linkers, causing non trivial bending behaviour. This is a more realistic study of the forces an F-actin filament would experience in the cytoskeleton. We define a general binding site in terms of the existing Cosserat segment endpoints and orientation basis as follows

$$\boldsymbol{r}_{\mathrm{ext}} = \boldsymbol{r}_i + A_0(\boldsymbol{r}_{i+1} - \boldsymbol{r}_i) + R(\boldsymbol{d}_{1,i}\cos(\phi_0) + \boldsymbol{d}_{2,i}\sin(\phi_0))$$

where $A_0 \in [0, 1]$ defines the cross-sectional location of the binding site along the length of the segment, R its radial distance from the filament centre-line, and ϕ_0 the angle in the cross-section the point forms relative to $\boldsymbol{d}_{1,i}$. This allows torsional effects of linkers on the Cosserat filament to be accurately modelled. As actin has high shear rigidity it is reasonably safe to assume that $\boldsymbol{d}_{3,i}$ is almost parallel to the segment tangent, meaning this rigid approximation to the position of the binding

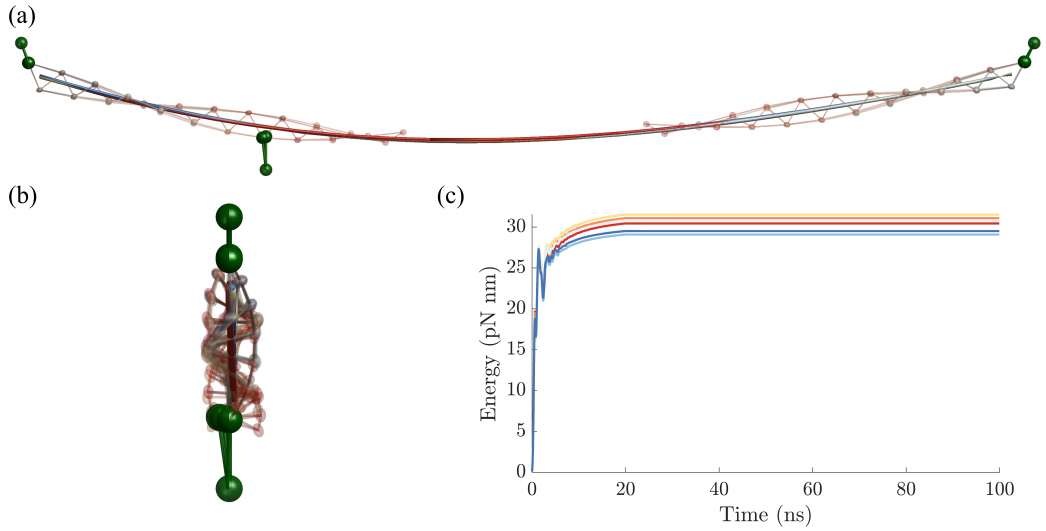


Figure 4.4: (a) *Multi-resolution actin filament attached to* 3 *planar external linkers. We overlay the results of* 6 *simulations, with the Cosserat model used to resolve the filament for between* 30–130 nm *in the middle section, with the remainder resolved using the Ellipsoid model at both ends of the filament. The colours vary through red-yellow-blue, with red almost entirely monomer-based and blue almost entirely rod-based. Filaments are initialised at rest in a straight configuration and allowed to reach equilibrium under the external forces.*
(b) *View along the filament from the right of panel* (a).
(c) *The total filament energy* (*in* pN nm) *varying with time* (*in* ns) *as the system approaches the equilibrium.*

site will hold even for long Cosserat segments. If we have a force $\boldsymbol{F}_{\text{ext}}$ at $\boldsymbol{r}_{\text{ext}}$ due to the external linkers, this can be translated to the existing filament coordinates by considering how it acts on the midpoint of the Cosserat segment

$$\boldsymbol{F}_i = \boldsymbol{F}_{i+1} = \frac{1}{2}\boldsymbol{F}_{\text{ext}}\,, \quad \boldsymbol{C}^i_{\mathcal{L}} = \boldsymbol{Q}_i^T\left(\left(\boldsymbol{r}_{\text{ext}} - \frac{1}{2}(\boldsymbol{r}_{i+1} + \boldsymbol{r}_i)\right) \times \boldsymbol{F}_{\text{ext}}\right)$$

with the external torque $\boldsymbol{C}^i_{\mathcal{L}}$ acting on the orientation basis $\boldsymbol{Q}_i$, enabling twisting of the filament to take place. In Figure 4.4 we see F-actin filaments of length 160 nm with 3 linkers attached from fixed external points. For this case, we place the 3 linkers in the same plane, illustrating a simple bending situation with low filament torsion. We have excellent structural agreement between the models, despite the binding location switching from the Cosserat to the Ellipsoid model as the resolution is varied. We extend this to a more complex example in Figure 4.5, where an F-actin filament of length 210 nm has 4 linkers attached to its from fixed points. In this case, rather than all of these linkers lying in a single plane, we place

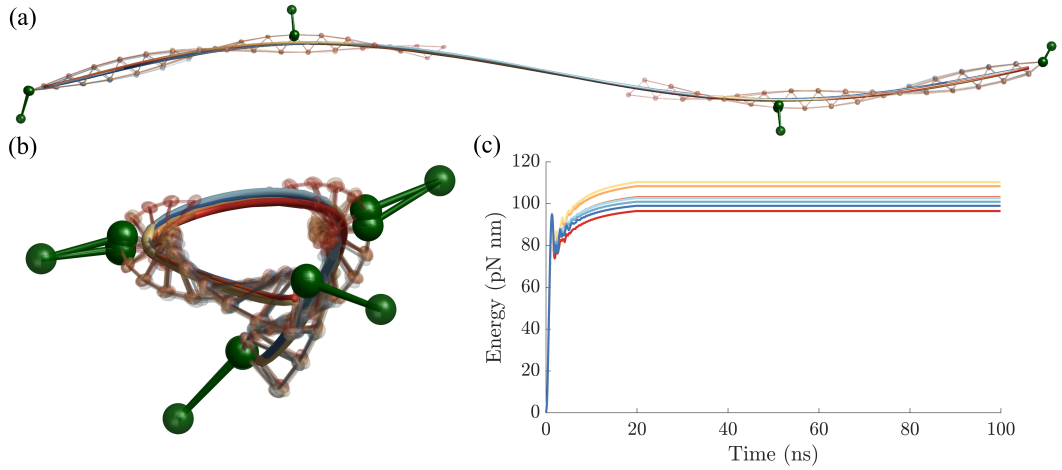


Figure 4.5: (a) *Multi-resolution actin filament attached to* 4 *non-planar external linkers. We overlay the results of* 8 *simulations, with the Cosserat model used to resolve the filament for between* 30–170 nm *in the middle, with the remainder resolved using the Ellipsoid model at both ends. The colours vary through red-yellow-blue, with red almost entirely monomer-based and blue almost entirely rod-based. Filaments are initialised at rest in a straight configuration and allowed to reach equilibrium under the external forces.*
(b) *View along the filament from the right of panel* (a).
(c) *The total filament energy* (*in* pN nm) *varying with time* (*in* ns) *as the system approaches the equilibrium.*

them more randomly in space, inducing larger torques and filament deformation. While agreement across model with differing resolutions is still good, deviations between the equilibrium configurations are more noticeable, which is also borne out in the larger variation in equilibrium filament energy.

4.3.3 Twisted filaments

While the multi-resolution model performs well under linear forces which cause it to bend, there is not perfect agreement, which is partly due to the double helical structure of actin. When the actin double helix is wound more tightly this is known as over-twisting, while when it is unravelled in the opposite direction it is under-twisted; it is easier to under-twist than over-twist the filament. In addition to variable twisting rigidity, the Ellipsoid model also exhibits twist-stretch coupling behaviour; when the filament is over-twisted it contracts more than when under-twisted. This was discussed in detail in Section 3.3.3, and here we will study the differences in the models under torsion, testing whether the extensions made to

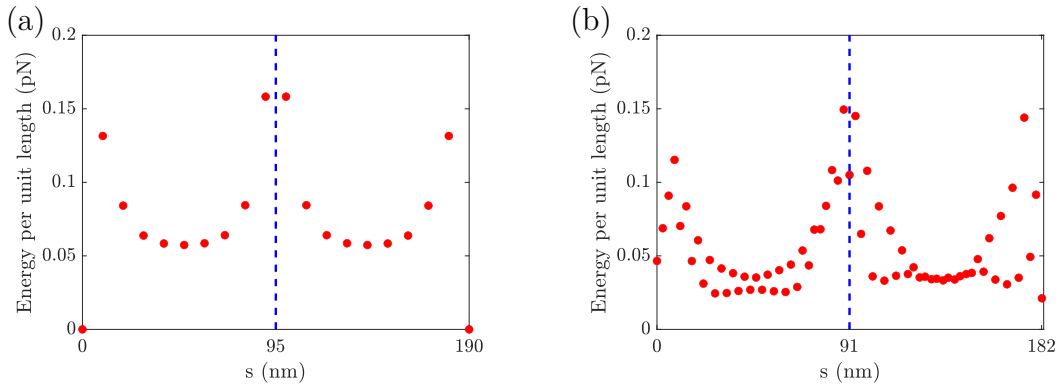


Figure 4.6: *The spatial distribution of potential energy under torsion for:* (a) *the Cosserat model with* 19 *segments; and* (b) *the Ellipsoid model with* 67 *ellipsoids. In both cases we clamped the ends of a filament and attach an external linker to the middle* (*blue dashed line*). *Filaments are initialised at rest in a straight configuration and allowed to reach equilibrium under the external forces. In the ellipsoid case the linker is attached to a monomer centre, while in the Cosserat model it is attached with the same radial distance from the centre-line, inducing an equivalent torque. The red markers indicate the energy per unit length.*

the Cosserat model based on simple twist measurements from the Ellipsoid model are robust when the filament also experiences bending deformations.

We start by highlighting the differences between the models by inducing a torque and small bending deformation in filaments resolved entirely using the standard Cosserat model and Ellipsoid model. We clamp both ends of $\sim 180\,\mathrm{nm}$ long filaments and attach external linker with zero length and spring constant $100\,\mathrm{pN\,nm^{-1}}$ to the middle, away from the centre-line so that a torque is created, and measure the spatial distribution of potential energy. In Figure 4.6 we see the comparison of the energy per unit length of the Ellipsoid and Cosserat filaments. We see that the spatial distribution of energy for the Cosserat model is symmetric on either side of location the central force is applied. However, there is a difference on either side of the binding site for the Ellipsoid model. This is due to the fact that one side is over-twisted (right side in second panel) with roughly 10% higher total energy than the other side which is under-twisted (left side in second panel).

To illustrate this more clearly, we construct an example case where we hold the ends of the filament and move them together to compress the filament, causing it to buckle once the force becomes too great. At the same time, the filament ends are

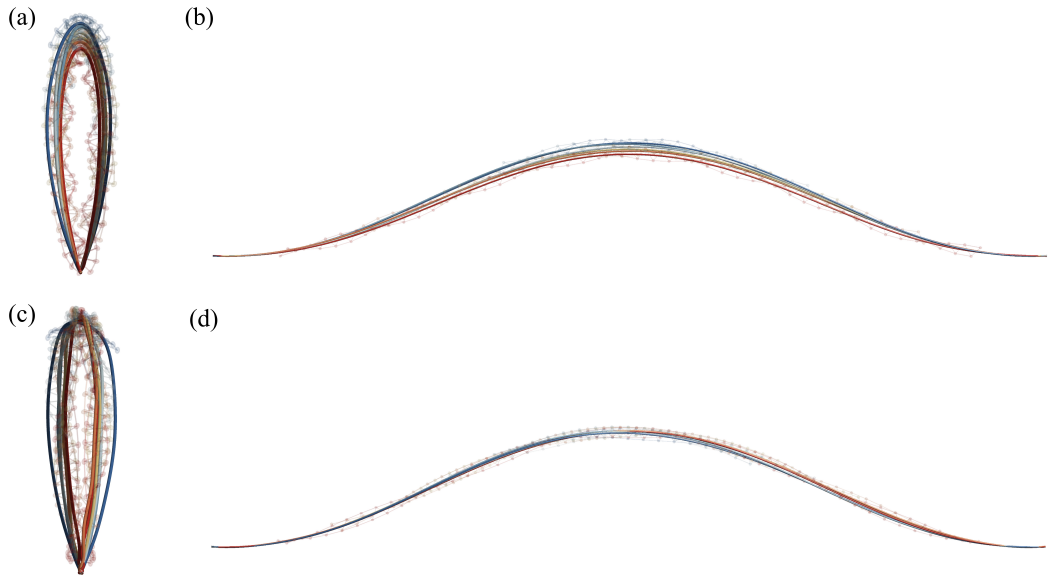


Figure 4.7: *Multi-resolution actin filaments under compression and twist. Each* 300 *nm long filament is twisted* 180° *at each end resulting in a full rotation, and compressed by* 7.5*nm at each end, giving a* 5% *reduction in length which causes buckling. We overlay the results of* 7 *simulations, with the Cosserat model used to resolve the filament for between* 40–240 nm *split evenly between both ends, with the remainder in the centre resolved using the Ellipsoid model. The colours vary through red-yellow-blue, with red almost entirely monomer-based and blue almost entirely rod-based. Filaments are initialised at rest in a straight configuration and allowed to reach equilibrium under the external forces.*
(a) and (b) *Over-twisted from end and side respectively.*
(c) and (d) *Under-twisted from end and side respectively.*

twisted in opposite directions along its length, resulting in helical buckling. As well as being a good setting to establish the torsional behaviour of the multi-resolution model, this has been observed in actin inside filopodia [181]. In Figure 4.7 we see a 300 nm filament under compression and torsion, with a 5% reduction in length and a cumulative 360° twist. In Figure 4.7(a) and (b) we over-twist the filament, meaning the double helical ellipsoidal section of F-actin becomes more tight. We see that the vertical height of the buckled filament is larger in the lower resolution filaments (results visualised in blue). In Figure 4.7(c) and (d) the filaments are set up in an identical manner to the previous case, but with the direction of twist reversed, under-twisting the filament. We again see the lower resolution filaments displaying larger deformations.

To account for these twisting anisotropies in the Ellipsoid model, we now apply

two amendments to the standard Cosserat model that we introduced in Section 3.3: (i) having curvature dependent twisting rigidity, and (ii) incorporating a twist-stretch coupling through scale factor. For the twisting rigidity, we use equation (3.4) and parameters set in equation (3.5) to calculate $B_3(\hat{\kappa}_{\mathcal{L},3})$. For the twist-stretch coupling, we use equation (3.6) and parameters set in equation (3.8) to calculate scale factor $\xi(\hat{\kappa}_{\mathcal{L},3})$, then use equation (3.7) to scale the Cosserat model segment extension values e_i. In Figure 4.8 we set up simulations in almost identical manner to those seen in Figure 4.7, testing the multi-resolution model with extended Cosserat model when over-twisted and under-twisted over a range of twist magnitudes. We see that the extended Cosserat model improves correspondence for the over-twist case across resolutions and over a range of different twist curvatures. However, for the under-twist case, while we observe slight improvement, at larger twists there is still low agreement between models of differing resolution.

4.4 Discussion

At the beginning of this chapter, we discussed how to solve the equations of motion for our dynamic multi-resolution model of F-actin. In Section 4.2, we were able to roughly estimate the timestep size required for stability and accuracy. This was large that than seen in MD simulations, but could still be considered restrictive if we look to scale this work to study longer time scales. A possible direction for future work would be to reformulate the multi-resolution model so it can be solved in a variational manner using energy minimisation [3]. While this would no longer be a dynamic model, we would be able to carry out investigations over much larger time and length spans, involving many actin filaments [107, 108, 182, 183]. As the coupling is defined in a manner consistent with the Cosserat model, the implementation can be extended to it without much difficulty. However, for the Ellipsoid model, the large number of harmonic bonds would adversely affect the efficiency of energy minimisation. One option would be to replace each set of harmonic interactions between a pair of monomers with a single, Cosserat style material interaction. This would allow us to maintain the monomer-based

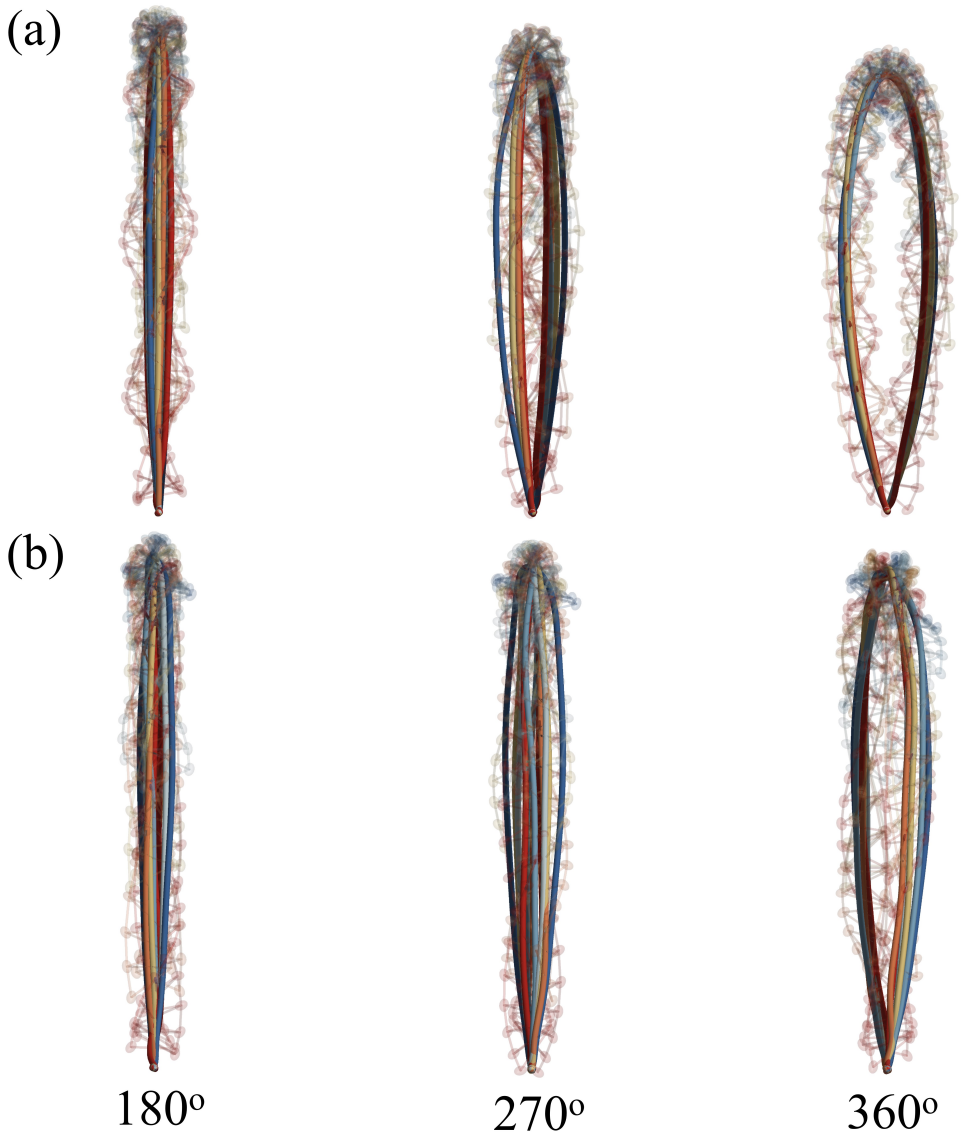


Figure 4.8: *Multi-resolution actin filaments under compression and twist, with variable twisting rigidity* $B_3(\kappa_{\mathcal{L},3})$ *and length scale factor* $\xi(\kappa_{\mathcal{L},3})$*. These simulations are initialised in an otherwise identical manner to Figure* 4.7*.*
(a) *over-twisted filaments subjected to* 180°*,* 270°*, and* 360° *twists.*
(b) *under-twisted filaments subjected to* 180°*,* 270°*, and* 360° *twists.*

representation of actin, while essentially using just the Cosserat model energy terms to describe the entirety of the multi-resolution filament. It would be possible to infer parameters for this model directly from the Ellipsoid model. To be more rigorous, we could return to atomistic MD simulation trajectories for F-actin and carry out a new fluctuation matching procedure [119, 122], fitting longitudinal and lateral values for the elements of $\boldsymbol{S}$ and $\boldsymbol{B}$.

We managed to roughly account for solvent effects on the multi-resolution model by using a simple dissipation term, scaled by mass to ensure the correct dynamic properties for the Cosserat and Ellipsoid models, along with a simple Langevin model. In Chapter 2, we also saw how the parameters for the generalised Langevin equation can be estimated from simulation data to give macroscopic solvent effects. By simulating our model in a solvent bath e.g. a Lennard-Jones fluid [10, 11], we could use this methods to determine our stochastic solvent parameters, knowing that if we scaled them for the models as described in Section 4.2 we would expect the dynamic behaviour of the filament to be matched, regardless of resolution. However, because of the design of the models, the Ellipsoid model surface area is non-uniform, with small gaps between monomers. This could make it difficult to match behaviour in this manner or by using hydrodynamic equations based on the surfaces of the models. Formally treating these difference in excluded volume effects between the models is an important direction of future study if the multi-resolution model is to be applied to further biologically relevant scenarios.

In Section 4.3, three main types of example case were presented; cantilever bending, external linker forces, and twisted filaments. The multi-resolution model, with standard Cosserat model used for low resolution regions, performed well under simple bending deformations. This was particularly impressive for planar deformations, though for non-planar deformations we observed slight differences in in the final equilibrium energy across filaments of varying resolutions. However, when placing the filaments under torsion, it was clear that the standard Cosserat model was unable to capture the anisotropies of the Ellipsoid model. To address this, we presented an extended Cosserat model where the twisting rigidity $B_3(\kappa_{\mathcal{L},3})$ was

dependent on the twist curvature $\kappa_{\mathcal{L},3}$. This allowed us to account for the fact that it is easier to under-twist actin than it is to over-twist it. In addition, we acted on the length of the filament via a newly introduced scaling $\xi(\kappa_{\mathcal{L},3})$, again motivated by the monomer-based representation of actin, where twist-stretch behaviour is observed. We presented functions for both $B_3(\kappa_{\mathcal{L},3})$ and $\xi(\kappa_{\mathcal{L},3})$, with their parameters inferred from data collected by placing the Ellipsoid model under twists of varying magnitude. When applying these extensions to the Cosserat formulation in our multi-resolution model we were able to demonstrate improved model correspondence under torsion, even when large portions of the filament had a low resolution.

In the cases where the filament experienced simple bending deformations, the Cosserat model adequately captured this behaviour, with much improved efficiency when compared to the Ellipsoid model. However, for the twist cases we saw that even when the extended Cosserat model was used, the multi-resolution model was not as accurate at low resolutions. This suggests that the required resolution for accuracy is variable, and would benefit from the formulation of an adaptive multi-resolution model. A filament would be initialised entirely using the Cosserat model, with conditions determining where and when to increase the resolution of the filament to improve accuracy. This could take the form of force thresholds or even a time-dependent local curvature threshold, where if a filament is in a deformed state for long enough we trigger the replacement of Cosserat segments with Ellipsoid monomers. The new ellipsoid orientations and positions can be directly calculated relative to an interpolated filament centreline curve passing through the original Cosserat segment endpoints.

5
Summary and conclusions

Over the course of this work we have studied situations in which a multi-resolution approach can be effective when carrying our simulations of biomolecules. In Chapter 1 we gave an overview of MD simulations and how they can be coarse-grained to give more efficient models of target particles. We then discussed previous multi-resolution and multiscale methodologies, before giving an overview of actin modelling in the literature. Chapter 2 focused on a multi-resolution approach to solvent effects, in which we replaced explicit particle interactions with stochastic noise and drag using the Langevin equation and generalised Langevin equation, presenting methods to determine parameters for these models analytically, or by inference from MD simulation data, with a new 2^{nd} order kernel estimation method presented. It is possible to verify the efficacy of these methods by using example cases, for which we calculated analytic solutions. We then moved to a multi-resolution approach to F-actin in Chapter 3, where parts of the filament are resolved using a detailed monomer-based model, with the rest described using a coarser rod-based model. We presented methods to connect the two models of differing resolution at a coupling region, along with a detailed study of how to match the parameters of the models. Extensions to the Cosserat model were presented to improve correspondence with the high resolution Ellipsoid model, accounting for variable twisting rigidity of actin, along with twist-stretch behaviour. We continued

in Chapter 4, describing methods to numerically integrate the equations of motions for the multi-resolution model to ensure the correct dynamics along the length of a filament of varying resolution. We showed the multi-resolution model of actin filaments was robust by testing it in a set of example cases. The model produced accurate results, both in terms of dynamics and final equilibrium spatial configurations, across a range of filament resolutions. By proceeding in this manner, it was possible to reduce the resolution of the filaments, improving simulation efficiency, without adversely affecting the macroscopic filament behaviour.

At the end of each chapter, we outlined possible future studies which could be carried out to extend the work presented in this thesis. The work in Chapter 2 could be extended to more biologically relevant systems; both heat baths and target particle could be described in increased complexity. In these situations, it may be impractical to calculate exact analytic relations between the high and low resolution representations. However, by using the presented numerical methods, the parameters of the generalised Langevin equation can be inferred, resulting in a coarse grain alternative to explicit solvent modelling. For the work in Chapter 3 and Chapter 4 on F-actin, there are several directions the work may lead. The extensions to the Cosserat model enables F-actin filaments with non-trivial twisting anisotropies to be modelled. To improve accuracy, we can further generalise the Cosserat model by letting all rigidities be variable in curvature, and carry out a comprehensive parameters search based on data collected from the monomer-based representation of F-actin. We could also further extend our methodology around parameter inference. In this work we have been primarily focused on demonstrating consistent behaviour of the models of varying resolution with matched or coupled parameters, and further validation against and inference from experimental data (e.g. parameter values changing based on external solvent conditions or binding proteins like cofilin) could widen the biological scope of the framework. There is also no need to restrict studies to actin, with a general Cosserat model able to describe filopodia, microtubules and other rod-like structures. As these themselves are often formed from actin filaments, there is scope to construct an entirely Cosserat based multi-resolution model. For

our current version of the multi-resolution model of actin, we can begin to explore more complex systems of multiple actin filaments, by using a dynamic approach as described in this work, or using an energy minimisation approach to improve efficiency. There is even scope to combine the methodology used in Chapters 2–4 to create a fully multi-resolution model of F-actin, with both the solvent and filament resolved in varying levels of detail across the simulation domain. Some of the methodological results of this work have been implemented in MEDYAN, a software package for efficient and scalable mechanochemical modelling of active matter networks, developed by my collaborators at the University of Maryland [2, 3, 107]. This will ensure there is a long-lasting impact on the field as a result of this work.

Appendices

A

Analytic Solution for Hamiltonian System in Chapter 2

We consider some further analytic results for the long-range model, to be used for comparison and validation in Section 2.2.1. To simplify our investigations, we will restrict ourselves to a single particle case, rather than the dimer considered in the previous section, with almost identical equations of motion for the particle and oscillators

$$\begin{aligned} M\frac{\mathrm{d}\mathbf{V}}{\mathrm{d}t} &= \sum_{j=1}^{N} k_j\,\alpha_j\Big(\mathbf{x}^j - \alpha_j\mathbf{X}\Big), \\ \frac{\mathrm{d}\mathbf{x}_i^j}{\mathrm{d}t} &= \mathbf{v}^j, \\ m\frac{\mathrm{d}\mathbf{v}^j}{\mathrm{d}t} &= -k_j\left(\mathbf{x}^j - \alpha_j\mathbf{X}\right), \end{aligned} \tag{A.1}$$

this time dropping the subscripts and harmonic bond potential from equation (2.35). As we did for equations (2.39)-(2.40), we can define these equations of motion in a form resembling the generalised Langevin model by solving the equations of the oscillators

$$M\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = -\int_0^t \kappa(\tau)\boldsymbol{V}(t-\tau)\,\mathrm{d}\tau + \boldsymbol{\xi}, \tag{A.2}$$

with friction kernel $\kappa(\tau)$ and noise term $\boldsymbol{\xi} \equiv \boldsymbol{\xi}(t) = [\xi_1, \xi_2, \xi_3]$, derived from the equations of motion for the N oscillators, with form identical to equation (2.41)

and equation (2.42) respectively (dropping the extra indices used for the dimer model) again sampling the initial oscillator particle velocities and positions from their equilibrium distributions. In Section 2.1.3 we saw that to coarse-grain equation (2.53), it is possible to model $\boldsymbol{\xi}(t)$ as a Wiener process scaled by some parameter, so long as the oscillator frequencies ω_i, $i = 1, \ldots, N$ were sampled from an exponential distribution with mean $\overline{\omega}$, and setting the interaction coefficients α_i, $i = 1, \ldots, N$ according to equation (2.43). Under the limit $\overline{\omega} \to \infty$ we showed that $\kappa(\tau) \approx \gamma\delta(\tau)$. For the single particle case, this leads to a simple Langevin model

$$M\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t} = -\gamma\boldsymbol{V} + \boldsymbol{\xi}. \tag{A.3}$$

As was the case for the dimer model, finding analytic solution to equations (2.53)-(2.54) is not straightforward. However, it is in fact possible to exactly solve equation (2.55) by using the integrating factor $e^{\gamma t/M}$ and the analytic expression for $\boldsymbol{\xi}(\tau)$ in equation (2.42), it is possible to derive a set of integrals involving trigonometric functions and exponentials which can be exactly integrated.

$$\begin{aligned}\boldsymbol{V}(t) = \sum_{i=1}^{N} \frac{mM\alpha_i\omega_i^2}{\gamma^2 + \omega_i^2 M^2}\Bigg(&\left[\omega_i\boldsymbol{x}^i(0) + \frac{\boldsymbol{v}^i(0)\gamma}{\omega_i M}\right]\sin(\omega_i t)\\ &+\left[\frac{\gamma}{M}\boldsymbol{x}^i(0) - \boldsymbol{v}^i(0)\right]\cos(\omega_i t) + Ae^{-\gamma t/M}\Bigg),\end{aligned}$$

where A is a constant. Integrating once more gives

$$\begin{aligned}\boldsymbol{X}(t) = \sum_{i=1}^{N} \frac{mM\alpha_i\omega_i^2}{\gamma^2 + \omega_i^2 M^2}\Bigg(&\left[\frac{\gamma}{\omega_i M}\boldsymbol{x}^i(0) - \frac{\boldsymbol{v}^i(0)}{\omega_i}\right]\sin(\omega_i t)\\ &-\left[x^i(0) + \frac{\boldsymbol{v}^i(0)\gamma}{\omega_i^2 M}\right]\cos(\omega_i t)\\ &-\frac{AM}{\gamma}e^{-\gamma t/M} + B\Bigg),\end{aligned}$$

where B is a constant. As $t \to \infty$ the exponential term disappears, mean we eventually tend to periodic solutions (ellipses depending on the coefficients of the sin/cos terms). For consistency with initial conditions $\boldsymbol{X}(0) = \boldsymbol{0}$ and $\boldsymbol{V}(0) = \boldsymbol{0}$,

we can rewrite the above as

$$\begin{aligned}\boldsymbol{V}(t) = \sum_{i=1}^{N} \frac{mM\alpha_i\omega_i^2}{\gamma^2+\omega_i^2M^2}\Bigg(&\left[\omega_i\boldsymbol{x}^i(0) + \frac{\boldsymbol{v}^i(0)\gamma}{\omega_i M}\right]\sin(\omega_i t)\\ &+\left[\frac{\gamma}{M}\boldsymbol{x}^i(0) - \boldsymbol{v}^i(0)\right](\cos(\omega_i t) - e^{-\gamma t/M})\Bigg),\end{aligned} \tag{A.4}$$

$$\begin{aligned}\boldsymbol{X}(t) = \sum_{i=1}^{N} \frac{mM\alpha_i\omega_i^2}{\gamma^2+\omega_i^2M^2}\Bigg(&\left[\frac{\gamma}{\omega_i M}\boldsymbol{x}^i(0) - \frac{\boldsymbol{v}^i(0)}{\omega_i}\right]\sin(\omega_i t)\\ &-\left[\boldsymbol{x}^i(0) + \frac{\boldsymbol{v}^i(0)\gamma}{\omega_i^2 M}\right](\cos(\omega_i t) - 1)\\ &+\left[\boldsymbol{x}^i(0) - \frac{M\boldsymbol{v}^i(0)}{\gamma}\right](e^{-\gamma t/M} - 1)\Bigg).\end{aligned} \tag{A.5}$$

Any difference between the trajectories calculated using equation (A.5)-(A.4) and those from equation (2.53) should arise from the accuracy of the approximation $\kappa(\tau) \approx \gamma\delta(\tau)$. As this analytic solution is only valid when we have a perfect reduction to the Langevin model form, we can use it in Section 2.2.1 to evaluate the accuracy of our approximations. One advantage of having these explicit forms for the position and velocity, rather than simulating trajectories directly from the equations of motion (2.53), is that we can eliminate errors which enter via temporal discretisation or choice of integrator, assuming we are using large values of $\overline{\omega}$ and N.

In Figure A.1 we see example simulation position and velocity trajectory compared against the analytic solution and see good correspondence, though there is a slight divergence in the position. This is expected as the analytic form is still only an approximation to the simulated form, and any errors will be differences will be compounded in time. Figure A.2 show us these local errors do not compromise the long time statistics, with the data from the analytic form matching that of the theoretical velocity autocorrelation function

$$C_m(\tau) = \lim_{t\to\infty}\frac{1}{3}\left\langle \mathbf{V}(t+\tau)\cdot\mathbf{V}(t)\right\rangle, = D\gamma\exp[-\gamma\tau]. \tag{A.6}$$

suggesting that the analytic form has the correct long term statistical properties. A way we can check this is by directly calculating the expected velocity fluctuation at equilibrium, i.e. $\langle \boldsymbol{V}^2\rangle = C_m(0) = D\gamma$. Assuming $t \gg 1$ and taking the

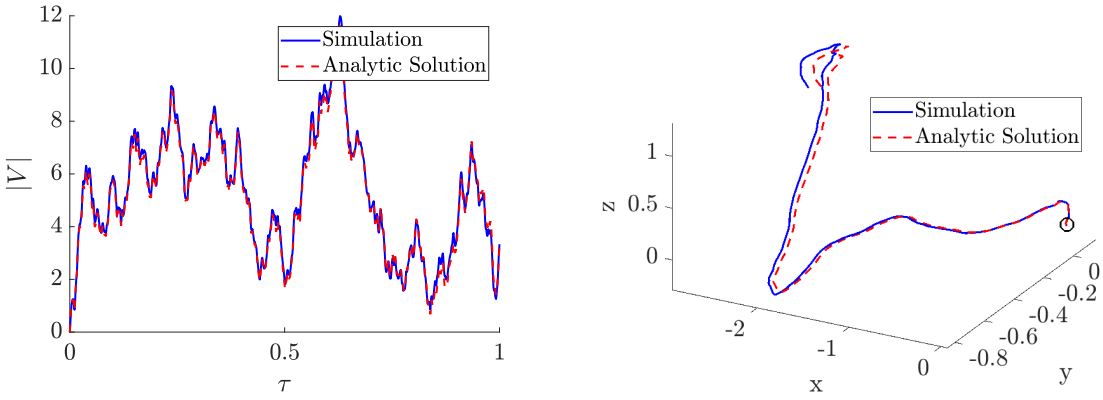


Figure A.1: *Qualitative comparison of position and velocity trajectories for simulated single particle Hamiltonian model equations of motion* (2.53) *(blue solid line) and the analytic solutions in equations* (A.4)-(A.5) *(red dashed line). In the first panel we compare simulated velocity with equation* (A.4)*, and in the second panel the simulated position is compared to equation* (A.5)*. Parameters used for both sets of data were* $\boldsymbol{X}(0), \boldsymbol{V}(0) = \boldsymbol{0}$*,* $\gamma = 10$*,* $D = 1$*,* $M = 1$*,* $m = 10^{-3}$*,* $\overline{\omega} = 100$ *and* $N = 10^5$*.*

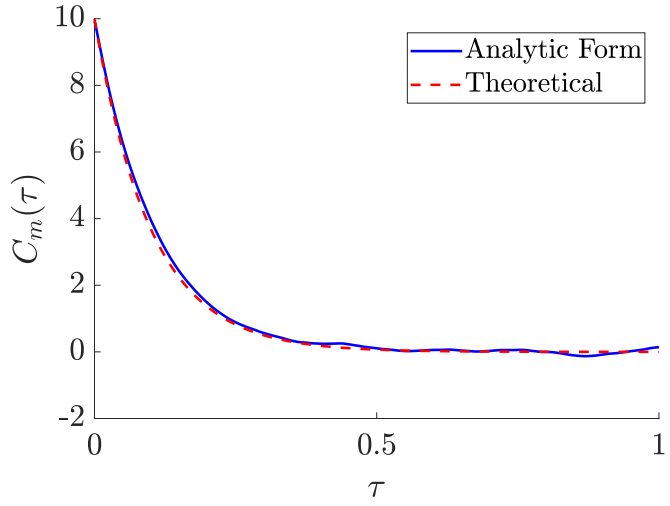


Figure A.2: *The velocity autocorrelation function for the single particle for long-range interactions model (blue solid line) , estimated from long time simulation over dimensionless time of* 10^3 *dimensionless time units. It is compared with the result for the Langevin description of single particle, given by equation* (2.70) *(red dashed line). The parameters used were as used in previous section for dimer models* $\gamma = 10$*,* $D = 1$*,* $M = 1$*,* $m = 10^{-3}$*,* $\overline{\omega} = 10^3$ *and* $N_1 = 10^5$*.*

equilibrium conditions $\langle \boldsymbol{x}^i(0)\boldsymbol{x}^j(0)\rangle = (D\gamma M/m\omega_i)\delta_{ij}$, $\langle \boldsymbol{v}^i(0)\boldsymbol{v}^j(0)\rangle = (D\gamma M/m)\delta_{ij}$, and $\langle \boldsymbol{x}^i(0)\boldsymbol{v}^j(0)\rangle = 0$, we see that

$$\begin{aligned}\langle \boldsymbol{V}(t)^2\rangle &= \sum_{i=1}^{N}\left(\frac{mM\alpha_i\omega_i^2}{\gamma^2+\omega_i^2M^2}\right)^2\left(\frac{D\gamma M}{m}\frac{\gamma^2+\omega_i^2M^2}{\omega_i^2M^2}\right)\\ &= \frac{2D\gamma^2M\overline{\omega}}{N\pi}\sum_{i=1}^{N}\frac{1}{\gamma^2+\omega_i^2M^2}\\ &= \frac{2D\gamma^2}{M\pi}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\overline{\omega}}{\frac{\gamma^2}{M^2}+\omega_i^2}\right).\end{aligned}$$

As N is large, we can approximate the sum as an integral over the distribution for the oscillator frequencies ω, and simplify by substitution and integration by parts

$$\begin{aligned}\langle \boldsymbol{V}(t)^2\rangle &= \frac{2D\gamma^2}{M\pi}\int_0^\infty \frac{e^{-\omega/\overline{\omega}}}{\frac{\gamma^2}{M^2}+\omega^2}\mathrm{d}\omega\\ &= \frac{2D\gamma^2}{M\pi\overline{\omega}}\int_0^\infty \frac{e^{-\omega'}}{\frac{\gamma^2}{\overline{\omega}^2M^2}+\omega^2}\mathrm{d}\omega'\\ &= \frac{2D\gamma}{\pi}\int_0^\infty e^{-\omega'}\arctan\left(\frac{M\overline{\omega}}{\gamma}\omega'\right)\mathrm{d}\omega'\\ &\approx D\gamma\int_0^\infty e^{-\omega'}\mathrm{d}\omega' = D\gamma.\end{aligned}$$

taking the limit $\overline{\omega}\to\infty$ to make the final approximation $\arctan(M\overline{\omega}\omega'/\gamma)\approx \pi/2$ for $\omega'>0$. In practice, this numerically converges reasonably slowly, with $\langle \boldsymbol{V}(t)^2\rangle = 9.68$ for $\overline{\omega}=10^3$, $M=1$, $D=1$, and $\gamma=10$.

B

Alternate Multi-Resolution Coupling Methods for Chapter 3

In this appendix we consider options to constrain the two models at a multi-resolution coupling region using just the existing co-ordinates of the Ellipsoid and Cosserat models. In Section B.1 these are simpler than the method presented in Section 3.2, with limitations highlighted, giving justification for the complexity of the Cosserat-based coupling. We also present a constraint algorithm scheme to attach the two models in Section B.2, in which the coupling itself does not have an energy expression. This can be seen as an alternative means of coupling the models requiring no new parameters to be inferred or justified.

B.1 Simpler multi-resolution couplings and their limitations

We begin by ensuring that the filament remain close together, with the simplest method of achieving this being to draw harmonic springs between the monomer centres and the final segment endpoint. We let $\bar{\boldsymbol{R}}_1$ be the centre of one of the two ellipsoids in the coupling, and $\boldsymbol{r}_\mathrm{n}, \boldsymbol{r}_{\mathrm{n}+1}$ be the endpoints of the final Cosserat segment.

$$\boldsymbol{l}_\mathrm{n} = \boldsymbol{r}_{\mathrm{n}+1} - \boldsymbol{r}_\mathrm{n}\,, \quad \boldsymbol{l}_\mathrm{MR} = \bar{\boldsymbol{R}}_1 - \boldsymbol{r}_{\mathrm{n}+1}$$

$$E_{\mathrm{r}} = \frac{k_{\mathrm{r}}}{2}(|\hat{\boldsymbol{l}}_{\mathrm{MR}}| - |\boldsymbol{l}_{\mathrm{MR}}|)^2,$$

$$\boldsymbol{F}_{\mathrm{Ell}} = -k_{\mathrm{r}}\boldsymbol{l}_{\mathrm{MR}}\left(1 - \frac{|\hat{\boldsymbol{l}}_{\mathrm{MR}}|}{|\boldsymbol{l}_{\mathrm{MR}}|}\right) \quad , \quad \boldsymbol{F}_{\mathrm{n+1}} = -\boldsymbol{F}_{\mathrm{Ell}}.$$

To set the parameters for these bonds, we would have to proceed in a manner that in consistent with the shear matrix $\boldsymbol{S}$ of the Cosserat model, along with the spring constants of the lateral and longitudinal interactions in the Ellipsoid model. However, as there is an absence of angular constraint, the monomers are free to rotate around the segment endpoint so is insufficient for accurate coupling.

One way to begin to incorporate bending properties for the coupling would be to have a harmonic angle constraint. This would maintain the angle between the segment tangent and the line between the segment endpoint and a monomer.

$$E_\theta = \frac{k_\theta}{2}(\theta_0 - \theta)^2,$$

$$\theta = \arccos\left(\frac{-\boldsymbol{l}_{\mathrm{MR}} \cdot \boldsymbol{l}_{\mathrm{n}}}{|\boldsymbol{l}_{\mathrm{MR}}||\boldsymbol{l}_{\mathrm{n}}|}\right),$$

where θ_0 is the value of θ in the rest configuration. If we differentiate this with respect to $\bar{\boldsymbol{R}}_1$, we find that

$$\frac{\mathrm{d}\theta}{\mathrm{d}\bar{\boldsymbol{R}}_1} = \frac{\boldsymbol{l}_{\mathrm{n}}|\boldsymbol{l}_{\mathrm{MR}}|^2 - \boldsymbol{l}_{\mathrm{MR}}(\boldsymbol{l}_{\mathrm{MR}} \cdot \boldsymbol{l}_{\mathrm{n}})}{|\boldsymbol{l}_{\mathrm{MR}}|^2\sqrt{|\boldsymbol{l}_{\mathrm{MR}}|^2|\boldsymbol{l}_{\mathrm{n}}|^2 - |\boldsymbol{l}_{\mathrm{MR}} \cdot \boldsymbol{l}_{\mathrm{n}}|^2}}.$$

While this expression looks intractable, we can simplify it by noting that $|\boldsymbol{a} \times \boldsymbol{b}|^2 = |\boldsymbol{a}|^2|\boldsymbol{b}|^2 - |\boldsymbol{a} \cdot \boldsymbol{b}|^2$, in addition to considering the vector triple product $\boldsymbol{a} \times (\boldsymbol{a} \times \boldsymbol{b}) = -\boldsymbol{b}|\boldsymbol{a}|^2 + \boldsymbol{a}(\boldsymbol{a} \cdot \boldsymbol{b})$ to reach

$$\frac{\mathrm{d}\theta}{\mathrm{d}\bar{\boldsymbol{R}}_1} = \frac{-\boldsymbol{l}_{\mathrm{MR}} \times (\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{l}_{\mathrm{n}})}{|\boldsymbol{l}_{\mathrm{MR}}|^2|\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{l}_{\mathrm{n}}|} = \frac{\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{n}_\theta}{|\boldsymbol{l}_{\mathrm{MR}}|^2},$$

$$\boldsymbol{n}_\theta = \frac{-\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{l}_{\mathrm{n}}}{|\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{l}_{\mathrm{n}}|},$$

where $\boldsymbol{n}_\theta$ is the unit normal to the plane of the angle. By following the same reasoning for $\boldsymbol{r}_{\mathrm{n}}$ and $\boldsymbol{r}_{\mathrm{n+1}}$ we can now write expressions for the forces for this constraint

$$\boldsymbol{F}_{\mathrm{Ell}} = k_\theta\left(\theta - \theta^0\right)\frac{\mathrm{d}\theta}{\mathrm{d}\bar{\boldsymbol{R}}_1},$$

$$\boldsymbol{F}_{\mathrm{n}} = k_\theta\left(\theta - \theta^0\right)\frac{\mathrm{d}\theta}{\mathrm{d}\boldsymbol{r}_{\mathrm{n}}},$$

$$\boldsymbol{F}_{\mathrm{n+1}} = -(\boldsymbol{F}_{\mathrm{Ell}} + \boldsymbol{F}_{\mathrm{n}}).$$

These forces only act on the segment endpoints and ellipsoid centre, so it is also desirable to calculate the torque associated with these forces so that the orientation bases can be updated in a consistent manner

$$\boldsymbol{\tau}_{\mathrm{Ell}} = \boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{F}_{\mathrm{Ell}},$$

$$\boldsymbol{\tau}_{\mathrm{n}} = -\boldsymbol{l}_{\mathrm{n}} \times \boldsymbol{F}_{\mathrm{n}}.$$

These constraints should have spring constants that are consistent with the bending matrix $\boldsymbol{B}$ for the Cosserat model. However, by only acting on points which are on the segment tangent, we lack the ability to describe angular velocities and torques related to filament twisting. To achieve this we must also consider forces that are dependent on the orientation of the basis matrices.

We can try maintaining the position of the centres of monomers relative to a potentially rotating Cosserat filament. We attempt this by introducing a further angle constraint defined around the Cosserat tangent, and by defining a normal to this constructed using the orientation basis of the Cosserat model

$$\boldsymbol{Q}_{\mathrm{Cos}} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3\},$$

$$\boldsymbol{t} = \frac{\boldsymbol{l}_{\mathrm{n}}}{|\boldsymbol{l}_{\mathrm{n}}|} \approx \boldsymbol{d}_1,$$

$$\boldsymbol{d}_3' = \frac{\boldsymbol{t} \times \boldsymbol{d}_2}{|\boldsymbol{t} \times \boldsymbol{d}_2|} \approx \boldsymbol{d}_3.$$

We can then define the angle constraint by projecting into the plane perpendicular to the segment tangent using the vector rejection

$$E_\phi = \frac{k_\phi}{2}(\phi_0 - \phi)^2,$$

$$\phi = \arccos\left(\frac{(\boldsymbol{l}_{\mathrm{MR}} - (\boldsymbol{l}_{\mathrm{MR}} \cdot \boldsymbol{t})\boldsymbol{t}) \cdot \boldsymbol{d}_3'}{|\boldsymbol{l}_{\mathrm{MR}} - (\boldsymbol{l}_{\mathrm{MR}} \cdot \boldsymbol{t})\boldsymbol{t}||\boldsymbol{d}_3'|}\right),$$

where ϕ_0 is the value of ϕ in the rest configuration. Even if we assume that our constructed basis vector $\boldsymbol{d}_3'$ and tangent $\boldsymbol{t}$ are static, calculating the angle force

by considering the derivative of ϕ will not have a simple closed form. However, using our expression for θ from earlier, we see that

$$|\boldsymbol{l}_{\mathrm{MR}} - (\boldsymbol{l}_{\mathrm{MR}} \cdot \boldsymbol{t})\boldsymbol{t}| = \sqrt{|\boldsymbol{l}_{\mathrm{MR}}|^2 - |\boldsymbol{l}_{\mathrm{MR}} \cdot \boldsymbol{t}|^2},$$

$$= \sqrt{|\boldsymbol{l}_{\mathrm{MR}}|^2 - (|\boldsymbol{l}_{\mathrm{MR}}| \cos\theta)^2},$$

$$= |\boldsymbol{l}_{\mathrm{MR}}| \sin\theta.$$

It is now possible to simplify the our expression for ϕ, also remembering that $\boldsymbol{d}'_3$ is a unit vector and $\boldsymbol{t} \cdot \boldsymbol{d}'_3 = 0$

$$\phi = \arccos\left(\frac{\boldsymbol{l}_{\mathrm{MR}} \cdot \boldsymbol{d}'_3}{|\boldsymbol{l}_{\mathrm{MR}}| \sin\theta}\right).$$

Now when we differentiate this expression we can find a simpler expression for the force at $\bar{\boldsymbol{R}}_1$

$$\frac{\mathrm{d}\phi}{\mathrm{d}\bar{\boldsymbol{R}}_1} = \frac{\boldsymbol{l}_{\mathrm{MR}} \times (\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{d}'_3)}{|\boldsymbol{l}_{\mathrm{MR}}|^2 \sqrt{|\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{d}'_3|^2 - |\boldsymbol{l}_{\mathrm{MR}}|^2 \cos^2\theta}},$$

$$= \frac{\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{n}_\phi}{|\boldsymbol{l}_{\mathrm{MR}}|^2 \sqrt{1 - \frac{|\boldsymbol{l}_{\mathrm{MR}}|^2 \cos^2\theta}{|\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{d}'_3|^2}}},$$

$$\boldsymbol{n}_\phi = \frac{\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{d}'_3}{|\boldsymbol{l}_{\mathrm{MR}} \times \boldsymbol{d}'_3|}.$$

Where $\boldsymbol{n}_\phi$ is the normal to the plane of motion. This is problematic as even though we may constrain our angle ϕ correctly, the resulting force is not perpendicular to the respective force to constrain θ, potentially leading to some redundancy and difficulty selecting parameters. Moreover, this still does not allow us to capture the effect rotations of the individual monomers have on the segment as we only constrain the positions of their centres relative to the segment.

Instead, using the same angles as in our previous constraints, we define a force based on the deviation of the line between the segment endpoint and the monomer. The line will be in the plane normal to the vector

$$\boldsymbol{n} = \exp(\phi_0, \boldsymbol{t})\boldsymbol{d}'_3,$$

where using the Rodrigues formula [164] for matrix exponentials (see Section 4.1.2), we can express this normal in terms of the segment tangent and the second orientation basis vector

$$\boldsymbol{n} = \cos\phi_0 \left(\frac{\boldsymbol{t} \times \boldsymbol{d}_2}{|\boldsymbol{t} \times \boldsymbol{d}_2|} \right) - \sin\phi_0 \left(\frac{\boldsymbol{d}_2 - (\boldsymbol{t} \cdot \boldsymbol{d}_2)\boldsymbol{t}}{|\boldsymbol{t} \times \boldsymbol{d}_2|} \right),$$

where $\boldsymbol{d}_2 - (\boldsymbol{t} \cdot \boldsymbol{d}_2)\boldsymbol{t}$ is the vector rejection of $\boldsymbol{d}_2$ along $\boldsymbol{t}$. This means that our definition is axially dependent, but as the shear strains for actin are relatively high errors introduced should be small. Having chosen our plane of movement, we can define the constraint

$$E_S = \frac{S}{2} |(\boldsymbol{l}_{\mathrm{MR}}) - |\hat{\boldsymbol{l}}_{\mathrm{MR}}|(\boldsymbol{t}\cos\theta_0 + \boldsymbol{n} \times \boldsymbol{t}\sin\theta_0))|^2,$$

$$\boldsymbol{F}_{S;\mathrm{n+1}} \approx S((\boldsymbol{l}_{\mathrm{MR}}) - |\hat{\boldsymbol{l}}_{\mathrm{MR}}|(\boldsymbol{t}\cos\theta_0 + \boldsymbol{n} \times \boldsymbol{t}\sin\theta_0))), \quad \boldsymbol{F}_{S;\mathrm{Ell}} = -\boldsymbol{F}_{S;\mathrm{n+1}},$$

where the force derived is an approximation, with $\boldsymbol{t}$ and $\boldsymbol{n}$ are assumed to be roughly constant. It is possible to replace the spring constant with a matrix similar to $\boldsymbol{S}$ used to account for shears in the the Cosserat model

$$\boldsymbol{F}_{S;\mathrm{n+1}} = \boldsymbol{Q}_{\mathrm{Ell}}^T \boldsymbol{S} \boldsymbol{Q}_{\mathrm{Ell}}((\boldsymbol{l}_{\mathrm{MR}}) - |\hat{\boldsymbol{l}}_{\mathrm{MR}}|(\boldsymbol{t}\cos\theta_0 + \boldsymbol{n} \times \boldsymbol{t}\sin\theta_0))), \quad \boldsymbol{F}_{S;\mathrm{Ell}} = -\boldsymbol{F}_{S;\mathrm{n+1}}$$

where we have transformed the matrix into the frame of the ellipsoid, with the assumption being that the filament will be roughly orientated in this direction between $\boldsymbol{r}_{\mathrm{n+1}}$ and $\bar{\boldsymbol{R}}_1$, so the shears and stretches will experience forces in this manner. However, further angular constraints would also need to be formulated between $\boldsymbol{Q}_{\mathrm{Cos}}$ and $\boldsymbol{Q}_{\mathrm{Ell}}$ to maintain the orientation of the ellipsoid monomers.

Given the above it seems that, even when attempting to create a minimal coupling, we still end up with a complex formulation to connect the models. On top of this, the above proposed couplings introduce redundancies and difficulties in parametrisation, along with axial anisotropies. Rather than proceeding in this ad hoc manner, we conclude that it our approach in Section 3.2 offers a better balance of complexity, accuracy and straightforward parametrisation.

B.2 Constraint algorithm coupling methods

Another approach to couple the two regions of differing resolution is to use constraint algorithms [184]. We do not make use of these methods in Chapter 4 when presenting results, but they are useful if two models need to be rigidly coupled in space without requiring new energy terms or parameters to infer, and are presented here for completeness. We can connect the models using a system of holonomic rigid constraints and enforce these at each timestep using an iterative Lagrange multiplier scheme. This will mean that the coupling region is rigid, with no potential energy. We can even set up the models such that they overlap, e.g. two monomers could be embedded in the final Cosserat segment.

B.2.1 SHAKE algorithm

To begin, we shall describe an example with simple distance constraints on a set of particles with spatial positions $\boldsymbol{x}_i$. We define the α^{th} constraint as follows

$$\sigma_\alpha(t) = |\boldsymbol{x}_{\alpha i}(t) - \boldsymbol{x}_{\alpha j}(t)|^2 - d_\alpha^2 = 0,$$

where $\boldsymbol{x}_{\alpha i}(t), \boldsymbol{x}_{\alpha j}(t)$ are the positions of the particles constrained to stay a distance of d_α apart. Assuming the particles are governed by Newton's laws, we can write the following equations of motion

$$m_i \frac{\partial^2 \boldsymbol{x}_i}{\partial t^2} = -\frac{\partial U(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_i},$$

where $U(\boldsymbol{x}_j)$ is the potential energy of the system. If we numerically integrate these equations we can find the unconstrained coordinates of the particles at the next timestep $\boldsymbol{x}_i(t) \rightarrow \boldsymbol{x}_i'(t + \Delta t)$. However, by making this step, our constraints may no longer hold, i.e $\sigma_\alpha(t) \neq 0$ for some α. To address this, we instead write the equations of motion as

$$m_i \frac{\partial^2 \boldsymbol{x}_i}{\partial t^2} = -\frac{\partial}{\partial \boldsymbol{x}_i}\left(U(\boldsymbol{x}_j) + \sum_{\alpha=1}^{n} \lambda_\alpha \sigma_\alpha(t))\right),$$

with the first term on the RHS is the force arising as a result of the original potential, and the second being forces to enforce the constraints, where λ_α are

the Lagrange multipliers which must be calculated. The constrained positions for the new timestep can be written as

$$\boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}'_i(t+\Delta t) + \frac{1}{m_i}\sum_{\alpha=1}^{n}\lambda_\alpha \frac{\partial \sigma_\alpha(t)}{\partial \boldsymbol{x}_i}\left(\Delta t\right)^2. \tag{B.1}$$

Now, using the above, we can write the constraints that must be satisfied at $t+\Delta t$ in terms of the unconstrained coordinated (which are calculable from the original equations of motion) and the Lagrange multipliers.

$$\begin{aligned}\sigma_\beta(t+\Delta t) &= |\boldsymbol{x}_{\beta i}(t+\Delta t) - \boldsymbol{x}_{\beta j}(t+\Delta t)|^2 - d_\beta^2 = 0,\\ &= \Bigg| \boldsymbol{x}'_{\beta i}(t+\Delta t) - \boldsymbol{x}'_{\beta j}(t+\Delta t)\\ &\qquad + \sum_{\alpha=1}^{n}\lambda_\alpha\left(\Delta t\right)^2\left[\frac{1}{m_{\beta i}}\frac{\partial\sigma_\alpha(t)}{\partial \boldsymbol{x}_{\beta i}} - \frac{1}{m_{\beta j}}\frac{\partial\sigma_\alpha(t)}{\partial \boldsymbol{x}_{\beta j}}\right]\Bigg|^2 - d_\beta^2 = 0.\end{aligned}$$

We have now constructed a set of equations in λ_α that can be solved using a Newton-Raphson scheme. To do this, we construct a Jacobian matrix, which can be done reasonably easily for simple linear constraints

$$\boldsymbol{J}_{\alpha\beta} = \left.\frac{\partial\sigma_\beta}{\partial\lambda_\alpha}\right|_{\boldsymbol{\lambda}=0} = 2(\boldsymbol{x}'_{\beta i}(t+\Delta t) - \boldsymbol{x}'_{\beta j}(t+\Delta t))\left[\frac{1}{m_{\beta i}}\frac{\partial\sigma_\alpha(t)}{\partial \boldsymbol{x}_{\beta i}} - \frac{1}{m_{\beta j}}\frac{\partial\sigma_\alpha(t)}{\partial \boldsymbol{x}_{\beta j}}\right]\left(\Delta t\right)^2.$$

This allows us to find the diagonal and off-diagonal elements of the Jacobian

$$\boldsymbol{J}_{\alpha\alpha} = 4(\boldsymbol{x}'_{\alpha i}(t+\Delta t) - \boldsymbol{x}'_{\alpha j}(t+\Delta t))\cdot(\boldsymbol{x}_{\alpha i}(t) - \boldsymbol{x}_{\alpha j}(t))\left[\frac{1}{m_{\alpha i}} + \frac{1}{m_{\alpha j}}\right]\left(\Delta t\right)^2,$$

$$\boldsymbol{J}_{\alpha\beta} = \begin{cases}\frac{4}{m_i}(\boldsymbol{x}'_i(t+\Delta t) - \boldsymbol{x}'_{\beta j}(t+\Delta t))\cdot(\boldsymbol{x}_i(t) - \boldsymbol{x}_{\alpha j}(t))\left(\Delta t\right)^2, & \text{if } \alpha i = \beta i = i\\ 0, & \text{otherwise.}\end{cases}$$

It is now possible to enforce the constraints, by solving (e.g. by LU decomposition)

$$\boldsymbol{\lambda} = -\boldsymbol{J}^{-1}\boldsymbol{\sigma},$$

and using these values of λ_α in Equation (B.1) we can shift the unconstrained positions. We then reset $\lambda_\alpha = 0$ and repeat the above until the constraints are again satisfied (to some preset tolerance). In this example it is reasonably simple to construct the Jacobian matrix, but this may not be the case for more complex holonomic constraints. This has given rise to many iterative schemes tailored to

specific situations. The most commonly known is the SHAKE algorithm, where it is assumed that the Jacobian is diagonally dominant (i.e. accounting for the off-diagonal terms is not essential for convergence) allowing for faster calculation and inversion at each timestep. For the above case, this would be written as

$$\lambda_\alpha = -\frac{\sigma_\alpha(t)}{\boldsymbol{J}_{\alpha\alpha}},$$

with the resulting force due to constraint α given by

$$\begin{aligned}\boldsymbol{F}_\alpha &= \lambda_\alpha \frac{\partial \sigma_\alpha(t)}{\partial \boldsymbol{x}_i} \\ &= \frac{\mu_{\alpha ij}(|\boldsymbol{x}'_{\alpha i}(t+\Delta t) - \boldsymbol{x}'_{\alpha j}(t+\Delta t)|^2 - d_\alpha^2)}{2\left(\Delta t\right)^2 \left(\boldsymbol{x}'_{\alpha i}(t+\Delta t) - \boldsymbol{x}'_{\alpha j}(t+\Delta t)\right)\cdot\left(\boldsymbol{x}_{\alpha i}(t) - \boldsymbol{x}_{\alpha j}(t)\right)}(\boldsymbol{x}_{\alpha i}(t) - \boldsymbol{x}_{\alpha j}(t)),\end{aligned}$$

$$\mu_{\alpha ij} = \frac{m_{\alpha i} m_{\alpha j}}{m_{\alpha i} + m_{\alpha j}}.$$

B.2.2 RATTLE algorithm

Though the SHAKE algorithm could be used in the coupling region of our actin model, we would only be able to update the positions at each time step and estimate the velocity (which is especially restrictive for the Cosserat model, where the potential depends on the angular velocity). Instead the constraint algorithm can be reformulated to mesh with the velocity Verlet scheme, in a scheme known as RATTLE, where the positions and velocities are linearly constrained [39]. A similar scheme can be constructed for the position Verlet scheme, but this would require 3/4 Lagrange multiplier problems to be solved at each timestep. We know that the discretisation of the velocity Verlet algorithm can be written, for a system of particles $\boldsymbol{x}_i$, as

$$\boldsymbol{v}_i\left(t + \tfrac{1}{2}\Delta t\right) = \boldsymbol{v}_i(t) + \frac{1}{2}\frac{\partial^2 \boldsymbol{x}_i(t)}{\partial t^2}\Delta t,$$

$$\boldsymbol{x}_i\left(t + \Delta t\right) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i\left(t + \tfrac{1}{2}\Delta t\right)\Delta t,$$

$$\boldsymbol{v}_i\left(t + \Delta t\right) = \boldsymbol{v}_i\left(t + \tfrac{1}{2}\Delta t\right) + \frac{1}{2}\frac{\partial^2 \boldsymbol{x}_i(t+\Delta t)}{\partial t^2}\Delta t.$$

This time, we have to consider how to transform the unconstrained values for both particle position and velocity so that the constraints and their time derivative hold

$$\sigma_\alpha(t) = |\boldsymbol{x}_{\alpha i}(t) - \boldsymbol{x}_{\alpha j}(t)|^2 - d_\alpha^2 = 0,$$

$$\dot{\sigma}_\alpha(t) = (\boldsymbol{x}_{\alpha i}(t) - \boldsymbol{x}_{\alpha j}(t)) \cdot (\boldsymbol{v}_{\alpha i}(t) - \boldsymbol{v}_{\alpha j}(t)) = 0.$$

We can write expression for the constrained positions and velocities after the first velocity half timestep and position full timestep

$$\boldsymbol{x}_i(t + \Delta t) = \boldsymbol{x}'_i(t + \Delta t) + \frac{1}{m_i} \sum_{\alpha=1}^{n} \lambda_\alpha^{\boldsymbol{x}} \frac{\partial \sigma_\alpha(t)}{\partial \boldsymbol{x}_i} (\Delta t)^2 ,$$

$$\boldsymbol{v}_i(t + \tfrac{1}{2}\Delta t) = \boldsymbol{v}'_i(t + \tfrac{1}{2}\Delta t) + \frac{1}{m_i} \sum_{\alpha=1}^{n} \lambda_\alpha^{\boldsymbol{x}} \frac{\partial \sigma_\alpha(t)}{\partial \boldsymbol{x}_i} (\Delta t) ,$$

in addition the the final velocity half time step

$$\boldsymbol{v}_i(t + \Delta t) = \boldsymbol{v}'_i(t + \Delta t) + \frac{1}{m_i} \sum_{\alpha=1}^{n} \lambda_\alpha^{\boldsymbol{v}} \frac{\partial \dot{\sigma}_\alpha(t)}{\partial \boldsymbol{v}_i} (\Delta t) ,$$

giving us two sets of Lagrange multipliers, $\lambda_\alpha^{\boldsymbol{x}}$ and $\lambda_\alpha^{\boldsymbol{v}}$, to solve for across each full timestep. It is obvious that the constraints for the position timestep can be satisfied using the same scheme as described in the previous section. The second set of Lagrange multipliers is calculated after the second velocity timestep, so the positions will still be constrained, giving a set of equations to solve for

$$\dot{\sigma}_\beta(t + \Delta t) = (\boldsymbol{x}_{\beta i}(t + \Delta t) - \boldsymbol{x}_{\beta j}(t + \Delta t)) \cdot \Bigg(\boldsymbol{v}'_{\beta i}(t + \Delta t) - \boldsymbol{v}'_{\beta j}(t + \Delta t) + \sum_{\alpha=1}^{n} \lambda_\alpha^{\boldsymbol{v}} (\Delta t) \left[\frac{1}{m_{\beta i}} \frac{\partial \sigma_\alpha(t)}{\partial \boldsymbol{v}_{\beta i}} - \frac{1}{m_{\beta j}} \frac{\partial \sigma_\alpha(t)}{\partial \boldsymbol{v}_{\beta j}} \right] \Bigg).$$

The Jacobian for these constraints is defined as

$$\boldsymbol{J}_{\alpha\beta} = \left. \frac{\partial \sigma_\beta}{\partial \lambda_\alpha} \right|_{\boldsymbol{\lambda}^{\boldsymbol{v}}=0} = (\boldsymbol{x}_{\beta i}(t + \Delta t) - \boldsymbol{x}_{\beta j}(t + \Delta t)) \cdot \left[\frac{1}{m_{\beta i}} \frac{\partial \sigma_\alpha(t)}{\partial \boldsymbol{v}_{\beta i}} - \frac{1}{m_{\beta j}} \frac{\partial \sigma_\alpha(t)}{\partial \boldsymbol{v}_{\beta j}} \right] (\Delta t) .$$

Simplifying allows us to find the diagonal and off-diagonal elements of the Jacobian

$$\boldsymbol{J}_{\alpha\alpha} = (\boldsymbol{x}_{\alpha i}(t+\Delta t) - \boldsymbol{x}_{\alpha j}(t+\Delta t)) \cdot (\boldsymbol{x}_{\alpha i}(t) - \boldsymbol{x}_{\alpha j}(t)) \left[\frac{1}{m_{\alpha i}} + \frac{1}{m_{\alpha j}} \right] (\Delta t) = \frac{d_\alpha^2}{\mu_{\alpha ij}} (\Delta t) ,$$

$$\boldsymbol{J}_{\alpha\beta} = \begin{cases} \frac{1}{m_i} (\boldsymbol{x}_i(t + \Delta t) - \boldsymbol{x}_{\beta j}(t + \Delta t)) \cdot (\boldsymbol{x}_i(t) - \boldsymbol{x}_{\alpha j}(t)) (\Delta t) , & \text{if } \alpha i = \beta i = i \\ 0, & \text{otherwise.} \end{cases}$$

This means that the Jacobian for the velocity constraint step is the same (up to a multiplicative constant) as the one found by the end of the position constraint stage and doesn't need to be recalculated. If we consider only the diagonal elements of the Jacobian, we find the force is of the form

$$\begin{aligned}\boldsymbol{F}_{\alpha}^{\boldsymbol{v}} &= \lambda_{\alpha}\frac{\partial\dot{\sigma}_{\alpha}(t)}{\partial\boldsymbol{v}_i}\\ &= \frac{\mu_{\alpha ij}}{2\,(\Delta t)\,d_{\alpha}^{2}}(\boldsymbol{x}_{\alpha i}(t+\Delta t)-\boldsymbol{x}_{\alpha j}(t+\Delta t))\cdot(\boldsymbol{v}'_{\alpha i}(t+\Delta t)-\boldsymbol{v}'_{\alpha j}(t+\Delta t))\\ &\qquad\times(\boldsymbol{x}_{\alpha i}(t+\Delta t)-\boldsymbol{x}_{\alpha j}(t+\Delta t)),\end{aligned}$$

and by iteratively updating the velocity in the same way as the position it is possible to fully constrain the system for a whole timestep. Using this adapted velocity Verlet algorithm, it is possible to almost completely couple our two choices of actin model using appropriately chosen holonomic constraints, with linear positions and velocities fully accounted for. However, both the Cosserat and Ellipsoid models have equations of motion formulated with angular velocity and orientations of rigid/semi-rigid units, in addition to linear velocity and spatial position. This means we must use more nuanced methods to account for all forces and energies of the system.

B.2.3 Projecting positions and velocities into coupling region

If we can efficiently project the angular and linear velocities, in addition to orientations and positions for both the Cosserat and Ellipsoid models, into virtual coupling coordinates with purely linear velocity and position, we could then run the Lagrange multiplier algorithms to enforce sensible constraints. We can then use these updated coordinates to transform the Cosserat and Ellipsoid model coordinates accordingly. If these mappings are unique and invertible and conserve energy then we would expect our complete model to conserve energy at each timestep.

Each Cosserat segment is defined by two spatial endpoints $\boldsymbol{r}_i$, which each have a linear velocity $\boldsymbol{v}_i$, in addition to an orientation basis $\boldsymbol{Q}$ and angular velocity $\boldsymbol{\omega}_{\mathcal{L}}$. On the other hand, each monomer in the Ellipsoid model has a centre $\bar{\boldsymbol{R}}$, orientation $\boldsymbol{Q}$, and angular velocity $\boldsymbol{\omega}$. We can consider the final two monomers in a chain as

a single rigid body, with their orientations fixed, or as two individual monomers. To fully describe the segment and ellipsoid positions, orientations and velocities we need to map to a minimum of 3 points. A simple choice would be equilateral triangles with point masses at the corners, with the centre of mass lined up with that of the Cosserat segment or ellipsoid monomers. By defining the corners of the triangle as $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$ we can uniquely define a basis as a function of these and their centre of mass (assuming all the points have equal mass)

$$\boldsymbol{x}_{\mathrm{CoM}} = \frac{1}{3}(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3),$$

$$\boldsymbol{d}_1 = \frac{\boldsymbol{x}_3 - \boldsymbol{x}_1}{|\boldsymbol{x}_3 - \boldsymbol{x}_1|},$$

$$\boldsymbol{d}_2 = \frac{\boldsymbol{x}_2 - \boldsymbol{x}_{\mathrm{CoM}}}{|\boldsymbol{x}_2 - \boldsymbol{x}_{\mathrm{CoM}}|},$$

$$\boldsymbol{d}_3 = \boldsymbol{d}_1 \times \boldsymbol{d}_2.$$

The first two vectors are unit vectors by definition, and it is clear that $\boldsymbol{d}_2 \cdot \boldsymbol{d}_3 = 0$ and $\boldsymbol{d}_3 \cdot \boldsymbol{d}_1 = 0$ by the scalar triple product. For the final orthogonality condition we have

$$\begin{aligned}
\boldsymbol{d}_1 \cdot \boldsymbol{d}_2 &= \frac{1}{|\boldsymbol{x}_3 - \boldsymbol{x}_1||\boldsymbol{x}_2 - \boldsymbol{x}_{\mathrm{CoM}}|}((\boldsymbol{x}_3 - \boldsymbol{x}_1) \cdot (\boldsymbol{x}_2 - \boldsymbol{x}_{\mathrm{CoM}})), \\
&\propto (\boldsymbol{x}_3 - \boldsymbol{x}_1) \cdot \left(\boldsymbol{x}_2 - \frac{1}{3}(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3)\right), \\
&= (\boldsymbol{x}_3 - \boldsymbol{x}_1) \cdot \left(\frac{2}{3}(\boldsymbol{x}_2 - \boldsymbol{x}_1) - \frac{1}{3}(\boldsymbol{x}_3 - \boldsymbol{x}_1)\right).
\end{aligned}$$

Using the cosine dot product rule we can simplify things

$$(\boldsymbol{x}_3 - \boldsymbol{x}_1) \cdot (\boldsymbol{x}_2 - \boldsymbol{x}_1) = |\boldsymbol{x}_3 - \boldsymbol{x}_1||\boldsymbol{x}_2 - \boldsymbol{x}_1| \cos\left(\frac{\pi}{3}\right),$$

$$\Rightarrow \boldsymbol{d}_1 \cdot \boldsymbol{d}_2 = \frac{2}{3} \cdot \frac{1}{2}|\boldsymbol{x}_3 - \boldsymbol{x}_1||\boldsymbol{x}_2 - \boldsymbol{x}_1| - \frac{1}{3}|\boldsymbol{x}_3 - \boldsymbol{x}_1|^2 = 0,$$

as the sides of the triangle are all equal and the angle between them is $\frac{\pi}{3}$. All that needs to be shown is that $|\boldsymbol{d}_3| = 1$ for orthonormality

$$\begin{aligned}
|\boldsymbol{d}_1 \times \boldsymbol{d}_2| &= \frac{1}{|\boldsymbol{x}_3 - \boldsymbol{x}_1||\boldsymbol{x}_2 - \boldsymbol{x}_{\mathrm{CoM}}|}(\boldsymbol{x}_3 - \boldsymbol{x}_1) \times (\boldsymbol{x}_2 - \boldsymbol{x}_{\mathrm{CoM}})|, \\
&= \frac{1}{|\boldsymbol{x}_3 - \boldsymbol{x}_1||\boldsymbol{x}_2 - \boldsymbol{x}_{\mathrm{CoM}}|} \left|(\boldsymbol{x}_3 - \boldsymbol{x}_1) \times \left(\frac{2}{3}(\boldsymbol{x}_2 - \boldsymbol{x}_1) - \frac{1}{3}(\boldsymbol{x}_3 - \boldsymbol{x}_1)\right)\right|.
\end{aligned}$$

Making use of the sine rule for triangles and the sine cross product we have

$$\frac{|\boldsymbol{x}_2 - \boldsymbol{x}_{\text{CoM}}|}{\sin\frac{\pi}{6}} = \frac{|\boldsymbol{x}_2 - \boldsymbol{x}_1|}{\sin\frac{2\pi}{3}} \quad , \quad |(\boldsymbol{x}_3 - \boldsymbol{x}_1) \times (\boldsymbol{x}_2 - \boldsymbol{x}_1)| = |\boldsymbol{x}_3 - \boldsymbol{x}_1||\boldsymbol{x}_2 - \boldsymbol{x}_1| \sin\left(\frac{\pi}{3}\right),$$

$$\Rightarrow |\boldsymbol{d}_1 \times \boldsymbol{d}_2| = \frac{2}{3}\frac{\frac{\sqrt{3}}{2}\frac{\sqrt{3}}{2}}{\frac{1}{2}} = 1.$$

We can generalise this method to create bases unique for large sets of points with different masses, provided they are not colinear. Having constructed a relationship between the coupling triangles (or more complex alternative) we must also draw constraints between the points of the different models. As the ellipsoids will already have forces defined between them, we simply need two sets of constraints, on set for each ellipsoid to the Cosserat segment. If we choose the maximum number of constraints, this results in $3 \times 3 = 9$ to maintain the triangle side lengths, and $2 \times 3 \times 3 = 18$ for the constraints between the Ellipsoid and Cosserat models.

Before applying the constraints, we can write the coupling points as

$$\boldsymbol{x}_i = Q(\boldsymbol{x}_i^0 - \boldsymbol{x}_{\text{CoM}}^0) + \boldsymbol{x}_{\text{CoM}}.$$

After applying the constraints, we can update this expression

$$\boldsymbol{x}_i' = Q'(\boldsymbol{x}_i^0 - \boldsymbol{x}_{\text{CoM}}^0) + \boldsymbol{x}_{\text{CoM}}'.$$

From this we can calculate the amendment to the linear and angular velocity as a result of enforcing the position constraints

$$\boldsymbol{v}_{\text{CoM}}' = \boldsymbol{v}_{\text{CoM}} + \frac{\boldsymbol{x}_{\text{CoM}}' - \boldsymbol{x}_{\text{CoM}}}{\Delta t},$$

$$\boldsymbol{\omega}_{\text{CoM}}' = \boldsymbol{\omega}_{\text{CoM}} + \frac{\log\left(\boldsymbol{Q}'\boldsymbol{Q}^T\right)}{\Delta t}.$$

However, we run into complications when considering second Lagrange multiplier step in the RATTLE scheme. The linear velocity constraint tries to ensure the relative velocity of two particles is perpendicular to the constraint vector. However, when angular velocity is also involved enforcing this becomes non-trivial. The constraints that maintain the lengths of the triangles are invariant under angular velocity around their own centre (meaning velocity contribution from angular

momentum satisfies the velocity constraint naturally), but this is not true of constraints between triangles with different centres and angular velocities. As a compromise, we can just apply the constraints to the linear velocities of the centres of mass of each triangle. More complex methods that formally deal with angular constraints, such are QSHAKE [41] and SHAPE [185] algorithms, can also be used to maintain more accurate and efficient constraints.

C

Further Details and Simulation Timings for Chapter 4

C.1 Full energy plots for example cases

Below we see the full energy profile for each of the example cases discussed in Chapter 4. For the Cosserat model these include the shear/stretch and bend/twist potential energy, along with angular and linear kinetic energy. For the Ellipsoid model, we have the harmonic bond potential energy in addition to angular and linear kinetic energy. We also include plots for the stretch-twist energy (not used in example cases) and the value of the dissipation constant used. For all examples, the Cosserat model discrete segment length is set at 10 nm, while the Ellipsoid model rise per actin monomer is ~ 2.76 nm. It is not straightforward to accurately compare filaments of differing resolutions if we have a fixed Cosserat segment discretisation length and set spatial parameters for the Ellipsoid model which are not commensurate. To accommodate this, the coupling region varies in length slightly from case to case, with ellipsoids placed up to the point where they do not overlap with any Cosserat segments. We will employ similar methods for all following examples. Energy is measured in pN nm and time in nanoseconds. As was seen in Chapter 4, plots vary through red-yellow-blue, with red lines for case which are mainly monomer based, and blue lines for mainly rod-based cases.

We see in Figure C.1 that there is very good agreement for the total energy of the system throughout the simulation, despite differing resolution for each case. The Cosserat region varied between 20–140 nm, with ellipsoids added to bring the filament length up to 160 nm in each case. A downward force of $F_{\text{cant}} = 0.35$ pN is applied across the final two ellipsoids, along with a dissipative force with $\gamma = 0.075$. The endpoints and orientation of the first Cosserat segment are kept constant throughout the simulation to create the effect of it being anchored to a wall, meaning the effective filament length is $L_{\text{cant}} = 150$ nm, which is roughly two full periods of helical twist. From the filament potential energy expressions for the Cosserat rod model (top row of Figure C.1) we see that bending is the dominant contributor, as the filaments are relatively stiff, making them easier to bend/twist rather than stretch/shear.

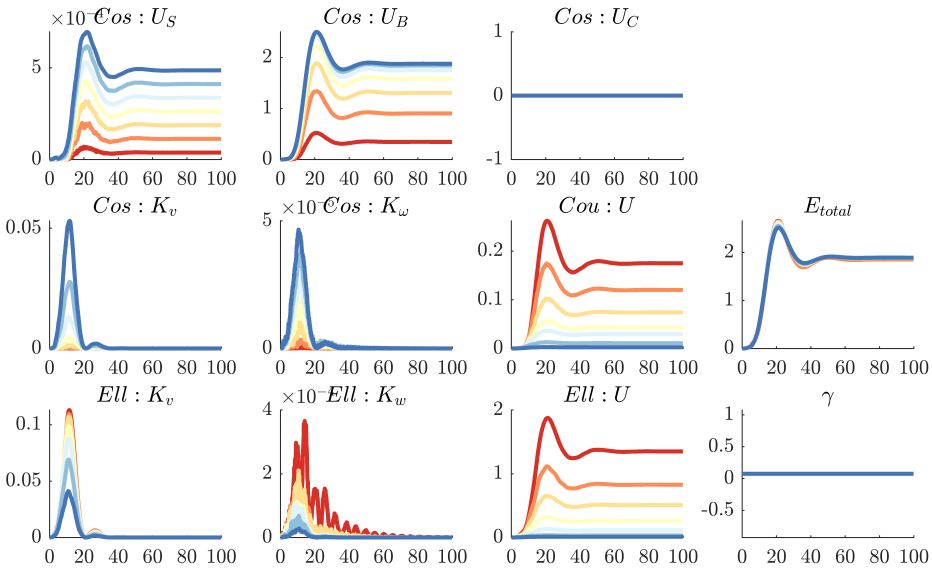


Figure C.1: *Full dynamic energy profile for the cantilever example case.*

For the planar linkers case the Ellipsoid model is used at both ends of the filament, with the Cosserat region in the centre varying between 30–130 nm. These linkers take the form of harmonic bonds with an equilibrium length of 5 nm and a very stiff spring constant $k_{\text{ext}} = 10^2\,\text{pN}\,\text{nm}^{-1}$ chosen, allowing the filament to be pulled into a strained configuration. The filament is also placed under a dissipative force with $\gamma = 0.5$. In Figure C.2 we see the energy agreement is very good, though

with slightly more variation than for the cantilever case. This is most likely down to increased overall forces, along with the slight torsion introduced by the central linker, especially in the case where the coupling is close to the binding site. For

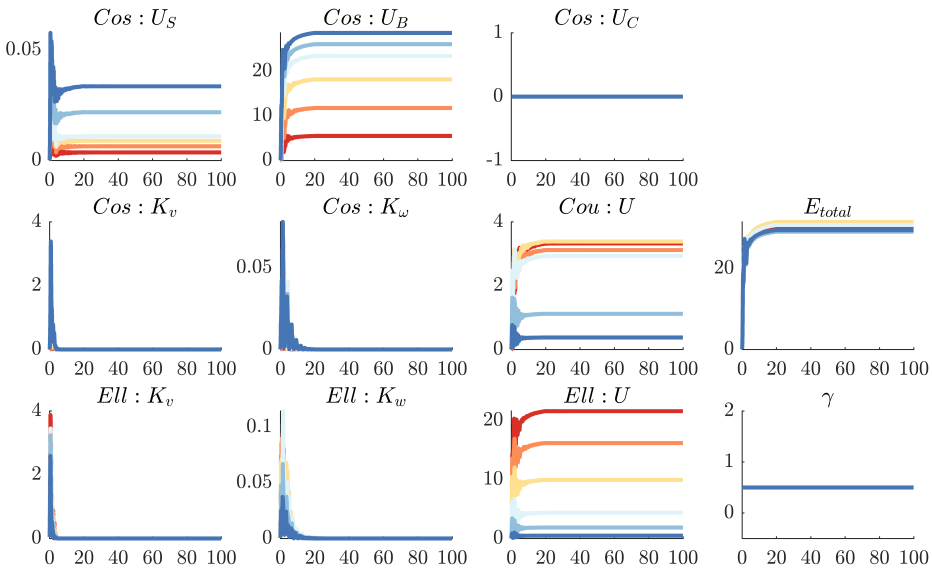


Figure C.2: *Full dynamic energy profile for the planar external linkers example case.*

the non-planar linkers case the Ellipsoid model is used at both ends of the filament, varying the central Cosserat region between 30–170 nm and again resolving the ends with the Ellipsoid model. Again, in Figure C.3 we see the energy agreement is good, though with more variation than for the planar linkers case, with much larger deformation and torsional forces the likely cause of this.

For the twist cases, we see the example where the two Cosserat regions are varied between 20–120 nm, with ellipsoids added from the middle to bring the filament length up to 300 nm in each case. They are then compressed 7.5 nm and twisted 180° at each end in opposite directions to induce an overall full 360° twist and 5% reduction in length We also use the Langevin equation to introduce noise to the simulations, which lowers the time taken for the filament to buckle, with dissipative force set with $\gamma = 0.5$. The final coordinates are rotated to overlap as the filaments may have buckled in different directions, due the symmetry of the system and ensuing sensitivity to noise. We see the energy profiles for the

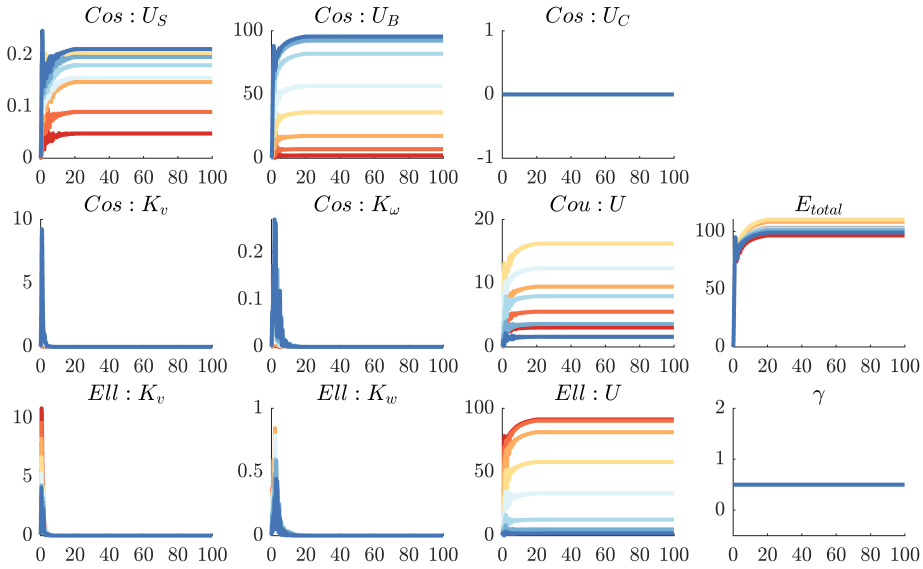


Figure C.3: *Full dynamic energy profile for the non-planar external linkers example case.*

twist case with constant torsional rigidity, and for both the over-twist (Figure C.4) and under-twist (Figure C.5) cases there is a consistent trend, with filaments that have a higher proportion resolved in the Cosserat model finishing with a higher equilibrium energy. Interestingly, for all twist cases we have much poorer dynamic

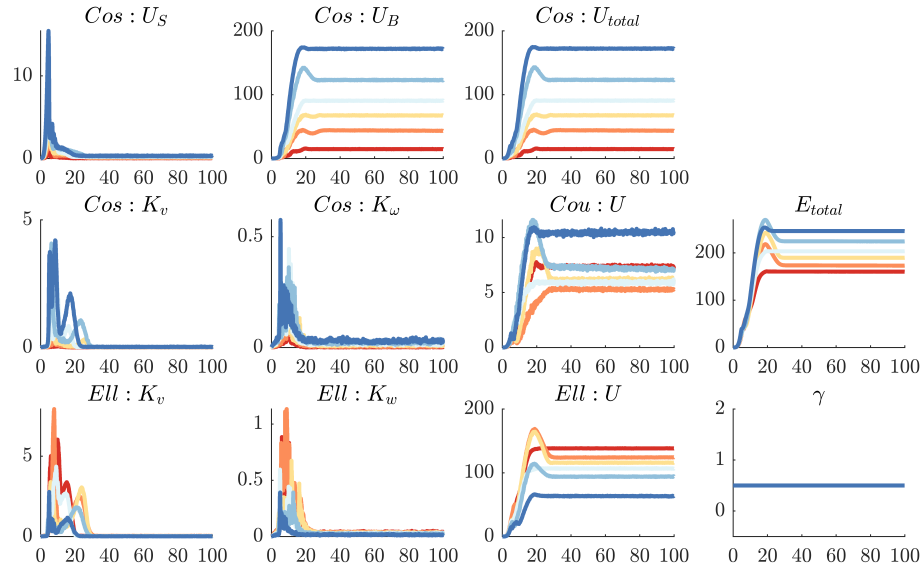


Figure C.4: *Full dynamic energy profile for the over-twist example case.*

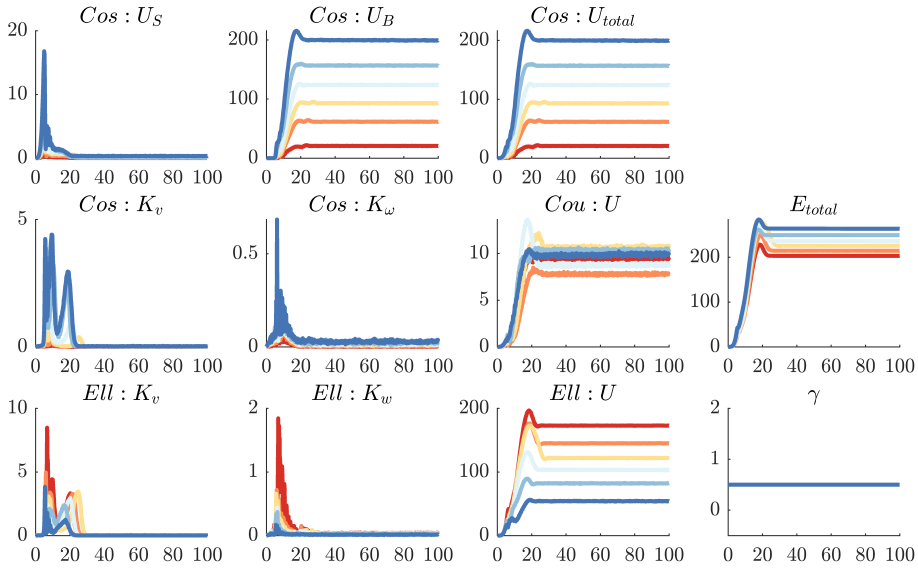


Figure C.5: *Full dynamic energy profile for the under-twist example case.*

agreement, especially when compared to the cantilever case. This suggests that under compression the shear/stretch rigidities play a large role, and future work made be need to refine these choices.

C.2 Simulation timings

Using the above integration methods, we simulate each of the example cases for varying resolutions using code written using MATLAB [180]. In each of these situations it is potentially useful to only resolve part of the filament using the Ellipsoid model as, while it provides more detail, it is more computationally expensive. In Figure C.6, we see the simulation times for each of the example cases in Section 4.3, along with a linear fit. We normalise relative to the shortest simulation time (in this case the cantilever case which is almost entirely rod-based) and filament length to allow for easier comparison. We see that as resolution detail decreases, we observe speed up for all cases, with this most pronounced in the twist case, which could be due to the higher forces involved relative to the external linker and cantilever cases. It is also possible to realise larger efficiency gains by increasing the Cosserat segment size; we used 10 nm for illustrative purposes,

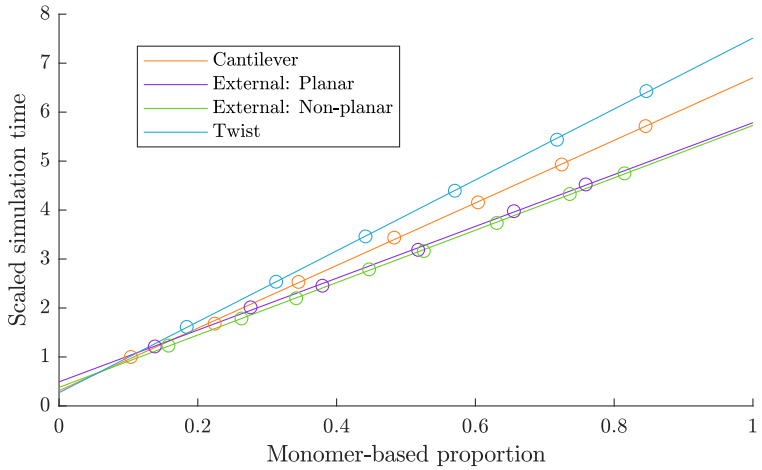


Figure C.6: *Relative simulation times for example cases presented. We plot the time taken to simulate each of the example cases over an equal length of time, and scale relative to filament length and the shortest simulation time across the examples (which therefore has a time equal to* 1*). This can be thought of as a normalised measure of simulation time per unit length. A linear fit line is plotted for example case.*

but it is possible to maintain stability and accuracy with 100 nm sized segments for situations where bending is simple.

References

[1] R. S. Gunaratne et al. “Multi-resolution dimer models in heat baths with short-range and long-range interactions”. *Interface Focus* 9.3 (2019).

[2] R. S. Gunaratne et al. *On Stretching, Bending, Shearing and Twisting of Actin Filaments II: Multi-Resolution Modelling*. submitted to *Journal of Chemical Theory and Computation*. 2022.

[3] C. Floyd et al. *On Stretching, Bending, Shearing and Twisting of Actin Filaments I: Variational Models*. submitted to *Journal of Chemical Theory and Computation*, arXiv: 2112.01480. 2021.

[4] J. N. Israelachvili. *Intermolecular and Surface Forces: Third Edition*. Elsevier Inc., 2011, pp. 1–676.

[5] J. S. Rowlinson. “Cohesion : a scientific history of intermolecular forces” (2002), p. 333.

[6] B. R. Brooks et al. “CHARMM: The biomolecular simulation program”. *Journal of Computational Chemistry* 30.10 (2009), pp. 1545–1614.

[7] W. F. van Gunsteren et al. “Validation of molecular dynamics simulation”. *The Journal of Chemical Physics* 108.15 (1998), pp. 6109–6116.

[8] H. C. Urey et al. “The vibrations of pentatonic tetrahedral molecules”. *Physical Review* 38.11 (1931), pp. 1969–1978.

[9] L. D. Landau et al. “Course of Theoretical Physics Volume 1: Mechanics”. *Course of Theoretical Physics* 1 (1982).

[10] J. E. Lennard-Jones. “Cohesion”. *Proceedings of the Physical Society* 43.5 (1931), pp. 461–482.

[11] J. E. Jones. “On the Determination of Molecular Fields. II. From the Equation of State of a Gas”. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 106.738 (1924), pp. 463–477.

[12] W. L. Jorgensen et al. “Comparison of simple potential functions for simulating liquid water”. *The Journal of Chemical Physics* 79.2 (1983), pp. 926–935.

[13] D. J. Huggins. “Correlations in liquid water for the TIP3P-Ewald, TIP4P-2005, TIP5P-Ewald, and SWM4-NDP models”. 136.6 (2012), p. 064518.

[14] P. Mark et al. “Structure and dynamics of the TIP3P, SPC, and SPC/E water models at 298 K”. *Journal of Physical Chemistry A* 105.43 (2001), pp. 9954–9960.

[15] H. W. Horn et al. “Development of an improved four-site water model for biomolecular simulations: TIP4P-Ew”. *Journal of Chemical Physics* 120.20 (2004), pp. 9665–9678.

[16] Y. Khalak et al. "Improved general-purpose five-point model for water: TIP5P/2018". *Journal of Chemical Physics* 149.22 (2018), p. 224507.

[17] M. W. Mahoney et al. "A five-site model for liquid water and the reproduction of the density anomaly by rigid, nonpolarizable potential functions". *Journal of Chemical Physics* 112.20 (2000), pp. 8910–8922.

[18] H. Nada et al. "An intermolecular potential model for the simulation of ice and water near the melting point: A six-site model of H2O". *Journal of Chemical Physics* 118.16 (2003), pp. 7401–7413.

[19] C. Vega et al. "The melting temperature of the most common models of water". *Journal of Chemical Physics* 122.11 (2005), p. 144506.

[20] J. Zielkiewicz. "Structural properties of water: Comparison of the SPC, SPCE, TIP4P, and TIP5P models of water". *Journal of Chemical Physics* 123.10 (2005), p. 104501.

[21] A. V. Onufriev et al. "Water models for biomolecular simulations". *WIREs Computational Molecular Science* 8.2 (2018), e1347.

[22] D. R. Nutt et al. "Molecular dynamics simulations of proteins: Can the explicit water model be varied?" *Journal of Chemical Theory and Computation* 3.4 (2007), pp. 1550–1560.

[23] S. Nosé. "A unified formulation of the constant temperature molecular dynamics methods". *Journal of Chemical Physics* 81.1 (1984), pp. 511–519.

[24] W. Hoover. "Canonical dynamics: Equilibrium phase-space distributions". *Physical Review A* 31.3 (1985), pp. 1695–1697.

[25] H. J. Berendsen et al. "Molecular dynamics with coupling to an external bath". *The Journal of Chemical Physics* 81.8 (1984), pp. 3684–3690.

[26] H. C. Andersen. "Molecular dynamics simulations at constant pressure and/or temperature". *The Journal of Chemical Physics* 72.4 (1980), pp. 2384–2393.

[27] J. Ruiz-Franco et al. "On the effect of the thermostat in non-equilibrium molecular dynamics simulations". *The European Physical Journal E* 41.80 (2018).

[28] R. Dominguez et al. "Actin Structure and Function". *Annu Rev Biophys* (2011), pp. 169–186.

[29] F. Merino et al. "Structural transitions of F-actin upon ATP hydrolysis at near-atomic resolution revealed by cryo-EM". *Nature Structural and Molecular Biology* 25.6 (2018), pp. 528–537.

[30] S. Z. Chou et al. "Mechanism of actin polymerization revealed by cryo-EM structures of actin filaments with three different bound nucleotides". *Proceedings of the National Academy of Sciences of the United States of America* 116.10 (2019), pp. 4265–4274.

[31] H. M. Berman et al. *The Protein Data Bank*. Tech. rep. 1. 2000.

[32] P. S. Gurel et al. "Cryo-EM structures reveal specialization at the myosin VI-actin interface and a mechanism of force sensitivity". *eLife* 6 (2017).

[33] B. Leimkuhler et al. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*. Springer International Publishing, 2015.

[34] G. Bussi et al. “Accurate sampling using Langevin dynamics”. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 75.5 (2007), p. 056707.

[35] D. Frenkel et al. *Understanding molecular simulation: From algorithms to applications*. Vol. 50. 7. 1996, pp. 66–66.

[36] L. Verlet. “Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules”. *Physical Review* 159.1 (1967), pp. 98–103.

[37] W. C. Swope et al. “A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters”. *The Journal of Chemical Physics* 76.1 (1982), pp. 637–649.

[38] D. Janežič et al. “Implicit Runge-Kutta Method for Molecular Dynamics Integration”. *Journal of Chemical Information and Computer Sciences* 33.2 (1993), pp. 252–257.

[39] H. C. Andersen. “Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations”. *Journal of Computational Physics* 52.1 (1983), pp. 24–34.

[40] S. G. Lambrakos et al. “A modified shake algorithm for maintaining rigid bonds in molecular dynamics simulations of large molecules”. *Journal of Computational Physics* 85.2 (1989), pp. 473–486.

[41] T. R. Forester et al. “SHAKE, rattle, and roll: Efficient constraint algorithms for linked rigid bodies”. *Journal of Computational Chemistry* 19.1 (1998), pp. 102–111.

[42] S. Miyamoto et al. “Settle: An analytical version of the SHAKE and RATTLE algorithm for rigid water models”. *Journal of Computational Chemistry* 13.8 (1992), pp. 952–962.

[43] M. C. Melo et al. “NAMD goes quantum: An integrative suite for hybrid simulations”. *Nature Methods* 15.5 (2018), pp. 351–354.

[44] W. Humphrey et al. “VMD: Visual molecular dynamics”. *Journal of Molecular Graphics* 14.1 (1996), pp. 33–38.

[45] H. J. Berendsen et al. “GROMACS: A message-passing parallel molecular dynamics implementation”. *Computer Physics Communications* 91.1-3 (1995), pp. 43–56.

[46] S. Plimpton. “Fast parallel algorithms for short-range molecular dynamics”. *Journal of Computational Physics* 117.1 (1995), pp. 1–19.

[47] K. R. Hadley et al. “Coarse-grained molecular models of water: A review”. *Molecular simulation* 38.8-9 (2012), pp. 671–681.

[48] S. Kmiecik et al. “Coarse-Grained Protein Models and Their Applications”. *Chemical Reviews* 116.14 (2016), pp. 7898–7936.

[49] A. C. Schramm et al. “Plastic Deformation and Fragmentation of Strained Actin Filaments”. *Biophysical Journal* 117.3 (2019), pp. 453–463.

[50] S. J. Marrink et al. “The MARTINI force field: Coarse grained model for biomolecular simulations”. *Journal of Physical Chemistry B* 111.27 (2007), pp. 7812–7824.

[51] J. F. Dama et al. "The theory of ultra-coarse-graining. 1. General principles". *Journal of Chemical Theory and Computation* 9.5 (2013), pp. 2466–2480.

[52] S. J. Marrink et al. "Perspective on the martini model". *Chemical Society Reviews* 42.16 (2013), pp. 6801–6822.

[53] W. Schommers. "Pair potentials in disordered many-particle systems: A study for liquid gallium". *Physical Review A* 28.6 (1983), pp. 3599–3605.

[54] D. Reith et al. *Deriving Effective Mesoscale Potentials from Atomistic Simulations*. Tech. rep. 2003, pp. 1624–1636.

[55] Z. Li et al. "A comparative study of coarse-graining methods for polymeric fluids: Mori-Zwanzig vs. iterative Boltzmann inversion vs. stochastic parametric optimization". *Journal of Chemical Physics* 145.4 (2016), p. 044102.

[56] T. C. Moore et al. "Derivation of coarse-grained potentials via multistate iterative Boltzmann inversion". *Journal of Chemical Physics* 140.22 (2014), p. 224104.

[57] A. Savelyev et al. "Chemically accurate coarse graining of double-stranded DNA". *Proceedings of the National Academy of Sciences of the United States of America* 107.47 (2010), pp. 20340–20345.

[58] R. H. Swendsen. "Monte Carlo Renormalization Group". *Physical Review Letters* 42.14 (1979), pp. 859–861.

[59] E. Lyman et al. "Systematic Multiscale Parameterization of Heterogeneous Elastic Network Models of Proteins". *Biophysical Journal* 95.9 (2008), pp. 4183–4192.

[60] J. W. Chu et al. "Coarse-grained modeling of the actin filament derived from atomistic-scale simulations". *Biophysical Journal* 90.5 (2006), pp. 1572–1582.

[61] L. Darré et al. "Another coarse grain model for aqueous solvation: WAT FOUR?" *Journal of Chemical Theory and Computation* 6.12 (2010), pp. 3793–3807.

[62] H. Chan et al. "Machine learning coarse grained models for water". *Nature Communications* 10.379 (2019).

[63] J. Zavadlav et al. "Bayesian selection for coarse-grained models of liquid water". *Scientific Reports* 9.99 (2019).

[64] R. Erban. "Coarse-graining molecular dynamics: stochastic models with non-Gaussian force distributions". *Journal of Mathematical Biology* 80.1-2 (2020), pp. 457–479.

[65] G. W. Ford et al. "Statistical mechanics of assemblies of coupled oscillators". *Journal of Mathematical Physics* 6.4 (1965), pp. 504–515.

[66] R. Zwanzig. *Nonequilibrium statistical mechanics*. Oxford University Press, 2001.

[67] R. Zwanzig. "Nonlinear generalized Langevin equations". *Journal of Statistical Physics* 9.3 (1973), pp. 215–220.

[68] H. Mori. "Transport, Collective Motion, and Brownian Motion". *Progress of Theoretical Physics* 33.3 (1965), pp. 423–455.

[69] R. Kubo. "The fluctuation-dissipation theorem". *Reports on Progress in Physics* 29.1 (1966), pp. 255–284.

[70] G. Jung et al. "Iterative Reconstruction of Memory Kernels". *Journal of Chemical Theory and Computation* 13.6 (2017), pp. 2481–2488.

[71] H. K. Shin et al. “Brownian motion from molecular dynamics”. *Chemical Physics* 375.2-3 (2010), pp. 316–326.

[72] R. D. Groot et al. “Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation”. *Journal of Chemical Physics* 107.11 (1997), pp. 4423–4435.

[73] M. Levitt et al. “Computer simulation of protein folding”. *Nature* 253.5494 (1975), pp. 694–698.

[74] M. Karplus et al. “Molecular dynamics simulations of biomolecules”. *Nature Structural Biology* 9.9 (2002), pp. 646–652.

[75] M. B. Flegg et al. “The two-regime method for optimizing stochastic reaction-diffusion simulations”. *Journal of the Royal Society Interface* 9.70 (2012), pp. 859–868.

[76] M. Robinson et al. “Multiscale reaction-diffusion simulations with Smoldyn”. *Bioinformatics* 31.14 (2015), pp. 2406–2408.

[77] M. Praprotnik et al. “Adaptive resolution simulation of liquid water”. *Journal of Physics Condensed Matter* 19.29 (2007), p. 292201.

[78] B. Ensing et al. “Energy conservation in adaptive hybrid atomistic/coarse-grain molecular dynamics”. *Journal of Chemical Theory and Computation* 3.3 (2007), pp. 1100–1105.

[79] M. Praprotnik et al. “Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly”. *Journal of Chemical Physics* 123.22 (2005), p. 224106.

[80] R. Potestio et al. “Computer simulations of soft matter: Linking the scales”. *Entropy* 16.8 (2014), pp. 4199–4245.

[81] J. Zavadlav et al. “Adaptive resolution simulation of an atomistic protein in MARTINI water”. *Journal of Chemical Physics* 140.5 (2014).

[82] J. Zavadlav et al. “Adaptive Resolution Simulation of a DNA Molecule in Salt Solution”. *Journal of Chemical Theory and Computation* 11.10 (2015), pp. 5035–5044.

[83] J. Zavadlav et al. “Adaptive resolution simulations coupling atomistic water to dissipative particle dynamics”. *Journal of Chemical Physics* 147.11 (2017), p. 114110.

[84] M. R. Machado et al. “MD Simulations of Viruslike Particles with Supra CG Solvation Affordable to Desktop Computers”. *Journal of Chemical Theory and Computation* 13.10 (2017), pp. 5106–5116.

[85] E. Tarasova et al. “All-Atom Molecular Dynamics Simulations of Entire Virus Capsid Reveal the Role of Ion Distribution in Capsid’s Stability”. *Journal of Physical Chemistry Letters* 8.4 (2017), pp. 779–784.

[86] M. B. Flegg et al. “Analysis of the two-regime method on square meshes”. *SIAM Journal on Scientific Computing* 36.3 (2014).

[87] M. B. Flegg et al. “Convergence of methods for coupling of microscopic and mesoscopic reaction-diffusion simulations”. *Journal of Computational Physics* 289 (2015), pp. 1–17.

[88] M. Robinson et al. "Adaptive two-regime method: Application to front propagation". *Journal of Chemical Physics* 140.12 (2014), p. 124109.

[89] C. A. Smith et al. "Spatially extended hybrid methods: A review". *Journal of the Royal Society Interface* 15.139 (2018), p. 20170931.

[90] B. Franz et al. "Multiscale reaction-diffusion algorithms: PDE-assisted Brownian dynamics". *SIAM Journal on Applied Mathematics* 73.3 (2013), pp. 1224–1247.

[91] R. Delgado-Buscalioni et al. "Coupling atomistic and continuum hydrodynamics through a mesoscopic model: Application to liquid water". *Journal of Chemical Physics* 131.24 (2009), p. 244107.

[92] A. C. Fogarty et al. "A multi-resolution model to capture both global fluctuations of an enzyme and molecular recognition in the ligand-binding site". *Proteins: Structure, Function and Bioinformatics* 84.12 (2016), pp. 1902–1913.

[93] N. Di Pasquale et al. "A multiple time step scheme for multiresolved models of Macromolecules". *Journal of Computational Chemistry* 35.16 (2014), pp. 1199–1207.

[94] N. Di Pasquale et al. "Local and global dynamics of multi-resolved polymer chains: Effects of the interactions atoms-beads on the dynamic of the chains". *Journal of Chemical Physics* 146.8 (2017), p. 084905.

[95] E. Rolls et al. "Varying the resolution of the Rouse model on temporal and spatial scales: Application to multiscale modeling of DNA dynamics". *Multiscale Modeling and Simulation* 15.4 (2017), pp. 1672–1693.

[96] E. Rolls et al. "Multi-resolution polymer Brownian dynamics with hydrodynamic interactions". *Journal of Chemical Physics* 148.19 (2018), p. 194111.

[97] E. M. De La Cruz et al. "Origin of twist-bend coupling in actin filaments". *Biophysical Journal* 99.6 (2010), pp. 1852–1860.

[98] M. Gazzola et al. "Forward and inverse problems in the mechanics of soft filaments". *Royal Society Open Science* 5.6 (2018), p. 171628.

[99] T. D. Pollard et al. "Actin, a central player in cell shape and movement". *Science* 326.5957 (2009), pp. 1208–1212.

[100] H. Lodish et al. *Molecular Cell Biology*. 9th edition New York W.H.Freeman & Co Ltd. 2021.

[101] J. Pfaendtner et al. "Structure and dynamics of the actin filament". *Journal of Molecular Biology* 396.2 (2010), pp. 252–263.

[102] A. Shamloo et al. "Nanomechanics of actin filament: A molecular dynamics simulation". *Cytoskeleton* 75.3 (2018), pp. 118–130.

[103] K. Tanaka et al. "Structural basis for cofilin binding and actin filament disassembly". *Nature Communications* 9.1 (2018).

[104] A. Narita et al. "Structural basis of actin filament capping at the barbed-end: A cryo-electron microscopy study". *EMBO Journal* 25.23 (2006), pp. 5626–5633.

[105] T. Oda et al. "The nature of the globular- to fibrous-actin transition". *Nature* 457.7228 (2009), pp. 441–445.

[106] J. W. Chu et al. "Allostery of actin filaments: Molecular dynamics simulations and coarse-grained analysis". *Proceedings of the National Academy of Sciences of the United States of America* 102.37 (2005), pp. 13111–13116.

[107] K. Popov et al. "MEDYAN: Mechanochemical Simulations of Contraction and Polarity Alignment in Actomyosin Networks". *PLoS Computational Biology* 12.4 (2016). Ed. by B. Ermentrout, e1004877.

[108] C. Floyd et al. "Quantifying dissipation in actomyosin networks". *Interface Focus* 9.3 (2019), p. 20180078.

[109] M. E. Janson et al. "Crosslinkers and Motors Organize Dynamic Microtubules to Form Stable Bipolar Arrays in Fission Yeast". *Cell* 128.2 (2007), pp. 357–368.

[110] F. J. Nédélec et al. "Self-organization of microtubules and motors". *Nature* 389.6648 (1997), pp. 305–308.

[111] H. Yamaoka et al. "Continuum dynamics on a vector bundle for a directed medium". *Journal of Physics A: Mathematical and Theoretical* 43.32 (2010), p. 325209.

[112] H. Yamaoka et al. "Coupling between axial stretch and bending/twisting deformation of actin filaments caused by a mismatched centroid from the center axis". *International Journal of Mechanical Sciences* 52.2 (2010). SPECIAL ISSUE: Advances in Modeling and Evaluation of Materials in Honor of Professor Tomita, pp. 329–333.

[113] M. Bergou et al. "Discrete elastic rods". *ACM Transactions on Graphics* 27.3 (2008), pp. 1–12.

[114] L. D. Landau et al. *Theory of elasticity*. Butterworth-Heinemann, 1986.

[115] B. Audoly et al. "A discrete geometric approach for simulating the dynamics of thin viscous threads". *Journal of Computational Physics* 253 (2013), pp. 18–49.

[116] Z. Wang et al. "Real time simulation of inextensible surgical thread using a Kirchhoff rod model with force output for haptic feedback applications". *International Journal of Solids and Structures* 113-114 (2017), pp. 192–208.

[117] M. Bergou et al. "Discrete Viscous Threads". In: *ACM SIGGRAPH 2010 Papers*. SIGGRAPH '10. Los Angeles, California: Association for Computing Machinery, 2010.

[118] U. Dobramysl et al. "Steric Effects Induce Geometric Remodeling of Actin Bundles in Filopodia". *Biophysical Journal* 110.9 (2016), pp. 2066–2075.

[119] A. C. Schramm et al. "Actin Filament Strain Promotes Severing and Cofilin Dissociation". *Biophysical Journal* 112.12 (2017), pp. 2624–2633.

[120] B. Horan et al. "Insights into actin polymerization and nucleation using a coarse grained model". *bioRxiv* (2019), p. 715383.

[121] H. H. Katkar et al. "Insights into the Cooperative Nature of ATP Hydrolysis in Actin Filaments". *Biophysical Journal* 115.8 (2018), pp. 1589–1602.

[122] F. Aydin et al. "Multiscale simulation of actin filaments and actin-associated proteins". *Biophysical Reviews* 10.6 (2018), pp. 1521–1535.

[123] H. Yamaoka et al. “Multiscale modeling and mechanics of filamentous actin cytoskeleton”. *Biomechanics and Modeling in Mechanobiology* 11.3-4 (2012), pp. 291–302.

[124] R. Holley. “The motion of a heavy particle in an infinite one dimensional gas of hard spheres”. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 17.3 (1971), pp. 181–219.

[125] D. Dürr et al. “A mechanical model of Brownian motion”. *Communications in Mathematical Physics* 78.4 (1981), pp. 507–530.

[126] J. Dunkel et al. “Relativistic Brownian motion: From a microscopic binary collision model to the Langevin equation”. *Phys. Rev. E* 74 (5 2006), p. 051106.

[127] R. Erban. “From molecular dynamics to Brownian dynamics”. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 470.2167 (2014).

[128] C. P. Robert. “Simulation of truncated normal variables”. *Statistics and Computing* 5.2 (1995), pp. 121–125.

[129] W. F. Van Gunsteren et al. “Algorithms for brownian dynamics”. *Molecular Physics* 45.3 (1982), pp. 637–647.

[130] W. Wang et al. “Analysis of a few numerical integration methods for the Langevin equation”. *Molecular Physics* 101.14 (2003), pp. 2149–2156.

[131] A. Brünger et al. “Stochastic boundary conditions for molecular dynamics simulations of ST2 water”. *Chemical Physics Letters* 105.5 (1984), pp. 495–500.

[132] N. Grønbech-Jensen et al. “A simple and effective verlet-type algorithm for simulating langevin dynamics”. *Molecular Physics* 111.8 (2013), pp. 983–991.

[133] K. Burrage et al. “Numerical methods for second-order stochastic differential equations”. *SIAM Journal on Scientific Computing* 29.1 (2007), pp. 245–264.

[134] A. Davtyan et al. “Dynamic force matching: A method for constructing dynamical coarse-grained models with realistic time dependence”. *Journal of Chemical Physics* 142.15 (2015), p. 154104.

[135] R. Erban. “Coupling all-atom molecular dynamics simulations of ions in water with Brownian dynamics”. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 472.2186 (2016).

[136] A. Davtyan et al. “Dynamic force matching: Construction of dynamic coarse-grained models with realistic short time dynamics and accurate long time dynamics”. *Journal of Chemical Physics* 145.22 (2016), p. 224107.

[137] B. Leimkuhler et al. “A gentle stochastic thermostat for molecular dynamics”. *Journal of Statistical Physics* 135.2 (2009), pp. 261–277.

[138] R. E. Miller et al. “A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods”. *Modelling and Simulation in Materials Science and Engineering* 17.5 (2009), p. 053001.

[139] I. G. Kevrekidis et al. “Equation-Free, Coarse-Grained Multiscale Computation: Enabling Mocroscopic Simulators to Perform System-Level Analysis”. *Communications in Mathematical Sciences* 1.4 (2003), pp. 715–762.

[140] E. Biyikli et al. "Multiresolution Molecular Mechanics: Dynamics". *Computer Methods in Applied Mechanics and Engineering* 274 (2014), pp. 42–55.

[141] R. Erban et al. *Stochastic modelling of reaction-diffusion processes*. Cambridge University Press, 2019.

[142] J.-L. Barrat et al. "Portable Implementation of a Quantum Thermal Bath for Molecular Dynamics Simulations". *Journal of Statistical Physics* 1 (2011), pp. 679–689.

[143] G. Jung et al. "Generalized Langevin dynamics: construction and numerical integration of non-Markovian particle-based models". *Soft Matter* 14.46 (2018), pp. 9368–9382.

[144] P. Linz. "Numerical methods for Volterra integral equations of the first kind". *The Computer Journal* 12.4 (1969), pp. 393–397.

[145] P. Linz. "Product integration methods for Volterra integral equations of the first kind". *BIT* 11.4 (1971), pp. 413–421.

[146] A.-M. Wazwaz. *Linear and Nonlinear Integral Equations*. Springer Berlin Heidelberg, 2011.

[147] R. Erban et al. "Stochastic modelling of reaction–diffusion processes: Algorithms for bimolecular reactions". *Physical Biology* 6.4 (2009), p. 046001.

[148] E. Roberts et al. "Lattice microbes: High-performance stochastic simulation method for the reaction-diffusion master equation". *Journal of Computational Chemistry* 34.3 (2013), pp. 245–255.

[149] R. Erban et al. "Multiscale Stochastic Reaction-Diffusion Modeling: Application to Actin Dynamics in Filopodia". *Bulletin of Mathematical Biology* 76.4 (2014), pp. 799–818.

[150] U. Dobramysl et al. "Particle-based multiscale modeling of calcium puff dynamics". *Multiscale Modeling and Simulation* 14.3 (2016), pp. 997–1016.

[151] T. W. Allen et al. "Molecular dynamics estimates of ion diffusion in model hydrophobic and KcsA potassium channels". *Biophysical Chemistry* 86.1 (2000), pp. 1–14.

[152] M. Jensen et al. "Principles of conduction and hydrophobic gating in K+ channels". *Proceedings of the National Academy of Sciences of the United States of America* 107.13 (2010), pp. 5833–5838.

[153] R. Erban et al. "Reactive boundary conditions for stochastic simulations of reaction-diffusion processes". *Physical Biology* 4.1 (2007), pp. 16–28.

[154] R. Erban et al. "Time scale of random sequential adsorption". *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 75.4 (2007), p. 041116.

[155] S. Chapman Jonathan et al. "Reactive boundary conditions as limits of interaction potentials for brownian and langevin dynamics". *SIAM Journal on Applied Mathematics* 76.1 (2016), pp. 368–390.

[156] R. Delgado-Buscalioni et al. "Open boundary molecular dynamics". *European Physical Journal: Special Topics* 224.12 (2015), pp. 2331–2349.

[157] J. Zavadlav et al. "Open-Boundary Molecular Dynamics of a DNA Molecule in a Hybrid Explicit/Implicit Salt Solution". *Biophysical Journal* 114.10 (2018), pp. 2352–2362.

[158] P. Español. "Hybrid description of complex molecules". *Europhysics Letters* 88.4 (2009), p. 40008.

[159] N. Di Pasquale et al. "Systematic derivation of hybrid coarse-grained models". *Physical Review E* 99.1 (2019), p. 013303.

[160] H. C. Gonzalez et al. "Transferable mixing of atomistic and coarse-grained water models". *Journal of Physical Chemistry B* 117.46 (2013), pp. 14438–14448.

[161] Z. Liu et al. "Hybrid modeling and simulation of stochastic effects on progression through the eukaryotic cell cycle". *Journal of Chemical Physics* 136.3 (2012).

[162] A. Duncan et al. "Hybrid framework for the simulation of stochastic chemical kinetics". *Journal of Computational Physics* 326 (2016), pp. 398–419.

[163] I. P. Omelyan et al. "Optimized Forest–Ruth- and Suzuki-like algorithms for integration of motion in many-body systems". *Computer Physics Communications* 146.2 (2002), pp. 188–202.

[164] O. Rodrigues. "Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire." *Journal de Mathématiques Pures et Appliquées* (1840), pp. 380–440.

[165] V. E. Galkin et al. "Near-Atomic Resolution for One State of F-Actin". *Structure* 23.1 (2015), pp. 173–182.

[166] E. Krissinel et al. "Inference of Macromolecular Assemblies from Crystalline State". *Journal of Molecular Biology* 372.3 (2007), pp. 774–797.

[167] "Sur la sphère vide. A la mémoire de Georges Voronoï". *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na* 6 (1934), pp. 793–800.

[168] J. Fan et al. "Molecular Origins of Cofilin-Linked Changes in Actin Filament Mechanics". 425.7 (2013), pp. 1225–40.

[169] E. Grazi. "What is the diameter of the actin filament?" *FEBS Letters* 405.3 (1997), pp. 249–252.

[170] H. Kojima et al. "Direct measurement of stiffness of single actin filaments with and without tropomyosin by in vitro nanomanipulation". *Proceedings of the National Academy of Sciences of the United States of America* 91.26 (1994), pp. 12962–12966.

[171] Y. Tseng et al. "Functional synergy of actin filament cross-linking proteins". *Journal of Biological Chemistry* 277.28 (2002), pp. 25609–25616.

[172] J. Gere et al. *Mechanics of Materials*. PWS Publishing Company, 1997.

[173] B. R. McCullough et al. "Cofilin Increases the Bending Flexibility of Actin Filaments: Implications for Severing and Cell Mechanics". *Journal of Molecular Biology* 381.3 (2008), pp. 550–558.

[174] H. Isambert et al. “Flexibility of actin filaments derived from thermal fluctuations. Effect of bound nucleotide, phalloidin, and muscle regulatory proteins”. *Journal of Biological Chemistry* 270.19 (1995), pp. 11437–11444.

[175] A. Ott et al. “Measurement of the persistence length of polymerized actin using fluorescence microscopy”. *Physical Review E* 48.3 (1993), R1642.

[176] B. Ling et al. “Experimentally program large magnitude of Poisson’s ratio in additively manufactured mechanical metamaterials”. *International Journal of Mechanical Sciences* 173 (2020), p. 105466.

[177] T. C. Michaels et al. “Mechanical basis for fibrillar bundle morphology”. *Soft Matter* 16.40 (2020), pp. 9306–9318.

[178] O. Maxian et al. *The hydrodynamics of a twisting, bending, inextensible fiber in Stokes flow*. arXiv: 2201.04187. 2022.

[179] I. Omelyan et al. “Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations”. *Computer Physics Communications* 151.3 (2003), pp. 272–314.

[180] *MATLAB version (R2021a)*. The Mathworks, Inc. Natick, Massachusetts, 2021.

[181] N. Leijnse et al. “Helical buckling of actin inside filopodia generates traction”. *Proceedings of the National Academy of Sciences of the United States of America* 112.1 (2015), pp. 136–141.

[182] F. Nédélec et al. “Collective Langevin dynamics of flexible cytoskeletal fibers”. *New Journal of Physics* 9.11 (2007), p. 427.

[183] S. L. Freedman et al. “A versatile framework for simulating the dynamic mechanical structure of cytoskeletal networks”. *Biophysical journal* 113.2 (2017), pp. 448–460.

[184] G. Ciccotti et al. “Molecular dynamics simulation of rigid molecules”. *Computer Physics Reports* 4.6 (1986), pp. 346–392.

[185] P. Tao et al. “Maintain rigid structures in Verlet based Cartesian molecular dynamics simulations”. *The Journal of Chemical Physics* 137.13 (2012), p. 134110.