UNIVERSITY OF LEEDS

KOBRA: A Fluctuating Elastic Rod Model for Slender Biological Macromolecules

Robert Welch

University of Leeds

School of Physics and Astronomy

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

December, 2020

Intellectual Property Statement

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others. All of the authors listed on the FFEA [1] and KOBRA [2] software publications have contributed to discussions, and authors of the latter have provided ideas, insights and corrections to this manuscript and the work presented within.

Parts of chapters 2 and 3 are adapted from a publication [2] of which the co-authors are the candidate's supervisors, and for which the candidate wrote the manuscript, designed the algorithm and implemented the software.

Figure 3.14 is reproduced with permission from the Journal of Molecular Biology. Figure 1.8 is reproduced with permission from the Biophysical Journal. Figures 3.9, 1.2, 1.7, 1.5, 1.6, and 4.5 are reproduced with permission from Springer Nature. Figures 1.3 and 1.15 are reproduced with permission from Elsevier. Figure 1.10 is reproduced under the terms of the Standard ACS AuthorChoice/Editors' Choice Usage Agreement. Figure 1.13 is reproduced with permission from the American Physical Society. Figures 1.9 and 5.3 are reproduced under the terms of the CC BY 4.0 license.

Acknowledgements

Doesn't writing an acknowledgements page here feel a bit presumptuous? Like giving your Oscar acceptance speech before you've got the Oscar. Oh well, here I go.

For the last few years, I have been continually folded over onto myself, much like a protein, and have found many new minima. Thank you to all those who have enabled this.

My supervisory team - Sarah Harris, Daniel Read and Oliver Harlen. This might seem like a formality, but it's definitely not, and I would refuse to acknowledge them if I didn't think they were great.

FFEA group members, past and present - Tom Ridley, Ben Hanson, Albert Solernou, Glenn Carrington, Robin Richardson, and Robin Oliver. Thanks to Tomo Tanaka for his extensive advice regarding the kinetochore. There are too many project students to name, but special thanks goes to Samantha Coffey, the first ever KOBRA user.

On a personal level, I thank Matthew Vaughan, whose constant fixing of broken household appliances and loading of the dishwasher cleared enough time in my schedule for at least 10 extra pages worth of thesis. And my parents, I guess (for some reason, they never limited the amount of time I could spend on the computer).

And now, in true FFEA group tradition, I leave my apologies to the next PhD students. What for, I'm not sure yet. But sorry anyway. It seemed a good idea at the time.

Abstract

Computational Biophysics lies at the intersection between biology, physics, applied mathematics and software engineering. Some of the most burning questions in molecular biology are concerned with biomechanical systems, the dynamics of which are driven by chemistry and physics. Unfortunately, we have extremely limited means to observe these dynamics experimentally. In the past, this problem has been solved with the use of molecular dynamics, sometimes referred to as a 'computational microscope'. Studying biomolecules *in silico* can provide a wealth of new information at temporal and spatial resolutions far beyond any current imaging modality. But molecular dynamics algorithms are limited by current computing power, and by the assumptions used to construct them.

The kinetochore, a supramolecular structure crucial to the process of cell division, operates on time and length scales outside the reach of atomistic molecular dynamics with current computing power. To overcome this limitation, we propose a new, coarse-grained algorithm, which allows for a more computationally inexpensive representation of the biomolecules that comprise the kinetochore.

This algorithm, KOBRA (KirchOff Biological Rod Algorithm) is designed to perform dynamical simulations of elongated biomolecules such as those containing alpha-helices and coiled-coils. It represents these as coarsely-discretised Kirchoff rods, with linear elements that can stretch, bend and twist independently. These rods can have anisotropic and inhomogeneous parameters and bent or twisted equilibrium structures, allowing for a coarse-grained parameterisation of complex biological structures. Each element is non-inertial and subject to thermal fluctuations. This coarse-grained representation allows for simulations of extremely large, long-lived systems at the biological mesoscale.

KOBRA has been extended with a parameterisation scheme that allows for rod parameters (in terms of stretching, bending and twisting constants) to be extracted from all-atom simulation trajectories. An all-atom representation of Ndc80C - a sub-unit of the kinetochore - was constructed, and the KOBRA parameters for the molecule were extracted from its trajectory.

The KOBRA algorithm is validated against both the physics of elastic rods and the biology of Ndc80C and the kinetochore. A partial kinetochore system was constructed and simulated using KOBRA and FFEA (Fluctuating Finite Element Analysis). The resulting trajectories were analysed and used to investigate the microtubule-binding ability of Ndc80C in a variety of configurations. A C++ implementation of KOBRA is available under the GNU GPLv3 free software licence, and can be downloaded at http://ffea.bitbucket.io.

Contents

List of Figures

List of Tables

Abbreviations, Symbols & Jargon

Table 1: Abbreviations

DNA	Deoxyribonucleic acid	Ndc80C	The protein complex made from Ndc80, Nuf2, Spc24 and Spc25 (C is short for complex)
Ndc80	part of NDC80C	Spc24	part of NDC80C
Nuf2	part of NDC80C	Spc25	part of NDC80C
ODE	Ordinary Differential Equation	DAM1	ring-shaped protein complex that encircles microtubles and binds Ndc80C
EM	Electron Microscopy	MD	Molecular Dynamics
ATP	Adenosine Triphosphate	KAR3	kinesin-like protien
MT	Microtubule	GROMACS	Growing Molecules and Cloning Shrimps/GROningen MAchine for Chemical Simulations
AMBER	Assisted Model Building with Energy Refinement	CHARMM	Chemistry at HARvard Macromolecular Mechanics
MARTINI	has no particular meaning	UNRES	UNited RESidue
CABS	C-Alpha, C-Beta	KOBRA	KirchOff Biological Rod Algorithm

FFEA	Fluctuating Finite Element Analysis	VFX	Visual Effects
CoRdE	Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects	STRANDS	no particlar meaning
OpenMP	Open Multi-Processing	VdW	Van der Waals
2VE7	the PDB code for Bonsai Ndc80C	QUARK	no particular meaning
UCSF	University of California, San Francisco	GBSA	Generalized Born Solvent Accessible Surface Area
YASARA	Yet Another Scientific Artificial Reality Application	SCWRL4	no particular meaning
PCA	Principal Component Analysis	RMSD	Root-mean-squared deviation
RMSE	Root-mean-squared error		

Table 2: Domain-specific jargon

Dynamic	out of equilibrium, in motion
Mesoscale	larger than proteins, smaller than cells
Conserved	Has roughly the same DNA or protein sequence between species.
Coarse-graining	removing detail from a mathematical description of an object
Viscoelastic	elastic but affected by a viscous drag (normally from a fluid medium)
Trajectory	a description of the dynamics of the system, e.g. where all the objects are at each point in time

Brownian dynamics	equations of motion with thermal fluctuations and no inertia
Kirchoff and Cosserat rods	representations of rods in 3-d and derivations of the energies and forces acting on them
Anisotropic	has different properties in different directions
Inhomogeneous	not the same everywhere
Unit vector/normalised vector	vector with a length of 1 (any vector can be normalized into a unit vector)
Orthonormal	orthogonal unit vectors
Hookean/harmonic	equation governing the energy is quadratic in form
Degree of freedom	a way a physical system can be modified to change the energy.
Stochastic	containing random noise, non-deterministic
Timestep	temporal resolution of a simulation
Residue	amino acid
Principal component / mode	a mode of vibration analogous to a normal mode
Side chain	the unique part of an amino acid
Supramolecular	on a scale larger than individual molecules
Forcefield	pre-computed potentials for atoms and residues
Van der Waals	short-range attractive force between molecules
Continuum	lacking discrete or easily-resolvable elements
Cryo-EM	Electron microscopy technique performed on molecules at very low temperatures
Negative stain EM	Electron microscopy technique based on contrast with an opaque medium
Lennard-Jones	simplified interatomic potential comprising of repulsive strong and attractive Van der Waals forces
Constitutive equation	an equation linking physical quantities of a material
Framed curve	a curve described by the orientation of co-ordinate systems
Coiled-coil	a coil made of two alpha-helices (which are, in themselves, coils)

Macroscopic	large, at least compared to proteins and molecular structures
Hockles	loops that form in twisted rods
Euler integration	first order numerical method of solving differential equation
Newtonian	modelled via Newton's equations of motion
Parallel transport	moving along a rod in a twist-free representation
Parallel (computing)	an operation that can be distributed through many CPU cores or CPUs
Equipartition	distribution of thermal energy through different degrees of freedom
Ab inito protein modelling	Finding the conformation of proteins from their sequences using physical models

Table 3: Symbols

$\boldsymbol{x}_i$	Some quantity at index i
$\widetilde{\boldsymbol{x}}_i$	Some quantity at equilibrium
$\boldsymbol{r}_i$	Rod node at index i
$\boldsymbol{m}_i$	Rod material axis at index i (in front of the node)
$\boldsymbol{n}_i$	Rod material axis at index i (in front of the node)
$\boldsymbol{p}_i$	Rod element (also called a segment) at index i (in front of the node)
$\boldsymbol{l}_i$	Rod tangent vector at index i (in front of the node)
N	Total number of nodes
$P(\boldsymbol{m}_a, \boldsymbol{l}_a, \boldsymbol{l}_b)$	Parallel transport material axis at index a from the element at a to the element at b
k_i	Stretching constant for the element at index i
$E_{stretch}$	Stretching energy
κ_i	Local stretching constant at element i
Y	Young's modulus
A	Cross-sectional area
θ	Angle (usually between two material axes)
$\mathrm{arctan2}(y, x)$	$\tan^{-1}\left(\frac{x}{y}\right)$

β_i	Twisting modulus about the node at index i
E_{twist}	Twisting energy
L_i	$\frac{\lvert\boldsymbol{p}_i\rvert+\lvert\boldsymbol{p}_{i-1}\rvert}{2}$
$(k\boldsymbol{b})_i$	Curvature binormal about the node at index i
$\boldsymbol{\omega}$	Centerline curvature 2-vector
$\boldsymbol{B}$	2×2 Bending stiffness matrix
E_{bend}	Bending energy
$\boldsymbol{l}^m$ and $\boldsymbol{m}^m$	Mutual element and material axis
Δt	Timestep
ζ	Friction constant (normally for fluid)
$\boldsymbol{F}_i$	Internal elastic force on node at index i
$\boldsymbol{f}_i$	Stochastic random force on node at index i
ζ_θ	Rotational friction constant
τ_i	Internal elastic torque on material axis at index i
g_i	Stochastic random torque on material axis at index i
E	Energy (often the sum of $E_{stretch}$, E_{bend} and E_{twist})
μ	Dynamic viscosity of solvent
a_i	Radius of rod element at index i
k_B	Boltzmann's constant 1.38×10^{-23}
$\boldsymbol{R}$	Random vector
R	Random number
T	Temperature
$p(x)$	Probability distribution
p_i	Persistence length of element i
$\mathring{A}$	Angstrom $(0.1nm)$

Chapter 1

Introduction

1.1 Study of Hideous and Visceral-Looking Objects

The kinetochore is a supramolecular structure that attaches the chromosomes, containing DNA, to the microtubules of the mitotic spindle. During cell division, the mitotic spindle draws the newly-duplicated genetic material from the centre of the original cell into the two daughter cells. The kinetochore's role is therefore to transfer the force from the spindle microtubules to the chromosomes [22].

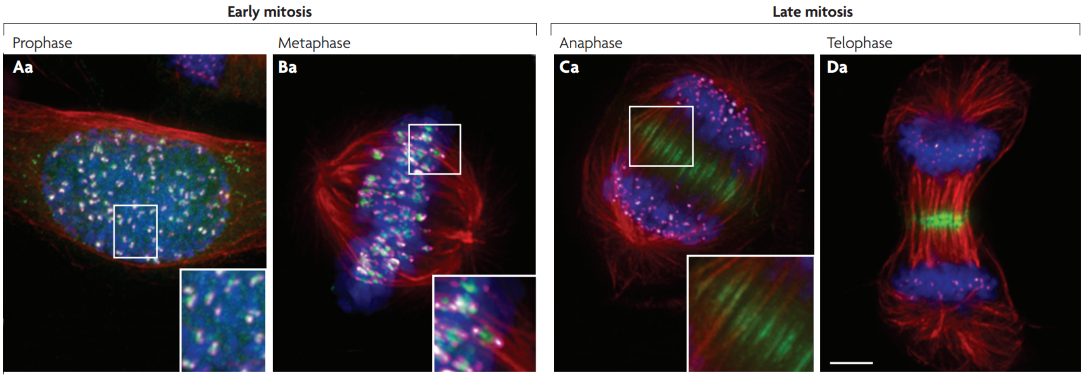
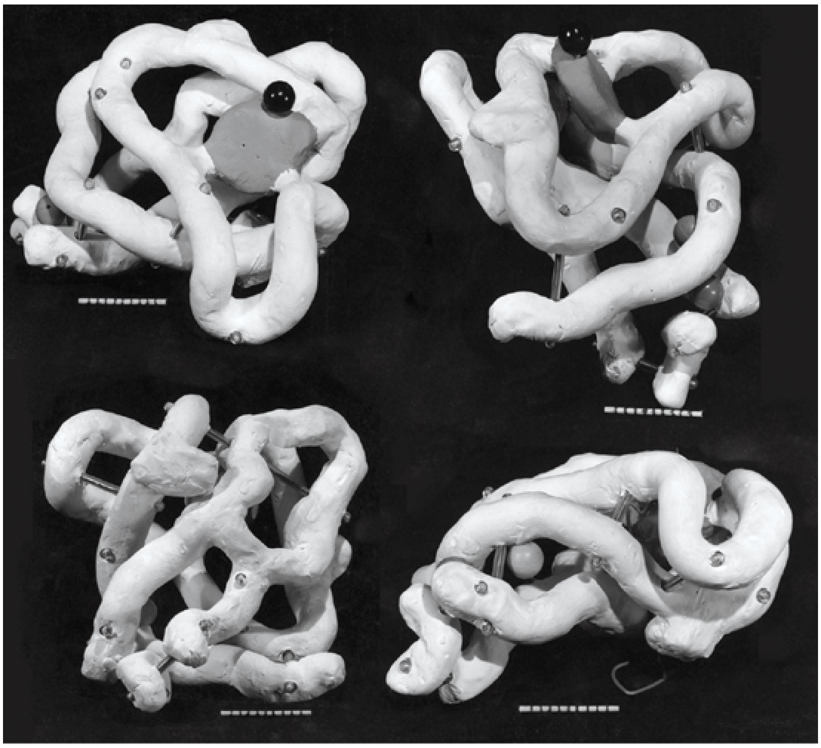


Figure 1.1: Images of cell division taken using indirect immunofluorescence. Kinetochores are stained in pink, microtubules in red, Aurora B kinase in green, and DNA in blue [3].

The first references [23] to the kinetochore in the literature date from the early 20th century. Most papers from this time aren't concerned with structure or function, instead they argue the validity of the term 'kinetochore' itself [24]. At this time, 'kinetochore' was a word describing a hypothetical object that went between the centromere and the mitotic spindle, although not everyone agreed that it should be considered a standalone molecule, instead considering it an extension of the centromere. The kinetochore existed in diagrams as the Platonic ideal of all biological molecules: a circle. Only recently has it been given a structure at all, and yet the structure ultimately poses more questions than it answers.



Nature Reviews | Molecular Cell Biology

Figure 1.2: Low-resolution structure of myoglobin as known in 1958. Polypeptide chains in white, haem group in grey, heavy atoms in black [4].

When X-ray crystallography pioneer Max Perutz first saw the structure of myoglobin (Figure 1.2), he famously remarked “Could the search for ultimate truth really have revealed so hideous and visceral-looking an object?” [25]. The biological community believed that once they had resolved the structure, all the secrets and intricacies of nature would be revealed to them — but none were. Instead, they found themselves staring into an abyss. For molecular biology, no end was in sight. But whether that’s a good thing or a bad thing is really a matter of personal preference. Perutz himself eventually came to see the strange beauty of myoglobin.

The kinetochore seems doomed to the same fate. For a molecule that is seemingly responsible only for attaching two things together, it’s huge — likely in the millions of atoms — and is made up of an array of interconnected protein complexes, arranged in a cylinder, with large globular domains connected to spindly coiled-coils [5]. Its structure

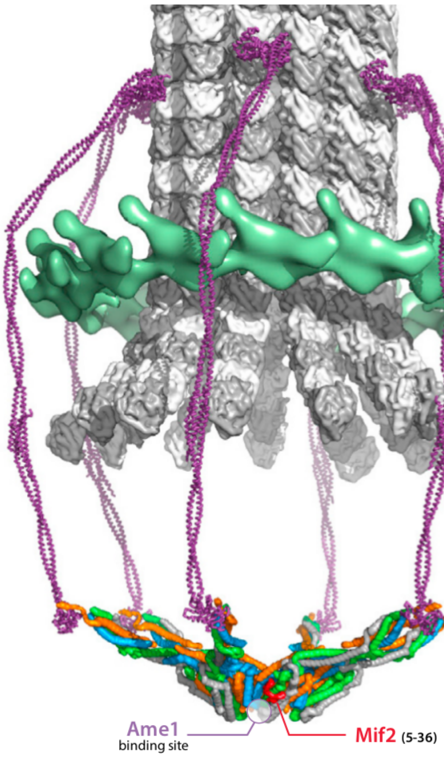


Figure 1.3: Structure of the kinetochore [5]. DAM1 is in green, Ndc80C is in purple, the microtubule is in grey. The structures of the DAM1 subunits are taken from cryo-EM, and the globular domains of Ndc80C are from X-ray diffraction. The rest is hypothesized.

only tells us the extent of what we do not know.

Our understanding of the kinetochore is also hampered by our inability to image it properly. Diagrams such as Figure 1.1 use images from fluorescence microscopy, and the kinetochore is represented, once again, as a dot. Figure 1.3 gives the impression of a structure resolved by cryogenic electron microscopy (Cryo-EM), but while the structures of the various subunits are indeed derived from cryo-EM, this image is actually a composite of many smaller protein complexes, fit to a much lower-resolution electron tomography structure. It's a painstakingly-constructed kinetochore diorama, an ideal of the kinetochore which is clean and orderly and logical. The kinetochore cannot be imaged in the same way that myoglobin was. X-ray crystallography requires the target molecule to be crystallizable, but even the subunits of the kinetochore, such as the

Ndc80 protein complex, do not have resolved crystal structures. Ndc80C may never have a resolved crystal structure, as it's too flexible. Electron microscopy — mostly negative-stain and cryo-EM — fares better [26] [27], but has its own limitations, and introduces its own artefacts.

Yet even a fully-resolved kinetochore structure would not tell the whole story. Before the invention of the video camera, horses in paintings looked wrong. They were commonly painted in a pose called the 'flying gallop', with all four legs outstretched at once. The structure of the horse was well-understood, but its dynamics were a mystery. Horse dynamics were only imaged properly in 1878 by Eadweard Muybridge, who took photographs of a galloping horse named Sallie Gardner and sequenced them in chronological order.

Any form of imaging, be it biomolecular or equine, is limited to producing snapshots of singular moments in time, under certain conditions which make the subjects easy to image. We cannot understand the dynamics of biomolecules from imaging alone, and yet, we cannot simply take more pictures, as Muybridge did. Although statistical methods, from Bayesian methods to machine learning, can be a powerful tool for finding correlations, they cannot provide us with an understanding of the underlying systems [28].

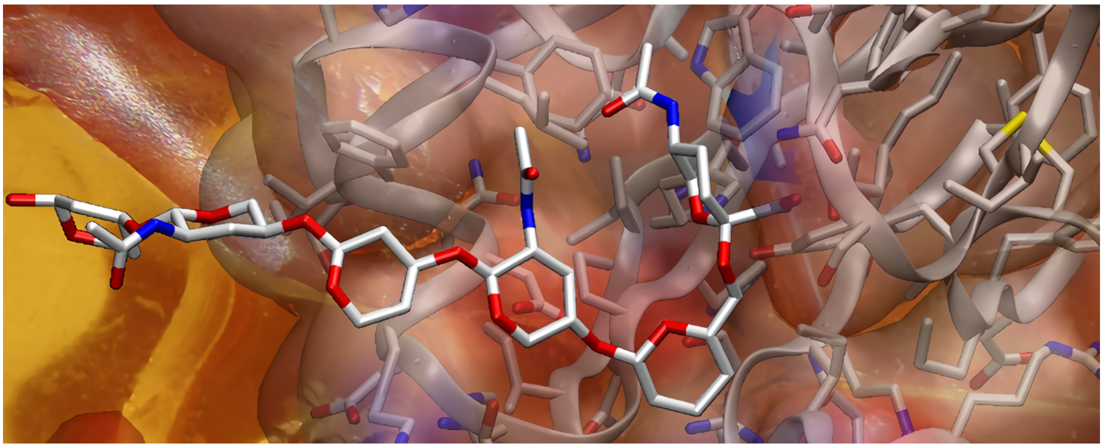


Figure 1.4: Sample molecular dynamics system from the AMBER MD website. Atoms and bonds are represented as cylinders. An abstract representation of the protein secondary structure and electron density profile are overlaid.

With the careful application of some physics, we can generate dynamics from static structural data using computer simulations. Molecular dynamics (MD) (Figure 1.4), and biological simulations in general, are sometimes referred to as a 'computational microscope' [29], because they allow us to see dynamics that we cannot image. These simulations can provide a wealth of new information, and their use in molecular biology has become ubiquitous.

But most of our current biomolecular simulation algorithms were, and still are, designed for small, single-protein systems on nanosecond to microsecond timescales. Supramolecular structures such as the kinetochore are many orders of magnitude larger and longer-lived, far out of the reach of our current computing power. Furthermore, decomposing the problem can only take us so far. These systems, composed of many interacting proteins and protein complexes, have emergent properties that cannot be understood by simulating their constituent parts in isolation [30].

The development of new simulation tools and new mathematical abstractions is essential to understanding and solving the current problems in molecular biology [31]. This work will cover one approach to one aspect of this problem: the modelling of slender biological objects - such as Ndc80C - using elastic rods. In this section, we will discuss the biology of the kinetochore, the limitations of molecular dynamics, the current state of elastic rod models and their suitability to biological systems.

1.2 What and Why is the Kinetochore?

1.2.1 Large-scale Structure and Function of the Kinetochore

When eukaryotic cells undergo mitosis (Figure 1.1), the DNA in the cell nucleus replicates - it splits into two identical copies. Pairs of sister chromatids line up, facing opposite directions, and attach to the microtubules of the mitotic spindle, which draw them from the nucleus of the original cell into the nuclei of the two daughter cells. The process which allows the DNA to separate into two separate copies, and ensures that each daughter cell gets a complete copy of the DNA, is called chromosome segregation. Correct chromosome segregation is crucial to maintaining genomic integrity [32] and thus preventing improper cell division.

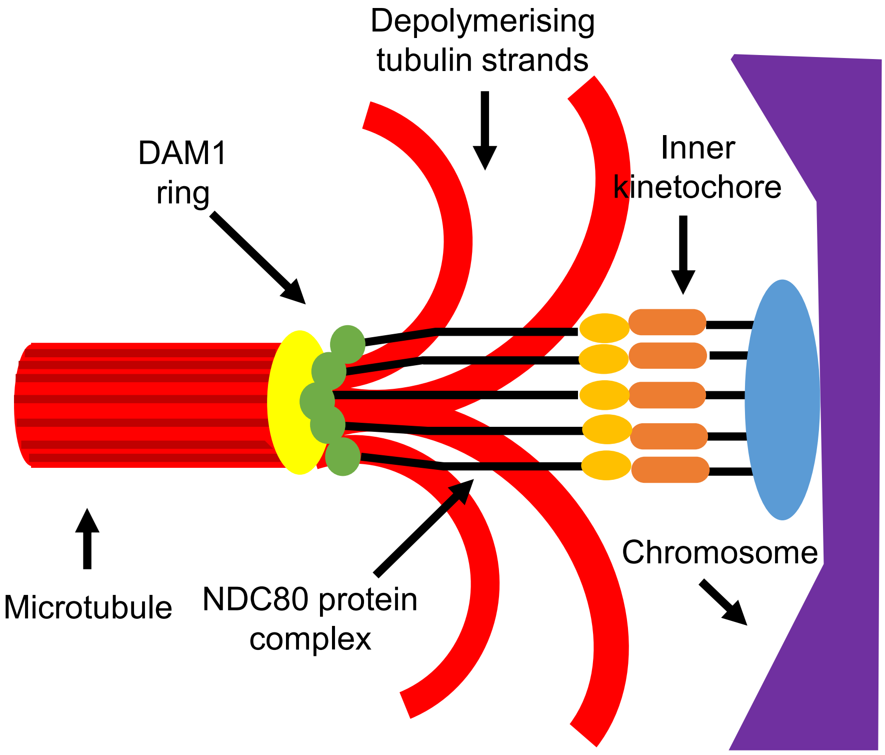


Figure 1.5: Schematic diagram of the kinetochore based on negative stain EM and ET studies. To give an idea of the scale, the Ndc80 protein complex is around 60nm long [6].

The kinetochore's role in this process is to transmit forces between the chromosomes and the microtubules. It is a supramolecular structure formed of three large protein complexes [5] (Figure 1.5). The inner kinetochore attaches to the chromosomes. A concentric ring of long, thin Ndc80 complexes (around 6) attach the inner kinetochore to the DAM1 complex [22]. DAM1 (also called DASH1) is a donut-shaped protein complex which encircles the microtubule [33] [34] [6]. Sections 1.2.2 through 1.2.5 will describe how the kinetochore might attach to the microtubules (which will be studied more closely in chapter 4), how this attachment is regulated, and the role played by the Ndc80C complex in this attachment (which will be studied more closely in chapter 3).

1.2.2 Force Transduction Through the Kinechore in Yeast

The kinetochore is attached to the spindle microtubules in two distinct phases: side-on and end-on. In side-on attachment, the slender Ndc80 protein complex (Ndc80C) binds to the MT surface laterally. Its motion is thought to be assisted by the kinesin-like KAR3, which moves toward the poles during anaphase [35], while the two daughter cells are beginning to take form (see Figure 1.3). At the same time, the microtubule is depolymerising. Live-cell imaging and mutant studies suggest that if the kinetochore cannot move faster than the microtubule depolymerises, it switches to 'end-on' attachment [7] (Figure 1.6).

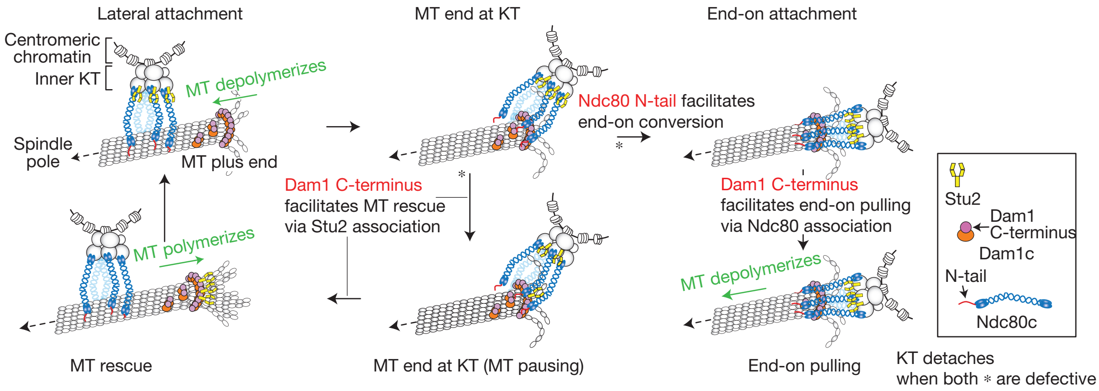

Lateral attachment
Centromeric chromatin
Inner KT
MT depolymerizes
Spindle pole
MT plus end
MT end at KT
Ndc80 N-tail facilitates end-on conversion
*
End-on attachment
Dam1 C-terminus facilitates MT rescue via Stu2 association
*
MT polymerizes
MT rescue
MT end at KT (MT pausing)
Dam1 C-terminus facilitates end-on pulling via Ndc80 association
MT depolymerizes
End-on pulling
Stu2
Dam1 C-terminus
Dam1c
N-tail
Ndc80c
KT detaches when both * are defective

Figure 1.6: Transition of the kinetochore from side-on to end-on attachment [7].

Over the years, many models have been proposed to explain how kinetochores can form stable end-on attachments with depolymerising microtubules. An early explanation was the sleeve model, which suggested that the MT and Ndc80C are held together by an external force of unknown origin [36]. The binding model suggests that there is a binding interaction between DAM1 and the MT that keeps it attached [37]. Perhaps the most compelling is the curling model. The tubulin fibres that make up the microtubule are intrinsically bent outward. FRET studies and negative stain EM have shown that, as it depolymerises, DAM1 and its cargo are forced up the microtubule (Figure 1.7) at 'the speed of depolymerisation' [22] [8]. Hybrid models have also been proposed [38].

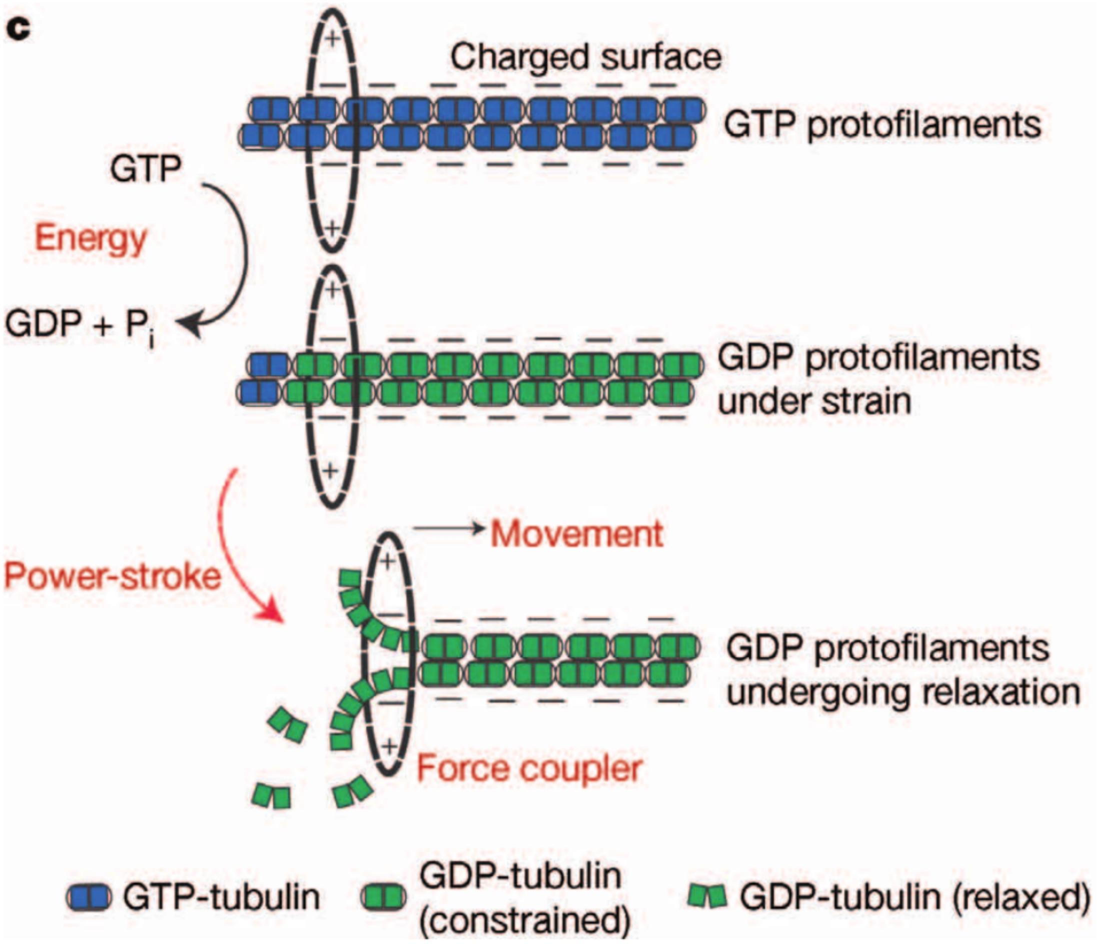


Figure 1.7: Force generation from depolymerising microtubule [8]. The outward bending of the microtubule forces the DAM1 ring away from the depolymerising end.

The precise nature of the interaction between Ndc80C and DAM1 is not well-understood. Ndc80C binds DAM1 at the CH domain — without the CH domain, binding is inhibited. Truncation of the N-terminal domain in Ndc80C is lethal to the cell [39]. There is also some evidence, via negative stain EM, that Ndc80C can bind to two DAM1 rings simultaneously on the same microtubule [40].

1.2.3 Force Transduction Through Human Kinetochores

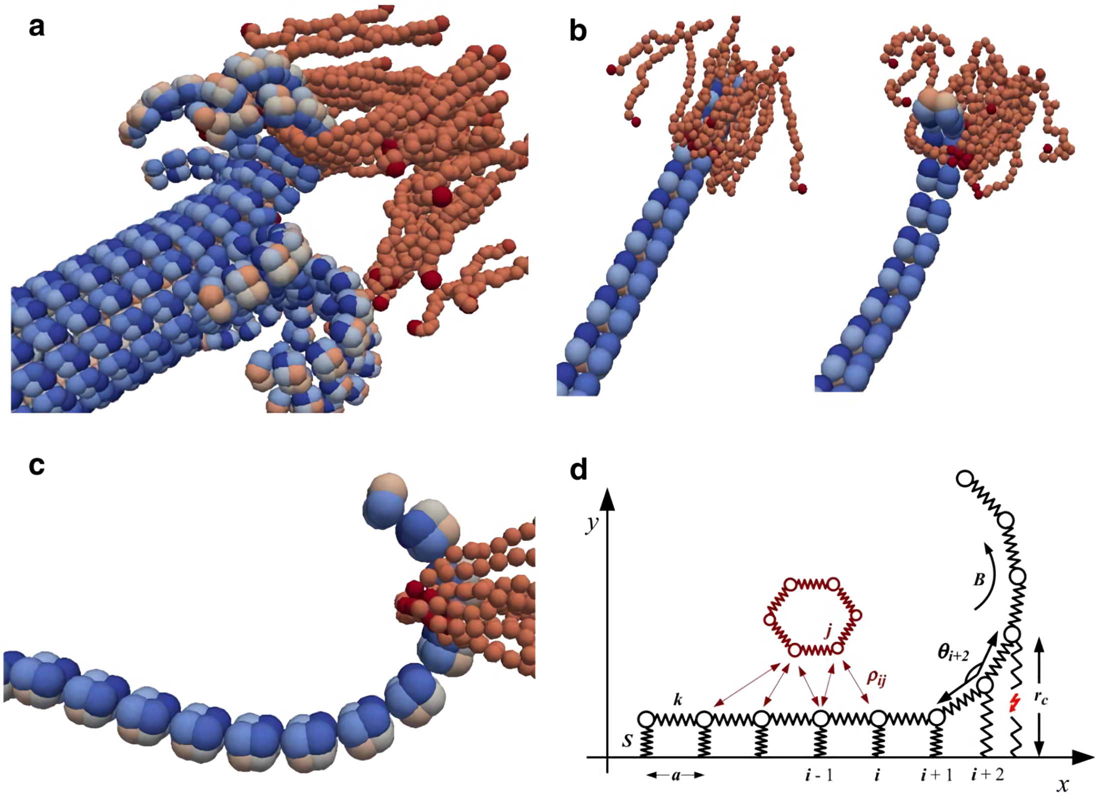


Figure 1.8: Mechanical model of proposed kinetochore-MT attachments. Tubulin and kinetochore fibrils are represented by networks of connected springs [9].

The DAM1 complex is not conserved in humans from budding yeast [41], and end-on attachment in humans may be achieved by the curled microtubules interlocking with multiple Ndc80Cs (Figure 1.8). Ndc80C binds directly to microtubules in both their uncurled and curled states [42], though it binds preferentially to uncurled microtubules [43]. Simulations suggest that the interaction of Ndc80C with the curled microtubules may be a catch-bond. Catch-bonds have unusual mechanical properties — they are strengthened by the application of force, not weakened, like most bonds [44]. As more force is applied, the tubulin and Ndc80Cs are pulled taught, and the bond becomes more stable. In the absence of external forces, the microtubules and kinetochore fibres can disentangle via thermal noise [9].

1.2.4 Error Correction and the Role of Aurora B Kinase

Bi-orientation is the process which ensures that chromosomes line up opposite to each other, and that pairs of duplicate chromosome are drawn into opposite cells. Through mechanical studies, mutation studies and EM imaging, we know that kinetochore bi-orientation is a dynamically regulated process. During bi-orientation, kinetochores are continually attaching to microtubules, and are detached if the interaction is unstable or the attachment is aberrant [7] [45]. This error-correction process is stabilised by tension, which is only present if two kinetochores have successfully bound to opposite chromosomes.

Two mechanisms have been proposed to explain this stabilisation by tension. In the catch-bond mechanism, the application of tensile force changes the conformation of the kinetochore, and the new kinetochore conformation results in a more stable attachment [46]. In the spatial separation mechanism, the force delocalizes Aurora B kinase from the outer kinetochore, and Aurora B dephosphorylates the tails of Ndc80C and Dam1 [47], resulting in a more stable attachment [48] [49]. These mechanisms are not mutually exclusive, they could both play a role in the error-correction process.

1.2.5 Ndc80C Structure and the Loop Region

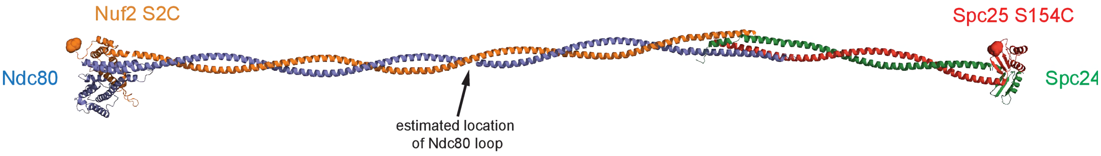


Figure 1.9: Mockup of Ndc80C showing the four chains (Nuf2, Ndc80, Spc25 and Spc24) and their secondary structure [10]. The entire molecule is approximately 60nm in length.

Ndc80C (Figure 1.9) is a long, slender protein complex at the centre of the kinetochore's structure and functions. Most of the structure is made up of a long coiled-coil, a structural motif comprising of two α-helical chains coiling around one another. At each end is a small globular domain, and at the coiled-coil's centre is a short, unstructured loop (see Figure 3.2). This loop is a protein-protein interaction motif [50], and EM imaging suggests that it acts as a molecular hinge, greatly increasing the flexibility of Ndc80C [18]. When Ndc80C bends, the loop straightens out, so Ndc80C can explore highly kinked conformations without stretching bonds. This tightly folded conformation is necessary for Ndc80C's role in the spindle checkpoint process, and mutations in this loop impede the ability of other proteins such as ipl1 and aurora B to disrupt aberrant attachments [51]. In addition, single molecule FRET analysis suggests that the highly bent conformations the loop allows access to can regulate Ndc80C's affinity to bind to microtubules [10].

Ndc80C is also involved in the spindle assembly checkpoint [52], a mechanism which ensures that the chromosomes have segregated correctly [53]. The correct functioning of the spindle assembly checkpoint is regulated by a combination of the proteins involved and the mechanical forces acting upon them [32].

Although end-on attachments are prohibited in mutants which lack the loop, lateral attachments are still possible. The loop may not be necessary for end-on attachment itself, but instead in the transition between the two states, which is less well-understood [54].

1.3 Why Simulating Biomolecules is So Difficult

1.3.1 What Is Atomistic Molecular Dynamics?

All organisms are highly sophisticated atom wigglers [55]. The forces that proteins experience from thermal noise cause them to fold themselves into different configurations (also called ‘conformations’), move between conformations, and interact with one another. Molecular dynamics algorithms apply physics to protein structures in order to make these interatomic interactions occur. Atoms can interact via the Pauli exclusion principle, van der Waals forces, torsion angles, bond length, and sometimes hydrophobicity. In atomistic molecular dynamics, these interactions are typically combined into a classical *force field*, which describes how every atom in a given system interacts with every other atom,

$$
\begin{aligned}
U = \sum_{\text{bonds}} \frac{1}{2} k_b \left(r - r_0\right)^2 + \sum_{\text{angles}} \frac{1}{2} k_a \left(\theta - \theta_0\right)^2 + \sum_{\text{torsions}} \frac{V_n}{2} [1 + \cos(n\phi - \delta)] \\
+ \sum_{\text{improper}} V_{imp} + \sum_{\text{LJ}} 4\epsilon_{ij} \left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}} \right) + \sum_{\text{elec}} \frac{q_i q_j}{r_{ij}}
\end{aligned}
. \qquad (1.1)
$$

where r are bond lengths, θ are bond (bending) angles, ϕ are bond torsion angles, V_{imp} represents the energy contributions due to improper dihedrals, ρ is the minimum of the Lennard-Jones potential, q_i and q_j are charges, and r_{ij} is the interatomic radius.

This energy equation is used to calculate the forces acting on atoms, which are in turn used to resolve the dynamics of those atoms [56]. The dynamical equations are found using the Euler method, a simple, first-order numerical integration scheme for finding the solutions to these equations at a set of equally-spaced points in time (timesteps).

Successive generations of molecular dynamics algorithms have benefited greatly from Moore's law. The first ever molecular dynamics simulations were performed by Fermi, Pasta, Ulam and Tsingou on the vacuum tubes of the MANIAC I computer [57] in 1955. The first molecular dynamics simulation of a protein dates from 1975 [58] - a short simulation of a simplified bovine pancreatic trypsin inhibitor (PTI). Current force fields and implementations of atomistic molecular dynamics algorithms for biological systems, such as GROMACS [59], AMBER [60] and CHARMM [61], were developed in the 1990s. They targeted systems similar in size to PTI — small, singular proteins, normally of order hundreds to thousands of atoms in size, with femtosecond-scale timesteps.

These algorithms have spent decades comfortably producing a prodigious volume of information about biomolecules with structures resolved by X-ray diffraction. But there is an increasingly large category of problems that atomistic molecular dynamics *cannot* solve — systems at the molecular mesoscale, comprised of tens or hundreds of proteins, with dynamics that are very long-lived. Molecular dynamics algorithms weren't designed to scale to multi-million atom systems, or millisecond-plus timescales, let alone both.

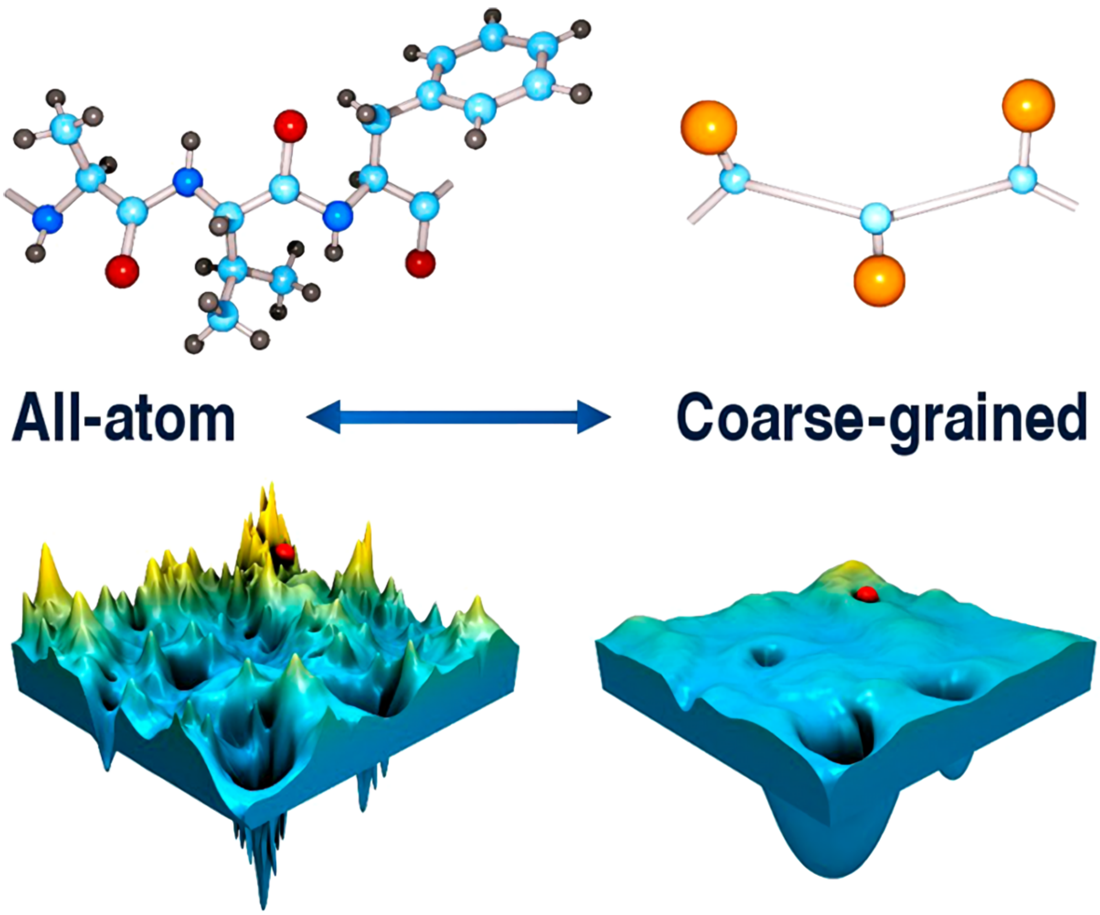


Figure 1.10: Top: All-atom protein representation vs Levitt-Warshel discrete coarse-grained representation, bottom: energy landscape associated with those potentials [11].

The most common solution to this problem has, in the past, been ‘coarse-grained’ molecular dynamics (Figure 1.10), which combines groups of atoms into a larger, compound objects. Examples of such systems include MARTINI [62], UNRES [63] and CABS [64]. At larger length scales, LAMMPS [65] is commonly used instead. These systems, however, have both philosophical and practical limitations. On a practical level, these systems increase the size of discontinuities in systems which are large enough to be approaching continua. On a more philosophical level, some of these tools are designed such that, eventually, atomistic resolution can be recovered. Is that really a truly ‘multiscale’ simulation?

It's also more difficult to generate crystal structures of large, flexible, slender objects. Many components of the Kinetochore, including Ndc80C, do not have fully resolved structures at atomistic resolution. As the time and length scales of current biological problems have increased, biologists and biophysicists have turned, increasingly, to electron microscopy — in particular, cryogenic electron microscopy (cryo-EM). Cryo-EM is a technique that can probe the structures of molecules flash-frozen with liquid ethane or propane. It has allowed for the resolution of almost 10,000 new macromolecules [66], and submissions to the Electron Microscopy Data Bank are still increasing, year-on-year (Figure 1.11).

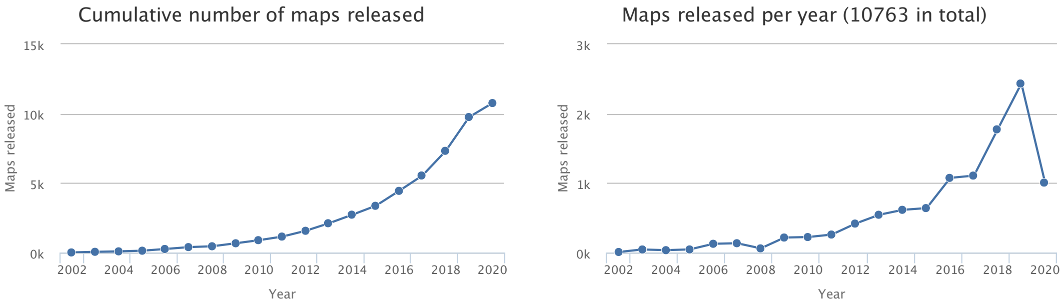


Figure 1.11: Cumulative (left) and per-year (right) submissions to the EMDB since 2002 [12].

1.3.2 What Are Continuum Models?

The proliferation of EM data presents a challenge for atomistic MD. Structures resolved by cryo-EM do not have atomistic resolution. They are intrinsically *continuum* objects, electron density maps — probability distributions, indicating where electrons are most likely to be found. There is no way to use cryo-EM data with an all-atom simulation algorithm. This has given rise to the practice of cryo-EM fitting [67] [68] [69], fitting atoms into cryo-EM structures so that they can be used in atomistic MD simulations. But what if atomistic MD wasn't an inevitability? What if we don't need atoms at all? At some point, the distance between atoms, relative to the total size of the system, will become very small, and the system will begin to behave less like a group of discrete atoms and more like a continuum. Using a more abstract representation of the structure could reduce the computational cost of the simulation, and allow for trajectories of larger objects on longer timescales.

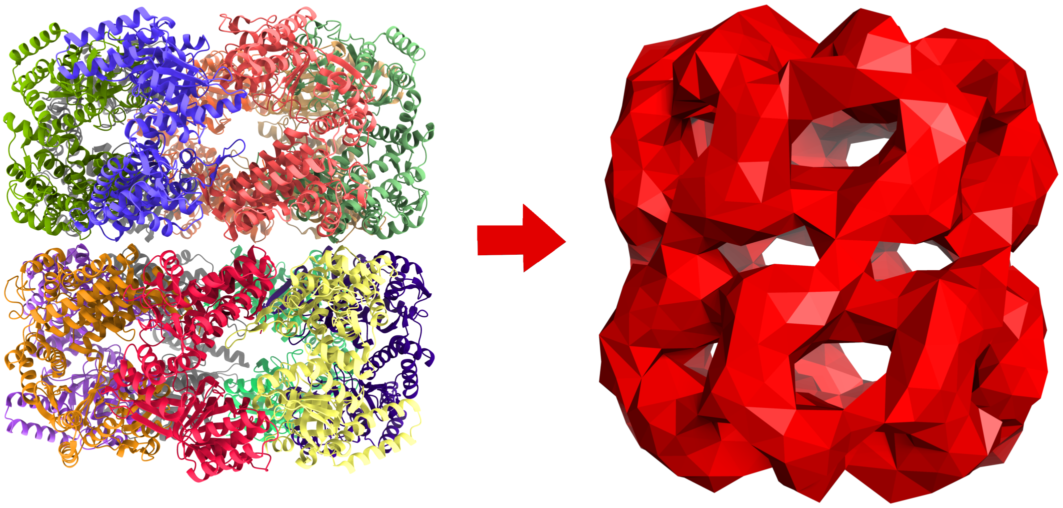


Figure 1.12: FFEA's coarse-grained tetrahedral representation of a biomolecule. Left: all-atom apo-GRoEL with cartoon representation. Right: tetrahedral mesh coarse-grained at $8\mathring{A}$ resolution.

Fluctuating Finite Element Analysis (FFEA) [1] is one attempt at solving this problem. While tools such as LAMMPS represent biological systems as coarse-grained particles, FFEA represents them as viscoelastic (Kelvin-Voigt) continuum objects which are subject to stochastic thermal fluctuations in accordance with the fluctuation-dissipation theorem [70]. These objects are tetrahedral meshes, with each tetrahedron representing many atoms (Figure 1.12). Tetrahedral meshes can be generated both from atomistic structures, and directly from cryo-EM surface profiles.

The FFEA method can best be explained in reference to Cauchy's momentum equation,

$$\rho \frac{D\boldsymbol{u}}{dt} = \nabla \cdot (\boldsymbol{\sigma}^{\nu} + \boldsymbol{\sigma}^{e} + \boldsymbol{\pi}) + \boldsymbol{f}. \tag{1.2}$$

Here, ρ is the density, and $D\boldsymbol{u}/Dt$ is the material derivative of the velocity with respect to time. The behaviour of FFEA objects is described by stresses resulting from the terms on the right-hand side of the equation: $\boldsymbol{\sigma}^{\nu}$ is the stress resulting from the internal viscosity of the material, $\boldsymbol{\sigma}^{e}$ is the elastic stress, $\boldsymbol{\pi}$ is the stress resulting from stochastic thermal noise. Finally, $\boldsymbol{f}$ describes any external force, which can include external viscosity, or short-range interactions such as a Lennard-Jones potential or steric repulsion.

This approach has allowed for long-lived simulations of large, complex systems, such as cytoplasmic dynein [71], myosin-VII [72], rotary ATPase [73] and antibody proteins [74]. However, three-dimensional continuum methods have distinct limitations in dealing with long, thin objects, such as coiled-coils and stable alpha-helices. In these algorithms, the numerical instability is largest in the axis that is smallest — the cross-section of the coiled-coil. This instability is compounded by the addition of thermal fluctuations, which become larger as the axis gets smaller. Therefore, an alternative way to represent long thin structures is needed. This need not be a finite element method — however, it should constitute a coarse-grained, abstract representation of the system, incorporating a 'top-down' parameterisation, as the FFEA algorithm does.

1.3.3 In Defence of Abstractions

The FFEA coarse-graining method makes some people justifiably sceptical. In biological systems, tiny changes to atoms or groups of atoms are magnified into enormous changes in protein dynamics and the functioning of molecular processes. For example, the phosphorylation of Aurora B (section 1.2.4) — a change affecting no more than a handful of atoms — completely alters the dynamics of the kinetochore during bi-orientation [48] [49]. This in turn is crucial to the process of bi-orientation, which is needed to ensure genomic integrity and correct cell division [32]. Small changes like this are called 'post-translational modifications' and they form the foundation of many biological processes. It might, therefore, be reasonable to ask: how could a coarse-grained scheme such as FFEA, or an elastic rod model, possibly capture the kind of information necessary to make all of this work?

The answer is to know what information is important and what information can be discarded, and build a model that allows the important information to persist across length scales. The FFEA algorithm uses, as parameters, data describing conformational changes, the locations of binding sites, and the kinds of potentials they experience [1]. The FFEA representation of conformational changes is an abstraction, but a necessary one. These kinds of trade-offs are not unique to coarse-graining, they are prevalent at every length-scale, including all-atom molecular dynamics, itself an approximation of quantum mechanics.

This extreme sensitivity of biological systems to small changes is also true of Ndc80C. In chapter 3 of this thesis, we will discuss the parameterisation of the elastic rod model from all-atom simulations. We can make the correct abstractions, as long as we carefully consider all assumptions when building the model, and seek to understand the physics and the biology that drives the behaviours of the systems that we wish to simulate.

1.3.4 What Are Elastic Rod Models?

For the purposes of this work, a rod will be defined as an object in 3-d space with a one-dimensional topology, whose radius is much smaller than its length. Mathematical representations of these objects were first fully described by Leonhard Euler, advancing on the work of James Bernoulli, in his elastica theory, in 1744 [75][1]. The elastica theory is, in many ways, an advancement on older theories such as Euler-Bernoulli beam theory. Elastica theory generates exact solutions for minimizing the curvature (and hence, the elastic energy) of three-dimensional curves. This model does not allow for twisting or stretching, and the number of analytical solutions is limited.

Euler and Bernoulli's ideas were advanced in 1859 by Kirchoff [76]. Kirchoff's theory of finite displacement was, in many ways, ahead of its time, developed without modern mathematical theories of elasticity. Unlike Euler's, this model assigns elastic energy to a curve based on strain, rather than curvature. Additionally, Kirchoff rods have cross-sections that can twist independently of one another, increasing the number of degrees of freedom to two.[2]

[1] This makes the study of elastic rods older than the spinning jenny.

[2] Although many current interpretations rest on the assumption that these infinitesimally thin cross sections are perpendicular to the axis of the rod, and stresses are not conducted through them, Kirchoff's original paper only states that the deformations themselves have to be small [77].

The most fully-formed conception of the elastic rod would be developed in 1909 by the Cosserat brothers, who generalized many of Kirchoff, Euler and Bernoulli's ideas to shells, and derived energies for rods from constitutive laws for stress and strain. Not only was the Cosserat theory able to make use of these ideas from continuum mechanics, it also benefited greatly from the works of Frenet [78] and Serret [79] on adapted, framed curves. This notion of *framed* curves allows for an elegant description of the local deformation of the rod, which will be discussed more in section 2.1. It should be noted that these rod theories were not created in a vacuum, and nor did the next theory directly supplant the previous one. Many aspects of Cosserat rod theory, particularly framed curves, made their way into Kirchoff rod theory, and both approaches are widely used today. Section 1.3.5 will discuss these approaches in more detail.

1.3.5 Examples of Elastic Rod Models in Biology

Existing rod models can be broken into three main categories; microscopic models for coiled-coils, microscopic models for DNA, and macroscopic models of rod-like systems such as ropes, hair and cords. Each of these approaches has its own strengths, weaknesses and assumptions. Coiled-coil and DNA models are not necessarily looking at the same kinds of forces and dynamics that we are interested in, so we will undertake a broader view, and examine general-purpose rod models as well.

Coiled-coils

Dynamic coiled-coil elastic rod models are not prolific in the literature. Most of the systems being studied are of time and length scales such that all-atom molecular dynamics is more appropriate. Additionally, in molecules such as Ndc80C, more importance is placed on binding and biochemical interactions than biophysical ones. Therefore, the role of the elastic rod model in this scenario is more of a supplementary one.

Linka et al. [80] use a Cosserat representation to create a constitutive damage model for double and triple-helical collagen fibrils. A triple-helical coiled-coil model is developed, and used to examine collagen fibrils as they ravel and unravel due to external stretch forces. This model does not offer a dynamic simulation, but rather a prediction of these stress/strain relationships and their implications on damage to the collagen.

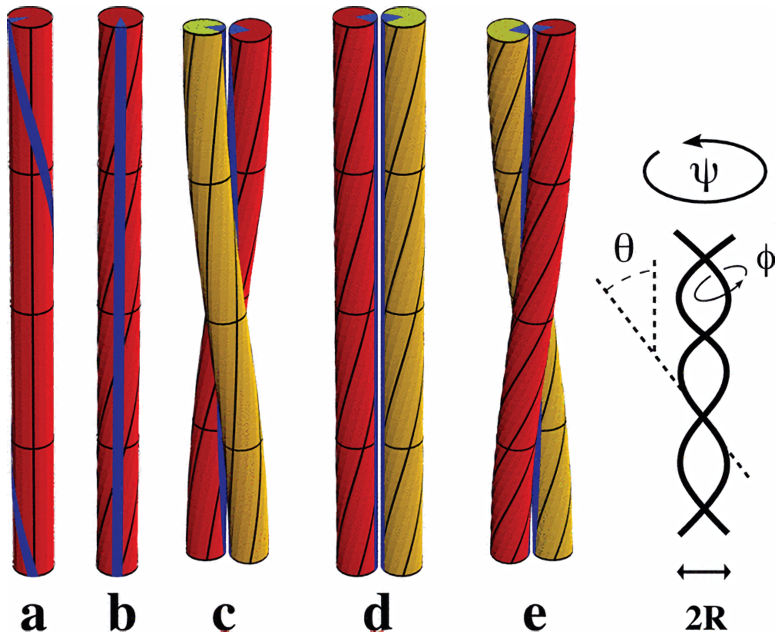


Figure 1.13: Interface curves predicted for rods with hydrophobic strips as in [13]. Hydrophobic strip shown in blue.

Similarly, Neukrich et al. [13] propose a model that predicts the structure of coiled-coils based on self-contact between the two strands at the interface curve. This model features isotropic and homogeneous bending and twisting, but also offers analytical solutions for the physical parameters of the coil (such as radius and pitch) (Figure 1.13). Wang et al. [81] use Cosserat rods to predict the relationship in coiled-coils between strain and chirality, and profile the stretch-twist and bend-shear coupling deformations.

Prior et al. [82], although focusing on helical birods and not specifically coiled-coils, develop a Kirchoff rod model with a similar purpose in mind - finding the bending and torsional response of helical birods from the bending and torsional responses of the constituent rods.

DNA models

DNA rod models have many similarities with coiled-coil models. DNA can also be described by bending, twisting and stretching stiffnesses, although these are normally considered homogeneous. However, DNA models are complicated by circularisation, major and minor grooves, and supercoiling.

Swigon et al. [83] use an elastic rod model to predict the structure and DNA-histone binding energies of DNA minicircles, while Smith et al. [84] use Cosserat rods to predict the onset of supercoiling with applied tension. Yang et al. [85] apply finite element analysis to resolve the equilibrium configurations of rods for DNA supercoiling.

Diverging from the realm of structure prediction, Purohit et al. [86] use elastic rod models in order to predict the formation of plectonemes in twisted, fluctuating DNA, due to torsional buckling. However, rather than using explicit dynamics, this model instead opts for an implicit representation of thermal fluctuations, based on the average energy imparted to the DNA double-helix by thermal noise.

Goyal et al. [87] formulate a DNA model that includes inhomogeneous, anisotropic Kirchoff rods, with a finite difference solution. They also introduce chirality, which couples the tension and torsion of the rod together. This model requires fixed boundary conditions, so although dynamical simulations are performed, these aren't subject to thermal noise, instead they study the response of clamped rods to compression and twisting, particularly the formation of loops ('hockles') and other dynamic instabilities.

Skoruppa et al. [88] study the effect of groove asymmetry on DNA. This model is effectively a twistable worm-like chain, featuring twisting, bending and twist-bend coupling. Skoruppa et al. investigate the effect of this coupling on DNA with both symmetric and asymmetric grooves. Not only is this the one of the rare biological models with explicit dynamics, it contains a relatively in-depth look into parameter extraction, a topic which will be the focus of chapter 3.

1.3.6 Macroscopic Elastic Rod Models and Their Applications

Macroscopic rods are also used to represent ropes, cables and hair. They encompass a broad range of disciplines, from the study of knots, to engineering, to visual effects. These models do not contain thermal noise, are typically more focused on dynamics than structure prediction, and utilize second-order equations of motion for single rods, rather than explicit coiled-coils.

Dynamic elastic rod models became viable as computer models in the 1990s, with an early example being Goldstein et al. [89], who developed a model for the folding dynamics of stiff polymers. They found new and elegant applications for framed curves in the development of a dynamical model which closely resembles a highly developed version of Euler's elastica. The use of the twist-free Bishop frame simplifies the calculation of the dynamics and energies.

Pai et al. [90] developed STRANDS as a computationally-inexpensive elastic rod model for use in surgical simulations. STRANDS is a Cosserat rod model designed to run in real-time on a modest computer. It accomplishes this by reducing the elastic rod dynamics to a boundary value ODE, but unfortunately, this only works for a rod which has a fixed position at one end and a known stress at the other.

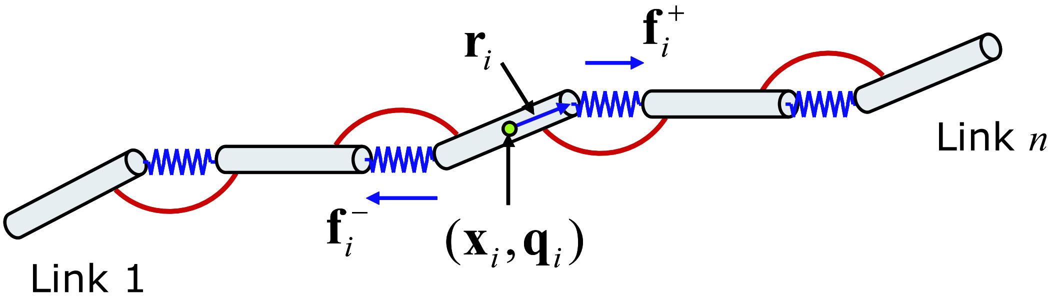


Figure 1.14: Choe et al.'s model [14] features a series of rigid bodies connected by joints that stretch and bend with harmonic potentials.

Simulations of elastic rods are common in Visual effects (VFX), as they can be used to create procedural animations for use in movies and TV. These models offer many interesting and novel solutions to traditional rod problems. For example, Choe et al. [14] (Figure 1.14) present a model primarily developed for hair and foliage, a hybrid of rigid body systems and elastic rod models. This is also one of the few elastic rod models with an explicit radius. Another VFX model, Hadap et al. [91], models rods as ‘strands’, chains of rigid segments connected by spherical joints. Unlike Choe’s model, this one includes a reduced co-ordinate formulation and implicit integration scheme, making it fast for stiff elements with large timesteps. Like many VFX models, however, it is more tuned for visual quality and speed than for physical accuracy.

Recently, a number of elastic rod algorithms have been developed which are general-purpose, feature-rich, and still fast enough to run for the timescales required by the dynamics of the Kinetochore.

CoRdE [92] is a dynamic Cosserat rod simulation employing a discretisation based on finite element methods and Lagrange equations of motion. It supports bending and twisting of the rod, but it is assumed to be inextensible, although few assumptions are made on that basis. The rod is represented as a series of linear nodes and elements, where the element’s rotation is described by a quaternion attached at the midpoint of the centreline. The dynamics use a second-order Lagrangian equation of motion which captures both internal friction and external viscosity, solved implicitly.

Theeten et al. [15] propose a spline-based representation of the centreline, including Kirchoff-derived stretching, bending and twisting energies, with both reversible and irreversible deformations. Unlike other models, the curve is considered to be intrinsically continuous, and the dynamics are calculated based on the energies associated with elements sampled from the spline at fixed intervals. This results in an interesting hybrid of first and second-order rod elements (Figure 1.15).

Finally, perhaps most straightforward comes from Bergou et al. [93]. They define a Kirchoff elastic rod model featuring bending and twisting energies, with the capacity for intrinsic bending and twisting, and a fixed element length, enforced by a constraint but not assumed. A second order Newtonian equation of motion is used, with Euler-integrated timesteps. The rod is constructed out of linear nodes and elements,

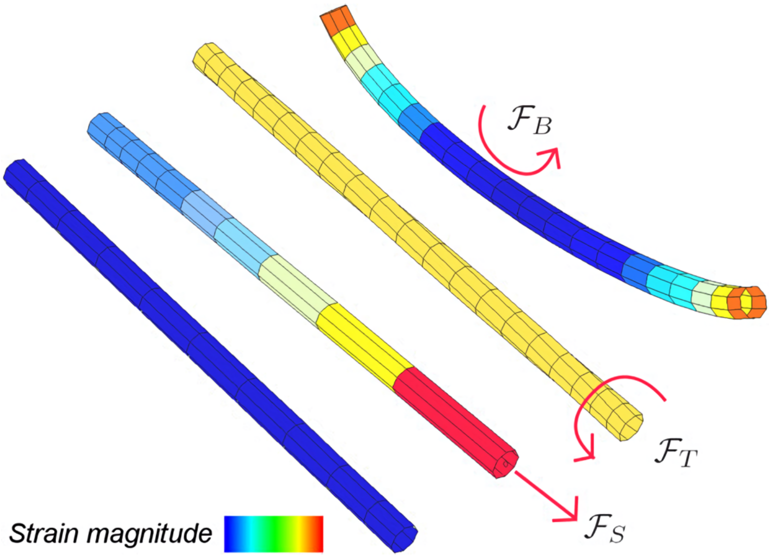


Figure 1.15: Spline-based representation of elastic rod under strain using Theeten et al.'s model [15]. Left-to-right: rest, stretching, twisting, bending.

represented by both the Frenet-Serret and Bishop frames, with the former being used to represent the twisting of the rod, and the latter being a twist-free representation. This model permits both anisotropy and inhomogeneity in the material parameters, although bending about a single node is not possible.

Chapter 2

KOBRA Algorithm and Validation

2.1 Formulation and Assumptions

How do you mathematically represent an elastic rod in three dimensions? This is perhaps one of the few questions that does not torment modern-day elasticians. With few notable exceptions (e.g. [14]), the modern-day conception of elastic rods is based on *framed curves.* For any point on a three-dimensional curve, a frame exists which both characterises the state of the curve at that point, and provides an orthonormal set of vectors that function as a basis for mathematical operations at that point. Figure 2.2 shows how these curves, and their frames, will be represented in this thesis.

There are other questions regarding the formulation of the model with less easily-defined answers. There is no single 'do-everything' rod model. Every model in the literature is designed with a particular purpose, and facilitating that purpose inevitably leads to compromises. At the same time, adding unnecessary complexity can hinder the development process and increase the computational cost of the model. Which features are the most relevant to biological rods, which are the easiest to implement, and which distinguish this model from the hundreds of other elastic rod models out there?

Dynamical or structure prediction? As we have seen, the use of Kirchhoff and Cosserat curves to study coiled-coils is widespread, but this has usually been for an equilibrium structure prediction and not dynamic, non-equilibrium simulations [13], [82]. This means that these methods cannot be re-used to study the dynamics of Ndc80C or the kinetochore.

Macroscopic or microscopic? Most rod models in the literature are designed for macroscopic rods, so thermal noise is not included in their dynamics. In addition, most of those that are molecular structural biology length-scales are interested in structure prediction, so do not incorporate any form of explicit thermal noise. Similarly, macroscopic models normally include mass and inertia [15] [93] [92], but the effects of inertia are negligible when considering biological molecules submerged in a viscous fluid.

Contact and collision detection? Models designed for structure prediction normally have an explicit representation of the protein helices and coiled-coils [13], rather than an implicit one, and often focus on forces between the two coils and finding the equilibrium conformation. Other self-contact models are built primarily for simulating knots [92], so the focus is on resolution rather than speed. Representations of other short-range interactions, such as the Lennard-Jones potential, are scarce.

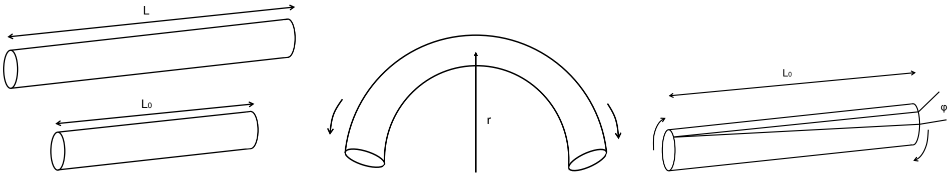


Figure 2.1: Degrees of freedom available to an elastic rod. Left: stretch. Mid: bend. Right: twist.

How many degrees of freedom (Figure 2.1)? The most basic models only include bending stiffness [89]. For biological molecules such as Ndc80C, bending stiffness is required, but it must also be anisotropic and inhomogeneous, in order to correctly replicate the properties of the hinge region. Although the twisting of the rod is less important to Ndc80C's dynamics, the twisting of the rod will affect the axis about which the rod bends, and so it must be represented as well.

Extensible or inextensible? Although, some important biopolymers (microtubules, for example) are effectively inextensible at this scale, this limit is not straightforward because the inextensibility constraint requires that the random forces must be confined to space in which the system motion is constrained. Algorithms for achieving this have been developed by Grassia and Hinch [94], Morse [95] and Liverpool [96]. Furthermore, smaller biopolymers such as coiled-coils can be approximated as harmonic springs, for small extensions [97].

Non-straight equilibrium? Many biological structures, including Ndc80C and tubulin, feature intrinsically-bent equilibrium structures. However, the idea of 'intrinsically bent equilibrium' can be applied to all degrees of freedom. However, as twisting is relative, it may be advantageous to set up systems so that the equilibrium is always untwisted.

Implicit or explicit equations of motion? Many current algorithms favour implicit equations of motion in order to increase speed and numerical stability for stiff rods [92] [91]. This approach can prove difficult for rods on this scale, as rods subject to thermal noise have stochastic equations of motion, for which implicit solutions are harder to obtain.

Implicit or explicit centreline? Some models favour an explicit representation of the centreline as a sequence of nodes and elements, whilst others use an abstract, reduced co-ordinate system representation [91]. These different approaches characterise the delicate balance between mathematical elegance, utility, complexity, and ease of implementation. Ultimately, an explicit centreline allows for easier integration with FFEA, particularly for the purposes of rod-blob interfaces and short-range interactions.

2.2 Definition of the Elastic Rod Model

The configuration of KOBRA rods is described in terms of the Frenet triad and Frenet-Serret relations illustrated in Figure 2.1. Each rod is represented as a continuous material curve. Each point on the curve has three orthonormal vectors associated with it: the tangent vector, the normal vector, and the binormal vector (shown in Figure 2.2). This is therefore a directed curve.

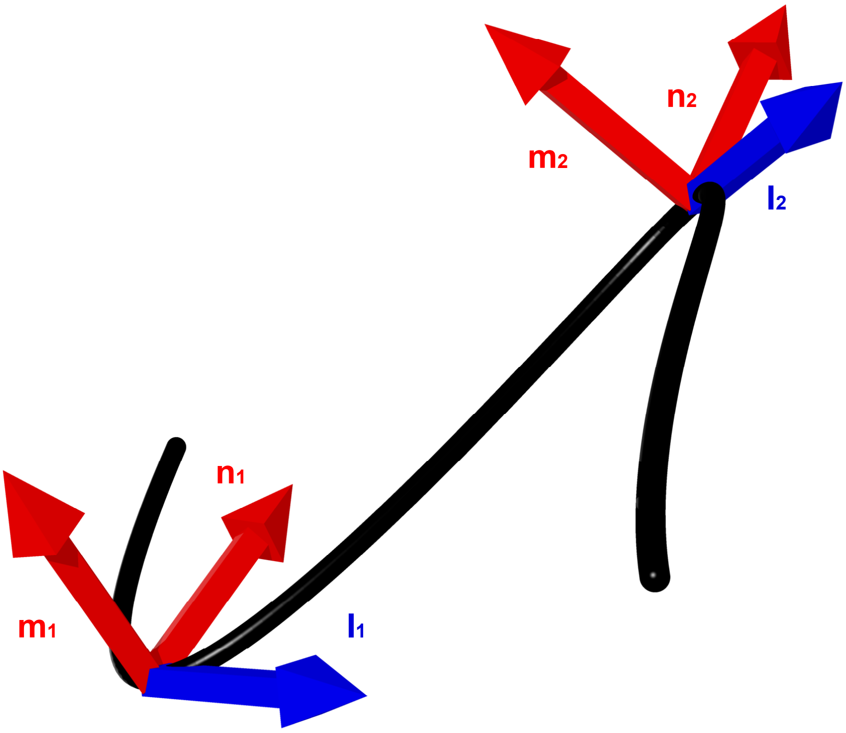


Figure 2.2: An example of a continuous framed curve. The tangent, $\boldsymbol{l}$, and material axes represented by the normal vector $\boldsymbol{m}$ and binormal vector $\boldsymbol{n}$ are shown at two points on the curve.

Although the rod model presented in this chapter is derived from a Kirchoff model, the discretisation and dynamics are different to most Kirchoff models. We opt for an approach that favours a flexible parameterisation and increased computational efficiency.

2.2.1 Construction and Notation

We construct a discretisation of the curve shown in Figure 2.2 as a sequence of extensible straight rod segments connecting a set of discrete nodes (Figure 2.3). The nodes are located at positions $\boldsymbol{r}_i$, where $i \in \{0, N-1\}$, so that there are $N-1$ segments, with end-to-end vectors $\boldsymbol{p}_i = \boldsymbol{r}_{i+1} - \boldsymbol{r_i}$, and the unit vector along the rod $\boldsymbol{l_i} = \hat{\boldsymbol{p}}_i = \boldsymbol{p}_i / |\boldsymbol{p}_i|$.

To represent the internal twisting of the rod, we define the material axes, $\boldsymbol{m}_i$ and

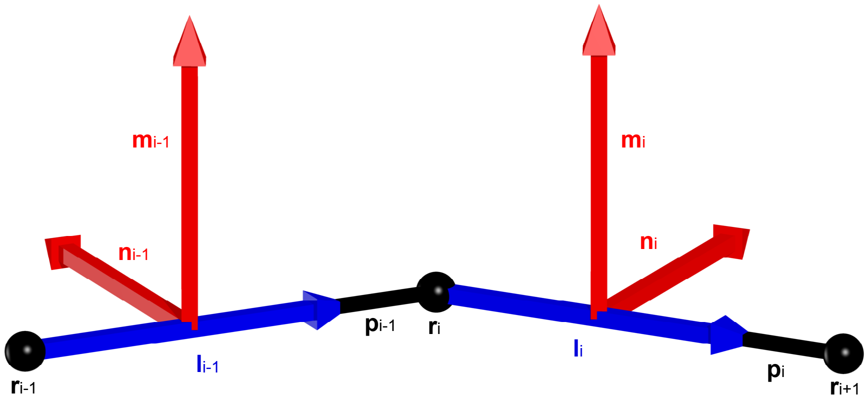


Figure 2.3: Discretisation of a continuous framed curve. The curve is constructed from discrete segments $\boldsymbol{p}_i$ that connect together nodes $\boldsymbol{r}_i$. Each segment has associated material axes ($\boldsymbol{m}_i$ and $\boldsymbol{n}_i$) and a tangent vector ($\boldsymbol{l}_i$).

$\boldsymbol{n}_i$, to be perpendicular to $\boldsymbol{l}_i$, such that $||\boldsymbol{l}_i|| = ||\boldsymbol{m}_i|| = ||\boldsymbol{n}_i|| = 1$ and $\boldsymbol{l}_i \cdot \boldsymbol{m}_i = \boldsymbol{m}_i \cdot \boldsymbol{n}_i = \boldsymbol{n}_i \cdot \boldsymbol{l}_i = 0$. The choice of initial direction of $\boldsymbol{m}_i$ and $\boldsymbol{n}_i$ is arbitrary except that they must obey these constraints. However, these vectors are used to represent the relative twisting of rod elements, and form the basis of a local co-ordinate system that rotates with the rod. As the local material properties rotate with the rod, the directions of $\boldsymbol{m}_i$ and $\boldsymbol{n}_i$ are also used to define the locally anisotropic material properties. A method to initialise a rod that is untwisted at equilibrium is provided in equation (3.4).

Finally, to describe rods with an arbitrary equilibrium state, we define equilibrium values $\widetilde{\boldsymbol{p}_i}$ and $\widetilde{\boldsymbol{m}_i}$, the configuration from which the rod is deformed.

This discretisation is based on the discretisation from Bergou et al. [93], but substantial modifications have been made in order to better support modelling of biological molecules: the addition of extension energy, the removal of inertia, the addition of viscous drag, the addition of thermal noise, changes in the bend energy formula to support hinge regions, and changes to the twist energy formula to support arbitrary equilibrium twisting. These differences will be discussed in greater detail in section 2.2.3.

2.2.2 Parallel Transport

We use the concept of parallel transport to distinguish between changes in the orientation of the material axes that arise from the curvature of the centre line and those associated with twist about the centre line. The easiest way to find the twist between two sets of material axes is via 'parallel transport' of the material axes of one element onto the other.

For two unit vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ we can use a specific form of Rodrigues' rotation formula (see equation (2.32)). [98] to construct a rotation matrix[1] $\boldsymbol{R}$ that rotates $\boldsymbol{a}$ onto $\boldsymbol{b}$,

$$\boldsymbol{R}(\boldsymbol{a},\boldsymbol{b}) = \boldsymbol{I} + [\boldsymbol{v}]_\times + [\boldsymbol{v}]_\times^2 \frac{1}{1+c}, \tag{2.1}$$

where

$$\boldsymbol{v} = \boldsymbol{a} \times \boldsymbol{b} \tag{2.2}$$

$$c = \boldsymbol{a} \cdot \boldsymbol{b} \tag{2.3}$$

and

$$[\boldsymbol{v}]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}. \tag{2.4}$$

To parallel transport the material axes of the ith segment onto the $(i+1)$th segment, we construct the rotation matrix $\boldsymbol{R}$, which rotates the normalised segment $\boldsymbol{l}_i$ onto the normalised segment $\boldsymbol{l}_{i+1}$ [99]. We then apply that matrix to the material axis $\boldsymbol{m}_i$ to obtain the vector $\boldsymbol{m}_i'$,

$$\boldsymbol{m}_i' = R(\boldsymbol{l}_i, \boldsymbol{l}_{i+1}) \cdot \boldsymbol{m}_i = P(\boldsymbol{m}_i, \boldsymbol{l}_i, \boldsymbol{l}_{i+1}). \tag{2.5}$$

The process of parallel transport is illustrated in Figure 2.4, where $\boldsymbol{m}_i'$ is the result of the parallel transport of the material axis $\boldsymbol{m}_i$ onto the $\boldsymbol{p}_{i+1}$th segment. This allows the relative rotations of the material axes for element i and $i+1$ to be compared in a manner that removes the bend between the elements.

[1] The uniqueness of this rotation matrix is not assured.

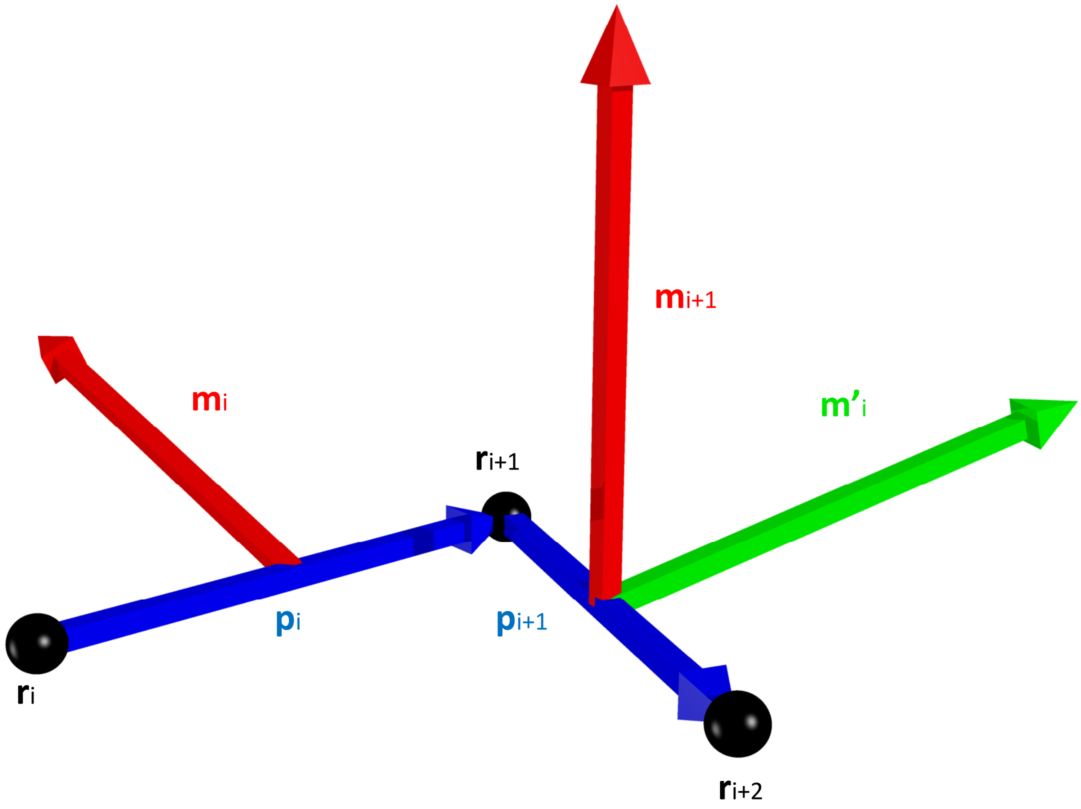


Figure 2.4: Parallel transport of the material axis $\boldsymbol{m}_i$ from the $\boldsymbol{p_i}$th segment to $\boldsymbol{m}'_i$ on the $\boldsymbol{p}_{i+1}$th segment.

2.2.3 Elastic Energy of Deformation

We calculate the forces and torques on the rod from the gradients in the elastic energy arising from deforming the rod away from its equilibrium state. This elastic energy is composed of three components — extensional, torsional, and bending.

Extensional deformation

We assume that the extensional deformation is sufficiently small that extensional elastic energy remains in the linear regime, which, for a coiled-coil, persists for extensions of up to 20% [97]. Thus, the elastic energy of a single segment due to extension (Figure 2.5) is given by

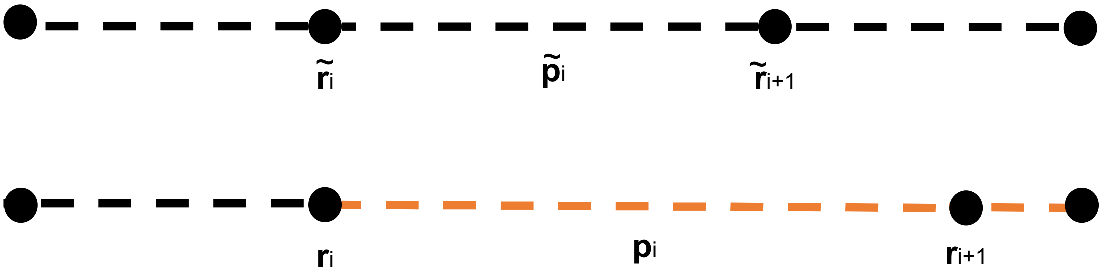


Figure 2.5: Changing the length of an element $\boldsymbol{p}_i$ creates a stretching energy $E_{stretch}$. Note that the stretching energy is a property of an element, and that moving a node results in two elements having different stretching energy (e.g. moving the $\boldsymbol{r}_{i+1}$th node results in changes in the stretching energy from both the $\boldsymbol{p}_i$th and the $\boldsymbol{p}_{i+1}$th node).

$$E_{stretch,i} = \frac{1}{2}k_i(|\boldsymbol{p}_i| - |\widetilde{\boldsymbol{p}}_i|)^2, \tag{2.6}$$

where $|\boldsymbol{p}_i|$ is the length of the ith segment and $|\widetilde{\boldsymbol{p}}_i|$ is its equilibrium length.

The value of the spring constant k_i for the ith element can be written as

$$k_i = \frac{\kappa_{s,i}}{|\widetilde{\boldsymbol{p}}_i|}, \tag{2.7}$$

where $\kappa_{s,i} = YA$, the product of the Young's modulus Y and cross-sectional area A. Note that $\kappa_{s,i}$ is therefore a property of local molecular structure, and so is independent of the rod discretisation, whereas the spring constant k_i depends on the discretised element length.

Torsional deformation

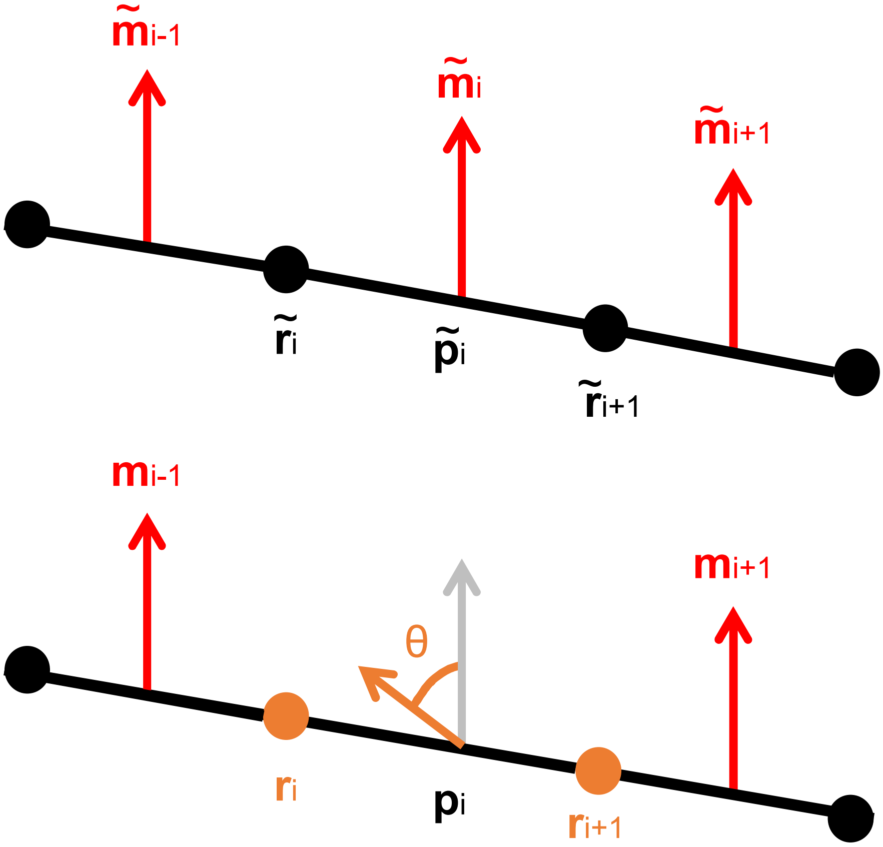


Figure 2.6: The energy of torsional deformation is defined between two elements (about a node). Therefore, applying twist to the material axes $\boldsymbol{m}_i$ and $\boldsymbol{n}_i$ will affect the twisting energy about the $\boldsymbol{r}_i$th and $\boldsymbol{r}_{i+1}$th nodes. The rod is normally not straight (as shown here), so parallel transport (section 2.2.2) is used to ensure that both material axis vectors are in the same basis.

To compute the torsional energy, we need to measure the degree of twist from one rod segment to the next, as shown in Figure 2.6. We define an angle of rotation, $\Delta\theta_i$, between the material frames of the rod segments i and $i+1$, such that

$$\Delta\theta_i = \operatorname{arctan2}\left((\boldsymbol{m}_{i+1} \times \boldsymbol{m}'_i) \cdot \boldsymbol{l}_{i+1}, \boldsymbol{m}'_i \cdot \boldsymbol{m}_{i+1}\right), \tag{2.8}$$

where $\boldsymbol{m}_{i+1}$ is the material axis of the $(i+1)$th segment, $\boldsymbol{m}'_i$ is $P(\boldsymbol{m}_i, \boldsymbol{l}_o, \boldsymbol{l}_{i+1})$, the material axis of the ith segment parallel transported (equation (4.9)) onto the $(i+1)$th segment, and l_{i+1} is the unit vector along the $(i+1)$th segment. We use the $\arctan 2(y, x)$ function, which returns a value of $\arctan(\frac{y}{x})$, in the range $-\pi$ to π by using the signs of the parameters. This allows us to use non-zero equilibrium twist angles while still retaining the correct potential.

Within the linear elastic regime, the torsional energy at node i arises from the rotation between the material axes of the adjacent segments,

$$E_{twist,i} = \frac{\beta_i}{L_i}\left(\text{mod}\left(\Delta\theta_i - \widetilde{\Delta\theta}_i + \pi, 2\pi\right) - \pi\right)^2. \tag{2.9}$$

Here β_i is the torsion constant (analogous to the stretching constant), $\Delta\theta_i$ and $\widetilde{\Delta\theta}_i$ are the angles between the material frames in the current and equilibrium configurations, and

$$L_i = \frac{|\boldsymbol{p}_i| + |\boldsymbol{p}_{i-1}|}{2}. \tag{2.10}$$

Note that equation (2.9) does not account for the number of turns between two elements being > 1, so the discretisation should be chosen so that $\Delta\theta$ is confined to the range $-\pi < \Delta\theta < \pi$. By choosing a periodic function in equation (2.9) we ensure that the energy remains continuous.

Equation (2.9) differs from Bergou et al. [93], who assume that $\widetilde{\Delta\theta}_i$ will always be zero, as the material axis is constructed to have $\widetilde{\Delta\theta}_i = 0$ for each rod element. However, for an intrinsically twisted rod, it is useful define $\widetilde{\Delta\theta}_i$ to be non-zero. If $\widetilde{\Delta\theta}_i \neq 0$ then the angles $\Delta\theta_i$ and $\widetilde{\Delta\theta}_i$ must be signed, hence our use of the $\arctan 2(y, x)$ function above. In Bergou et al. [93], the twist energy is given as

$$E_{twist,i} = \frac{\beta_i}{L_i}(\Delta\theta_i)^2 \tag{2.11}$$

where β is the twist constant, L_i is given in equation (2.10), and the values of $\Delta\theta$ are given by

$$\Delta\theta_i = \cos^{-1}(\boldsymbol{m}_i' \cdot \boldsymbol{m}_{i+1}), \tag{2.12}$$

which is the angle between one material frame and the next, but parallel transported as to ignore the material curvature. This solution is more elegant and faster to compute, but it produces incorrect results for values of $\Delta\theta \neq 0$, shown in Figure 2.7.

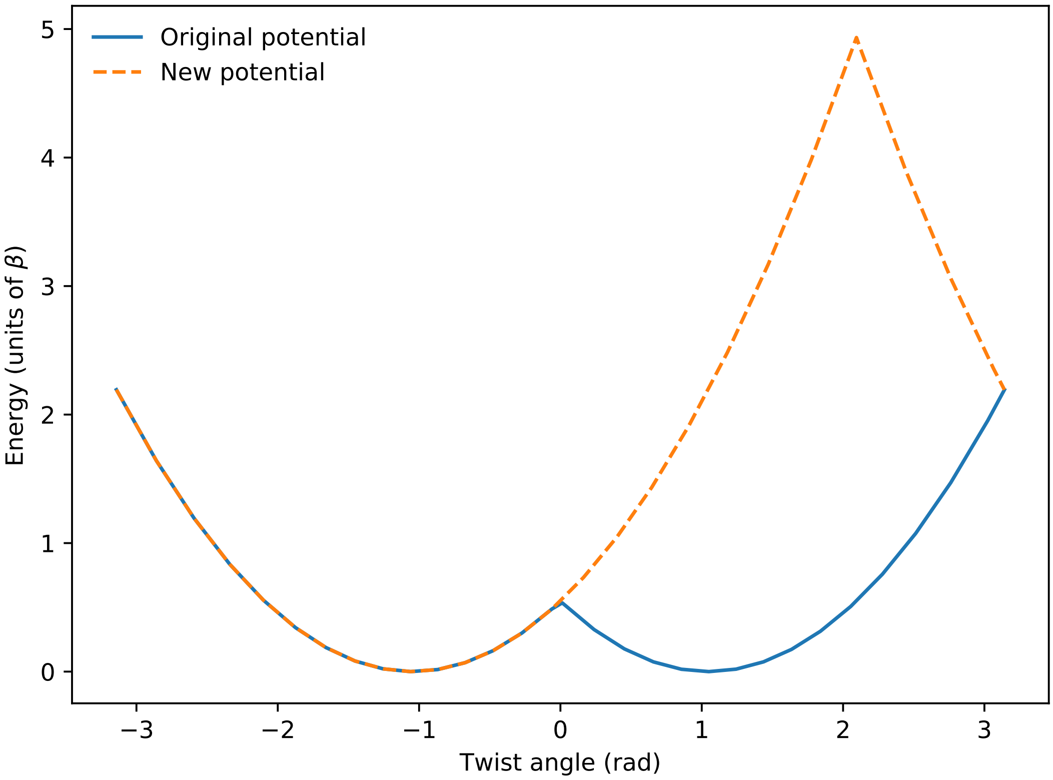


Figure 2.7: Comparison of the harmonic twist energy potentials about a single node with $\widetilde{\Delta\theta} = \frac{1}{3}\pi$, for the old twist energy (Equation (2.11)) and the new twist energy (Equation (2.9)).

The two energy formulae are identical when $\widetilde{\Delta\theta} = 0$. However, the original potential is symmetric about $\Delta\theta = 0$, which produces a potential which incorrectly has two minima and a lower maximum (see Figure 2.7). This is because the inverse cosine in equation (2.12) results in an angle which is not signed, meaning that $\Delta\theta - \widetilde{\Delta\theta}$ will have the same value for both positive and negative values of $\Delta\theta$.

Bergou et al. [93] avoided this problem by specifying that all rods should be parameterised such that, at equilibrium, $\widetilde{\Delta\theta_i} = 0$. In some cases, it is useful to have non-zero equilibrium values (e.g. when defining the bending stiffness matrix) to maintain similar directions for the material axes of adjacent elements. A method for removing intrinsically-twisted rods ('rod unrolling') will be described in section 3.2.

Bending deformation

To obtain the bending energy (Figure 2.8), we compute the curvature binormal, $(k\boldsymbol{b})_i$, a vector which defines the change in orientation between two segments due to bending: $(k\boldsymbol{b})_i$ is orthogonal to both of the segments, with a magnitude proportional to the tangent of half of the angle between them,

$$(k\boldsymbol{b})_i = \frac{2\boldsymbol{p}_{i-1} \times \boldsymbol{p}_i}{|\boldsymbol{p}_i||\boldsymbol{p}_{i-1}| + \boldsymbol{p}_{i-1} \cdot \boldsymbol{p}_i}, \tag{2.13}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i-1}$ are the ith and $(i-1)$th segment vectors. This formula contains a singularity at $\boldsymbol{p}_i = -\boldsymbol{p}_{i-1}$, which corresponds to maximum bending energy, so is encountered very infrequently.

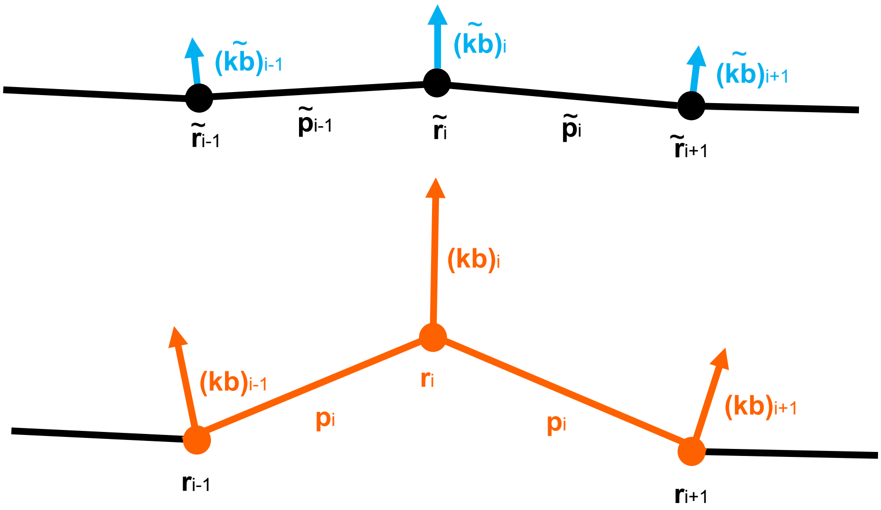


Figure 2.8: A bend about a node increases the length of the curvature binormal $(k\boldsymbol{b})_i$ associated with that node. For an isotropic, untwisted rod, the bending energy is proportional to the square of the curvature binormal. Note that the bending energy is the property of a node, not of an element, and that moving a single node affects the bending energy of that node, and the two adjacent nodes.

For a rod with an isotropic bending stiffness, the curvature binormal $(k\boldsymbol{b})_i$ alone is sufficient to calculate the bending energy. However, for a rod with an anisotropic bending stiffness, we need to resolve the components of bend with respect to the local material axes.

In the algorithm of Bergou et al. [93], this information is encoded in a column 2-vector called the centreline curvature, $\boldsymbol{\omega}$.

$$\boldsymbol{\omega}(i,j) = ((k\boldsymbol{b})_i \cdot \boldsymbol{n}_j, -(k\boldsymbol{b})_i \cdot \boldsymbol{m}_j)^T \tag{2.14}$$

where $\boldsymbol{m}_j$ and $\boldsymbol{n}_j$ are the material axes of the jth segment, and $(k\boldsymbol{b})_i$ is the curvature binormal about the ith node.

Assuming that the material properties of the rod are such that torque and curvature are linearly related, the centreline curvature can be used to define the bending energy similarly to the previous energies — as a quadratic potential centred about $\widetilde{\boldsymbol{\omega}}_{\boldsymbol{i}}$, the equilibrium material curvature,

$$E_{bend,i} = \frac{1}{4\widetilde{L}_i} \sum_{j=i-1}^{i} \left[(\boldsymbol{\omega}(i,j) - \widetilde{\boldsymbol{\omega}}(i,j))^T \widetilde{\boldsymbol{B}}_i (\boldsymbol{\omega}(i,j) - \widetilde{\boldsymbol{\omega}}(i,j)) \right] \tag{2.15}$$

Where $\boldsymbol{\omega}$ is the centreline curvature, L_i is given in equation (2.10), $\widetilde{\boldsymbol{\omega}}$ represents the equilibrium curvature, and $\boldsymbol{B}_i$ is the bending stiffness. In general $\boldsymbol{B}_i$ is a positive definite 2×2 matrix to allow the stiffness to be a function of the direction of bend, such as occurs in a hinge.

As written, equation (2.15) is an average over two bending energies, one in which the curvature binormal $(k\boldsymbol{b})_i$ is projected into one material frame of the segment $(j = i-1)$, and a second in which it is projected into the material frame for segment $j = i$. If these two material axes are in different directions, this will reduce the anisotropy of $\boldsymbol{B}$. Such averaging is undesirable if there is a local ‘hinge’ with strongly localised, anisotropic flexibility.

To allow for the creation of an anisotropic hinge region at a single node, we need to define a material axis at the node. We do this by defining an intermediate segment, centred on this node, from a weighted average of the two segments on either side of this node, called the *mutual segment*. Material axes from surrounding segments are parallel transported onto this mutual segment. This allows us to have a hinge at a single node, but since it comes at a higher computational cost, it is only used for curves with anisotropic bending stiffness.

After computing $(k\boldsymbol{b})_i$ as per equation (2.13), we compute the value of a new, normalised mutual segment centred on the node between the two original segments (Figure 2.9).

We define the unit vector of the mutual segment as

$$\boldsymbol{l}^m = \frac{\boldsymbol{P}^m}{|\boldsymbol{P}^m|}, \tag{2.16}$$

$$\text{where } \boldsymbol{P}^m = \left(\frac{\boldsymbol{l}_{i-1}}{|\boldsymbol{p}_{i-1}|} + \frac{\boldsymbol{l}_i}{|\boldsymbol{p}_i|} \right),$$

and $\boldsymbol{l}_i$ and $\boldsymbol{l}_{i-1}$ are the unit vectors of the segments on either side of the node. We use an inverse weighting of element lengths in the average so that the orientation of the mutual material frame is more affected by the shorter of the two adjacent elements (whose centre is closer to the node).

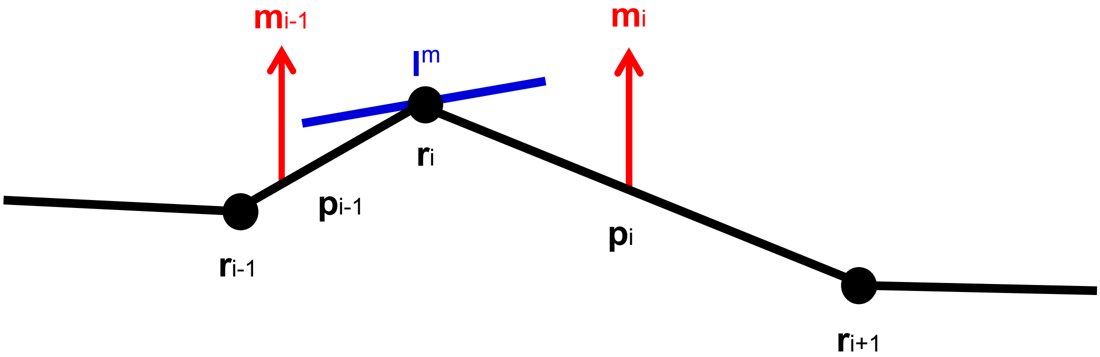


Figure 2.9: A mutual rod segment $\boldsymbol{l}^m$ about the ith node. The segment is weighted more towards the ith node, to which it is closer.

Next, we form the mutual material axis at the node by parallel transporting the material axes of the two adjacent elements onto the mutual segment,

$$\boldsymbol{m}_i^- = P(\boldsymbol{m}_{i-1}, \boldsymbol{l}_{i-1}, \boldsymbol{l}^m), \tag{2.17}$$

$$\boldsymbol{m}_i^+ = P(\boldsymbol{m}_i, \boldsymbol{l}_i, \boldsymbol{l}^m), \tag{2.18}$$

and then by averaging these two material axes, weighted according to inverse element length, and then normalising to give (Figure 2.10)

$$\boldsymbol{m}^m = \frac{\boldsymbol{M}^m}{|\boldsymbol{M}^m|} \tag{2.19}$$

$$\text{where } \boldsymbol{M}^m = \left(\frac{\boldsymbol{m}_i^-}{|\boldsymbol{p}_{i-1}|} + \frac{\boldsymbol{m}_i^+}{|\boldsymbol{p}_i|} \right).$$

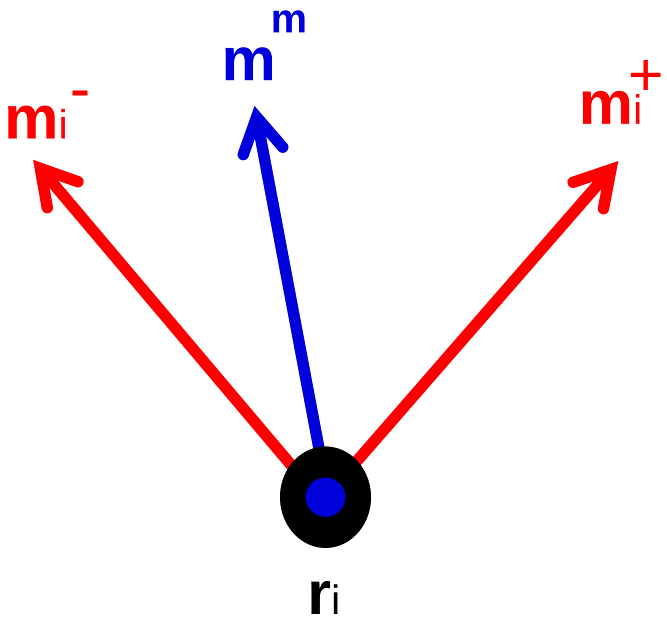


Figure 2.10: A mutual material axis $\boldsymbol{m}^m$ about the ith node. In this diagram, the mutual segment is facing into the page. The mutual material axis is again weighted toward the nearest node.

We now have a weighted average segment and weighted average material axis, valid at a node, not a segment[1] By projecting the curvature binormal into the components of the material axes of the new mutual segment, we obtain the material curvature $\boldsymbol{\omega}$, the amount of bending in the local material axis frame,

$$\boldsymbol{\omega}_i^m = ((k\boldsymbol{b})_i \cdot \boldsymbol{n}^m, -(k\boldsymbol{b})_i \cdot \boldsymbol{m}^m)^T \,, \tag{2.20}$$

where $\boldsymbol{n^m} = \boldsymbol{m^m} \times \boldsymbol{l^m}$, and $(k\boldsymbol{b})_i$ is the curvature binormal about the ith node.

From there, we can simply use the bend energy equation (equation (2.15)) from before, but without the summation between the jth and ith segments,

$$E_{bend,i} = \frac{1}{2\widetilde{L}_i} \left[(\boldsymbol{\omega}_i^m - \widetilde{\boldsymbol{\omega}}_{\boldsymbol{i}}^m)^T \boldsymbol{B}_i (\boldsymbol{\omega}_{\boldsymbol{i}}^m - \widetilde{\boldsymbol{\omega}}_{\boldsymbol{i}}^m) \right] , \tag{2.21}$$

[1] Equations (2.16) and (2.19) both have singularities in the case where $\boldsymbol{m}'_{i-1} = \boldsymbol{m}'_i$ and $\boldsymbol{p}'_{i-1} = \boldsymbol{p}'_i$. The former corresponds to the previously-mentioned singularity in bend energy, the latter corresponds to maximum twisting energy. Like the singularity in bending energy, then, they are rarely-observed.

where $\widetilde{L}_i$ is given by equation (2.10) evaluated for the equilibrium configuration, and $\boldsymbol{B}_i$ is the bending stiffness matrix in the local material axis frame, a positive definite 2×2 matrix.

This formulation does not assume that $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_i$ are in similar directions, so it can account for arbitrarily large bend angles about a single element.

2.2.4 Dynamical Equations

The changes to the configuration of a rod are defined by $\Delta \boldsymbol{r}_i$, the changes of node positions, and $\Delta\theta_i$, which determine the rotations of the material axes orientations $(\boldsymbol{m}_i)$. The translational motion of each node is given by a stochastic equation of the form

$$\Delta \boldsymbol{r}_i = \mathcal{M}_i \cdot (\boldsymbol{F}_i + \boldsymbol{f}_i)\, \Delta t, \tag{2.22}$$

where Δt is the simulation timestep, $\mathcal{M}_i$ is the mobility tensor of the rod segment in the fluid medium, $\boldsymbol{F}_i$ is the internal elastic force, and $\boldsymbol{f}_i$ is the random force from stochastic thermal noise. This equation assumes that the system is overdamped, so that the inertia of the rod is negligible. We also neglect hydrodynamics, but include a resistive (damping) force against a background fluid medium. This is therefore a Brownian equation of motion.

For a general axisymmetric body, the mobility tensor will be anisotropic. However, for simplicity we have approximated the mobility as being isotropic and equal to mobility of a sphere of radius a_i, equal to half the equilibrium length of each segment (see equation (2.10) [100],

$$\mathcal{M}_i = \zeta_i^{-1} \boldsymbol{I} \implies \Delta \boldsymbol{r}_i = \frac{\Delta t}{\zeta} \left(\boldsymbol{F}_i + \boldsymbol{f}_i\right), \tag{2.23}$$

where the mobility,

$$\zeta_i = 6\pi\mu a_i \tag{2.24}$$

where μ is the dynamic viscosity of the medium and a_i the radius of the sphere.

Similarly, for the rotational motion of the material axes,

$$\Delta\theta_i = \frac{\Delta t}{\zeta_\theta} \left(\tau_i + g_i\right), \tag{2.25}$$

where g_i is a random thermal torque, τ_i is the torque due to the internal elastic forces (2.28), and ζ_θ is the corresponding friction constant for rotation. For the rotational friction constant we use the rotational drag on cylinder of radius a and length $|\boldsymbol{p}_i|$,

$$\zeta_\theta = 8\pi\mu a^2 \cdot |\boldsymbol{p}_i|. \tag{2.26}$$

It would be possible to include hydrodynamic interactions by replacing equation (2.22) for each segment with a single equation for the entire rod with a single mobility matrix including the hydrodynamic coupling between the elements. However, this would result in replacing ζ_i in equation (2.30) with the inverse of the mobility tensor, which is computationally more expensive. If the rod remains nearly straight, the effect of hydrodynamic interactions can be approximated through a difference in the local mobility parallel and perpendicular to the rod axis.

For a particular configuration of nodes and material axes, the forces acting on nodes ($\boldsymbol{F}_i$) and the torques acting on material axes ($\boldsymbol{\tau}_i$) can be determined from the partial gradients of the energy:

$$\boldsymbol{F}_i = \frac{\partial E}{\partial \boldsymbol{r}_i}, \tag{2.27}$$

where $\boldsymbol{F}_i$ is the force on the ith node at $\boldsymbol{r}_i$ due to the energy gradient, and

$$\boldsymbol{\tau_i} = \frac{\partial E}{\partial \theta_i}\boldsymbol{l}_i, \tag{2.28}$$

where $\boldsymbol{\tau_i}$ is the torque resulting from the energy gradient of segment i, $\boldsymbol{l}_i$ is the unit vector representing the axis of rotation, and the total internal energy, $E = E_{stretch} + E_{twist} + E_{bend}$, where $E_{stretch}$, E_{twist} and E_{bend} are obtained by summing the node and element contributions to stretch, twist and bend energies from Equations (2.6), (2.9) and (2.21).

Due to the complexity of these formulae we compute the gradient numerically by taking central numerical differences of the energy functions, i.e.

$$\frac{\partial E}{\partial x} \simeq \frac{\Delta E}{\Delta x} = \frac{E(x+\Delta x) - E(x-\Delta x)}{2\Delta x}, \tag{2.29}$$

where x is some co-ordinate (e.g. a component of node position $\boldsymbol{r}$ or material axis rotation θ) in the system, and $E(x)$ is the energy of the system x for a given value of x. This means that, for each index i, we need to compute each of the three energies twice (for $x + \Delta x$ and $x - \Delta x$) in each of the four degrees of freedom associated with that index: x, y, z, and twist.

When a node is translated, or when a material axis is twisted, it changes the energies of the two nodes on either side of it. Figures 2.5, 2.6 and 2.8 display the affected geometry in orange. Therefore, computing the gradient in energy for the translation of node i requires information about a 5-node window of the rod between $\boldsymbol{r}_{i-2}$ and $\boldsymbol{r}_{i+2}$.

Using the drag given by equation (2.23), the force acting on each node from thermal noise in a discrete timestep δt is given by the fluctuation dissipation theorem, [70]

$$\boldsymbol{f}_i = \sqrt{\frac{24 k_B T \zeta_i}{\Delta t}} \cdot \boldsymbol{R}, \tag{2.30}$$

where T is the temperature of the system, ζ_i the friction constant, Δt the timestep, k_B is Boltzmann's constant, and $\boldsymbol{R}$ is a random vector, where R_x, R_y and R_z are independently drawn from uniform distributions in the range $-0.5 \leq R_{x,y,z} \leq 0.5$, such that $\langle \boldsymbol{R} \rangle = 0$ and $\langle R_i R_j \rangle = \frac{1}{12}\delta_{ij}$.

Likewise, the rotational thermal torque is given by

$$g_i = \sqrt{\frac{24k_B T \zeta_\theta}{\Delta t}} R \tag{2.31}$$

Where T is the temperature of the system, ζ_θ is the torsional friction, Δt is the timestep, k_b is Boltzmann's constant, and R is a uniform random variable in the range $-0.5 \leq R \leq 0.5$.

To apply the rotation $\Delta\theta_i$ to the material axis, we use Rodrigues' rotation formula [98],

$$\boldsymbol{v}_{\text{rot}} = \boldsymbol{v} \cos \Delta\theta + (\boldsymbol{k} \times \boldsymbol{v}) \sin \Delta\theta + \boldsymbol{k}(\boldsymbol{k} \cdot \boldsymbol{v})(1 - \cos \Delta\theta) \tag{2.32}$$

where $\boldsymbol{v}_{rot}$ is the resultant vector, $\Delta\theta i$ is the rotation angle, $\boldsymbol{v}$ is the original vector and $\boldsymbol{k}$ is the axis of rotation. In this case, we would apply the rotation to the vector $\boldsymbol{m}_i$ about the axis $\boldsymbol{l}_i$.

2.2.5 Material Axis Update

Displacing a node (which occurs when executing node movements, or when perturbing nodes during numerical integration) affects not only the segments on either side of it, but also the orientation of the material axes associated with those segments. We can use parallel transport as described in section 2.2.2 to transport the material axis from the previous (unperturbed) segment to the new (perturbed) one. For example, if we move the node at position i, then the two elements $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i-1}$ will change, and so we update the material axis $\boldsymbol{m}_i$ and $\boldsymbol{m}_{i-1}$

$$\boldsymbol{m}_i' = P(\boldsymbol{m}_i, \boldsymbol{l}_i, \boldsymbol{l}_{\boldsymbol{i}}') = \boldsymbol{R}(\boldsymbol{l}_i, \boldsymbol{l}_{\boldsymbol{i}}') \cdot \boldsymbol{m}_i \tag{2.33}$$

$$\boldsymbol{m}_{i-1}' = P(\boldsymbol{m}_{i-1}, \boldsymbol{l}_{i-1}, \boldsymbol{l}_{\boldsymbol{i-1}}') = \boldsymbol{R}(\boldsymbol{l}_{i-1}, \boldsymbol{l}_{\boldsymbol{i-1}}') \cdot \boldsymbol{m}_{i-1} \tag{2.34}$$

Where $\boldsymbol{m}'$ is the updated material axis, P is an application of parallel transport, $\boldsymbol{m}$ is the unperturbed material axis, $\boldsymbol{l}$ is the (normalised) unperturbed segment, $\boldsymbol{l}'$ is the (normalised) perturbed segment, and $\boldsymbol{R}$ is the rotation matrix defined in equation (2.1).

2.3 Algorithm Implementation

Biological systems such as the kinetochore are millions of atoms in size. The depolymerisation of spindle microtubules, one of the primary generators of force in this system, occurs on timescales of seconds to minutes [101] [8]. It is therefore necessary to build extremely fast and highly parallel algorithms capable of reaching the time (s or more) and length ($> 100nm$) scales required to understand such processes.

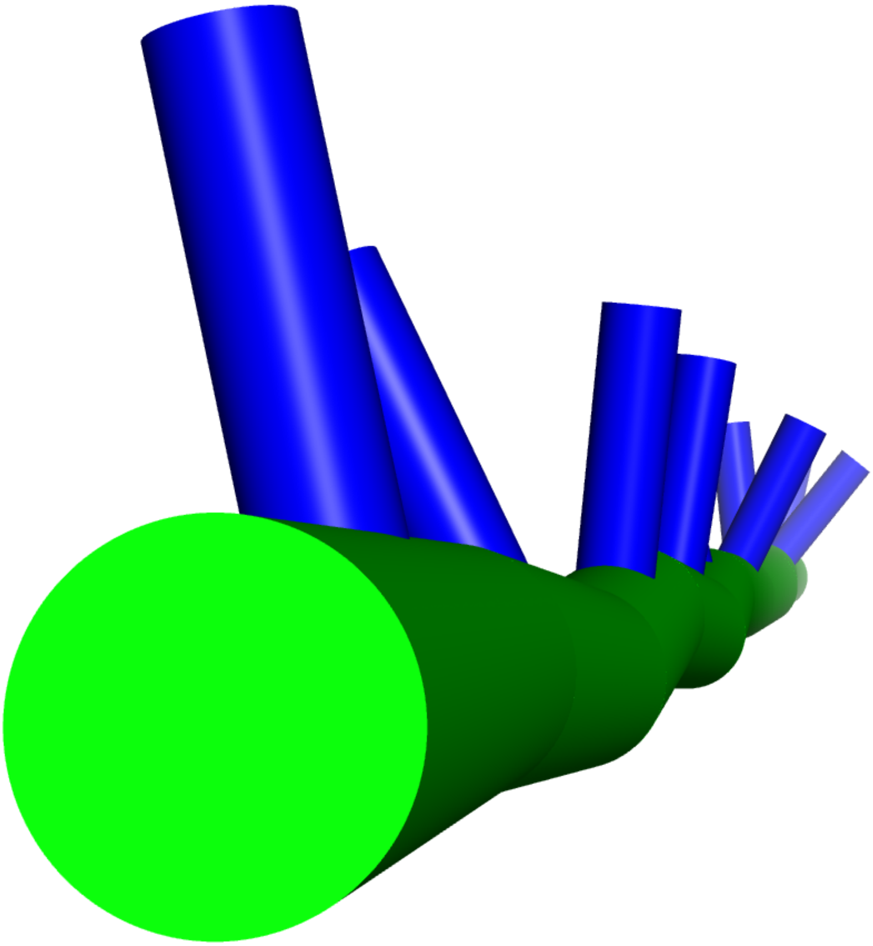


Figure 2.11: Example KOBRA rod rendered using the FFEA plugin for PyMOL [16]. Green cylinders represent rod elements/segments ($\boldsymbol{p}$) and blue cylinders represent material axes ($\boldsymbol{m}$).

KOBRA (Figure 2.11) is implemented in C++ and is designed to run with a small memory footprint and cache-optimised data structures. On startup, the memory needed to store the rod structure is allocated in a series of large, contiguous blocks. This flat, contiguous structure minimizes cache misses, which are a major source of performance loss in numerical simulation algorithms. For the same reason, stack allocation is used for all functionality after initialisation, keeping variables which are likely to be used often as close to the CPU registers as possible. I/O is performed asynchronously. More detailed information on the design of the software can be found in section C.

The structure of the algorithm is as follows:

- Initialise the rod
    - Read rod meta info
    - Allocate memory
    - Read structural data and convert to internal units (see section 2.3.3)
- For each timestep:
    - For each i (OpenMP parallel):
        - Get energies for all perturbations (x, y, z, twist) (section 2.2.3)
    - For each i:
        - Get thermal noise (section 2.2.4)
        - Compute and apply dynamics, update node positions (section 2.2.4)
        - Update material axes (section 2.2.5)
    - If simulation is on a check timestep:
        - Write to trajectory file

2.3.1 Performance

The most intensive task is the calculation of the forces from the gradient of the energy, is parallel per-node using the OpenMP API, and takes around 96% of the program's runtime. This gives the program a serial fraction of 4%, although computing the dynamics in parallel is also possible. Performance and scaling are given in Figures 2.12 and 2.13. The simulation time per timestep is proportional to the number of elements in the rod, and inversely proportional to number of processors.

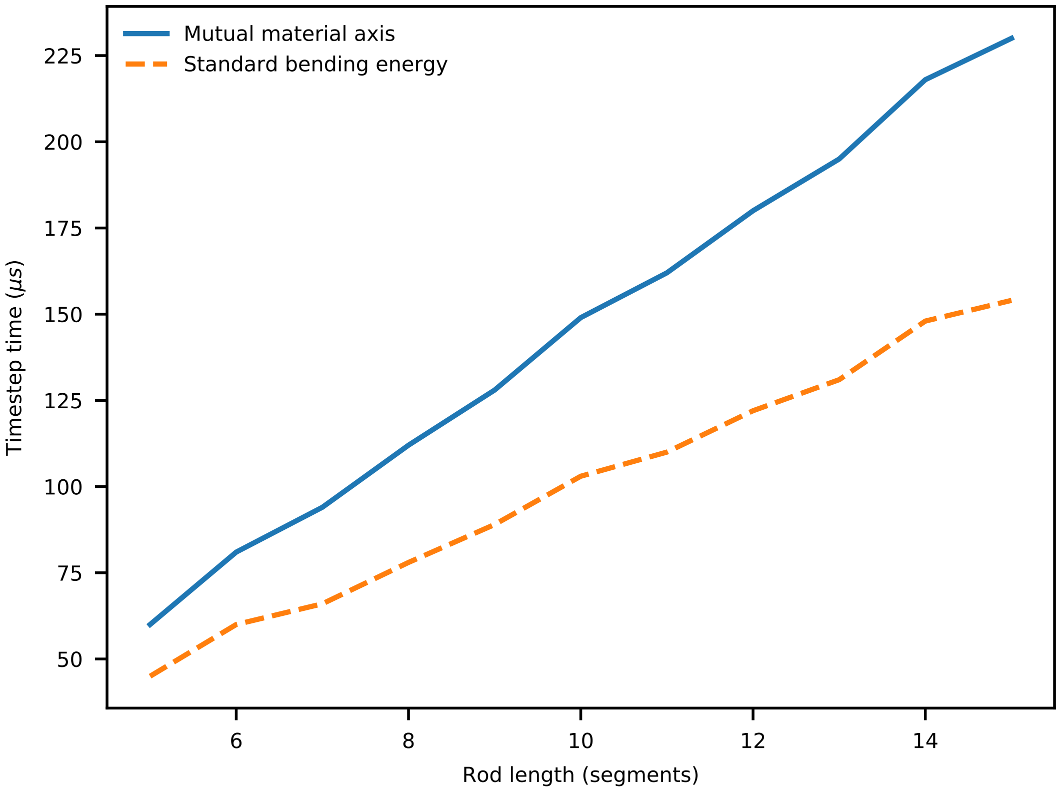


Figure 2.12: Weak scaling: single-thread performance scaling with number of elements, running on an Intel Xeon E5-2670.

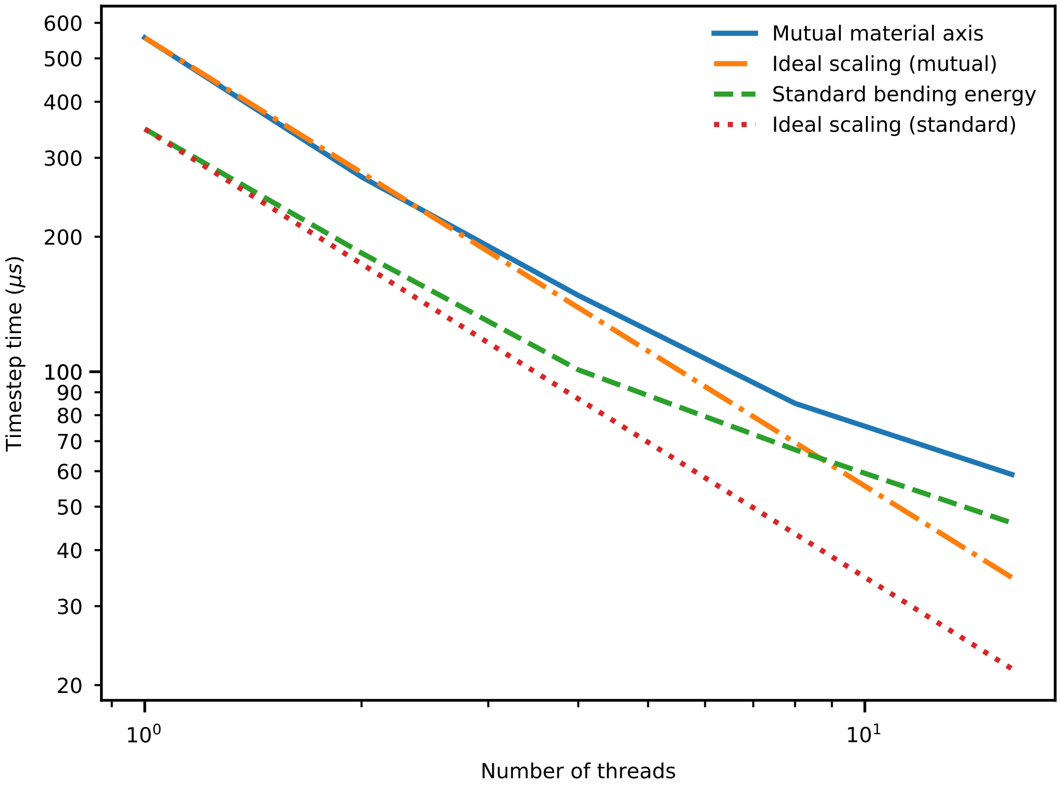


Figure 2.13: Strong scaling: shared memory parallel scaling on a dual Intel Xeon E5-2670 system, with a fixed rod length of 32 nodes.

2.3.2 Factors Determining Performance

400000 iterations of the KOBRA algorithm were profiled using gprof, a software profiling tool. The performance of the algorithm is dominated by calculations of the twist energy, bend energy, material axis updates. The results of this profiling are shown in Figure 2.14.

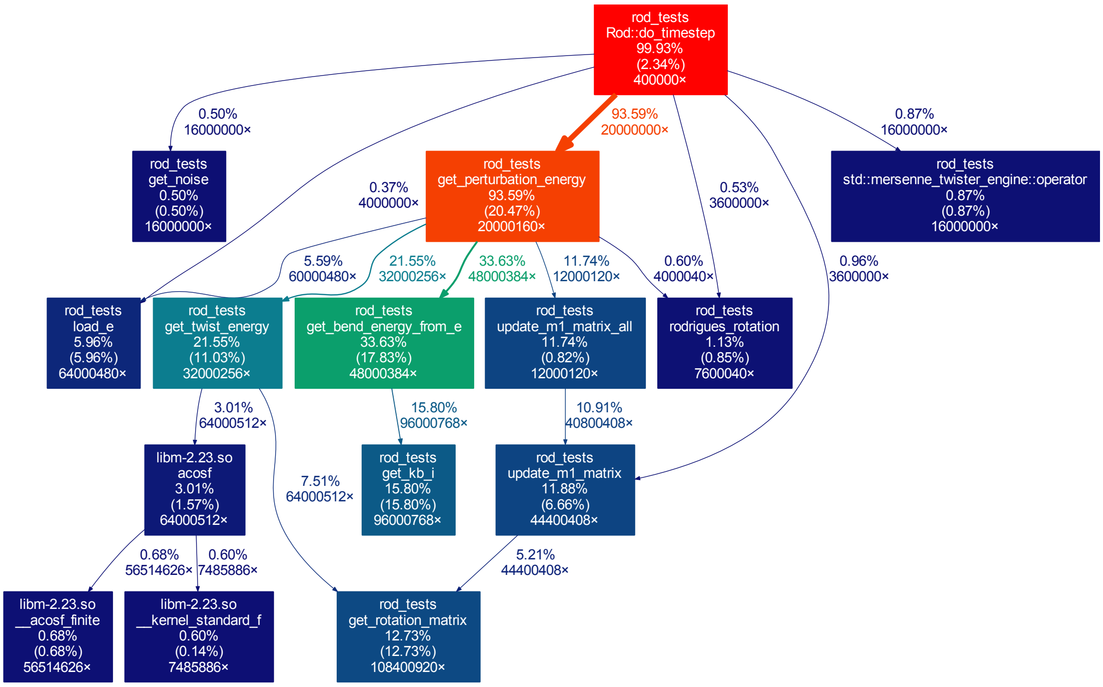


Figure 2.14: Results of profiling the KOBRA algorithm using gprof. This result only pertains to the 'lean' version of algorithm, meaning the mutual material axis and periodic twist energy are disabled. The first % value is the percentage of the runtime spent inside that function, with the number in brackets excluding functions called by that function.

A simple API allows for the manipulation of rods within the simulation and the construction and analysis of rod structures and trajectories. The details of this API are available as part of the official FFEA documentation at http://ffea.bitbucket.io. Information for software developers wanting to improve KOBRA or add KOBRA to their own programs can also be found in this documentation.

2.3.3 Numerical Precision

For a greater degree of cache-optimisation, internal values are specified as single-precision (32-bit) floating point numbers. To preserve numerical accuracy, values are converted to internal units (called 'mesounits') also used by FFEA [1]. The values of these units are given in table 2.1.

Dimension	Symbol	Unit	Comment
Length	l	$1.7 \times 10^{-10}\ m$	C atom VdW radius
Energy	E	$4.142 \times 10^{-21}\ J$	$k_B T, T = 300K$
Mass	m	1.99×10^{-26}	C atom mass
Area		$2.89 \times 10^{-20}\ m^2$	l^2
Volume		$4.913 \times 10^{-30}\ m^3$	l^3
Force		$2.44 \times 10^{-11}\ N$	E/l
Time		$3.73 \times 10^{-13}\ s$	$l \times (m/E)^{\frac{1}{2}}$
Pressure		$8.431 \times 10^{8}\ N/m^2$	E/l^3

Table 2.1: Mesounits used internally in FFEA and KOBRA.

2.4 Validation

Testing and validation is an important and often-overlooked aspect of model-building and implementation. Computer programs, even those created by huge teams at multi-billion dollar corporations, are often buggy and barely work. How do we test that the output of the program we have written matches exactly the algorithm we've designed? How do we know if that algorithm can produce the correct physics?

One way is to design end-to-end tests for the *emergent properties* of the physical systems that are being simulated. These emergent properties are not explicitly included in the algorithm or software implementation, but instead come as a natural consequence of the physics that the KOBRA model represents. In software development, end-to-end tests such as these are called *integration tests*, and the KOBRA software package provides automated versions of these tests.

If the assumptions behind the model are incorrect, or if an error has been made at any stage of the software implementation, these tests will fail. These tests describe physical properties of the algorithm, not biological ones. Chapter 3 will explore the validation of KOBRA in a biological context.

2.4.1 Equipartition Theorem

The equipartition theorem states that the thermal energy of a system will be shared equally between the different degrees of freedom in the system. This can be thought of as a consequence of the maximisation of entropy in the system.

If the rod is intrinsically straight, and the bending modulus is isotropic and stiff enough to avoid large, non-linear deformations, then the energies are quadratic, and the average energy, per node/element and per degree of freedom, will be $\frac{1}{2}k_B T$, where T is the temperature and k_B is Boltzmann's constant. We can write this as

$$\frac{1}{2}\kappa\langle x^2\rangle = \frac{1}{2}k_B T \tag{2.35}$$

Where κ is the constant for any harmonic potential, x is the deformation (Δx, $\Delta\theta$ or $\Delta\omega$ as in expressions for stretching, bending and twisting energy), T is the temperature in Kelvin and k_B is Boltzmann's constant.

To correctly compute the expected equipartition energy, we must consider the number of degrees of freedom the system has. Table 2.2 shows the number of degrees of freedom for a rod of N nodes and $N-1$ segments.

Energy	Number of D. O. F.
Stretch	N-1
Bend	2(N-2)
Twist	N-2

Table 2.2: Number of degrees of freedom for each energy.

The number of degrees of freedom associated with the stretch energy is simply $N-1$, the number of elements. For twist, this is $N-2$, the number of nodes which have elements on both sides of them (and thus which have twist energy between them). There are double the number of degrees of bend ($2(N-2)$) as there are two independent axes of bending for each node. Additionally, the entire rod can rotate and translate in six degrees of freedom. These do not affect the energy, and so do not contribute to the equipartition sum. Thus, the total number of D.O.F. for equipartition is $4N - 7$ out of $4N - 1$ D.O.F. in total.

A test system, comprised of 10 elements, with a total equilibrium length of $100nm$, was simulated for $10\mu s$. The parameters of the system are given in table E.1.

The constants are isotropic, homogeneous and relatively stiff, such that the degree of bending is small. This ensures that these energies remain linear, assumed in equation (2.35). We expect the thermal energy to be $\frac{1}{2}k_B T = 2.07 \times 10^{-21} J$. The mean energies per degree of freedom from this trajectory are given in table 2.3.

D.O.F	Energy
Stretch	$(2.11 \pm 0.16) \times 10^{-21} J$
Bend	$(2.16 \pm 0.11) \times 10^{-21} J$
Twist	$(2.06 \pm 0.16) \times 10^{-21} J$

Table 2.3: Average energies in each degree of freedom for equipartition test.

As the average energies are close to the equipartition value of $\frac{1}{2}k_B T$, this means that the system is getting the correct force from thermal noise, and this force is creating the correct energy in each degree of freedom. The values are on average slightly too large, which may be due to nonlinear effects, which will be discussed more in section 3.2.2. We can observe this in more detail by considering the distribution of bend angles, twist angles and stretch amounts, rather than just the averages.

For a given degree of freedom x, with an associated spring constant κ, the energy is $E = \frac{1}{2}kx^2$ and x is Boltzmann distributed,

$$p(x) \propto e^{\frac{-E}{k_B T}} = \exp\left(-\frac{1}{2}\frac{k}{k_B T}x^2\right) \tag{2.36}$$

Where k_B is Boltzmann's constant, T is the temperature in Kelvin, the deformation x could refer to stretch, bend, or twist (Δp, $\Delta\theta$ or $\Delta\Omega$) and k is the associated constant ($\kappa, \boldsymbol{B}$ or β).

The normalized probability distributions predicted by equation (2.36) were compared to measurements from KOBRA simulation trajectories. For each degree of freedom, we generated a histogram of observed variation in that degree of freedom, and compared this to the probability distribution described in equation (2.36), using the appropriate spring constant. The results are displayed in Figures 2.15, 2.16, 2.17 and 2.18.

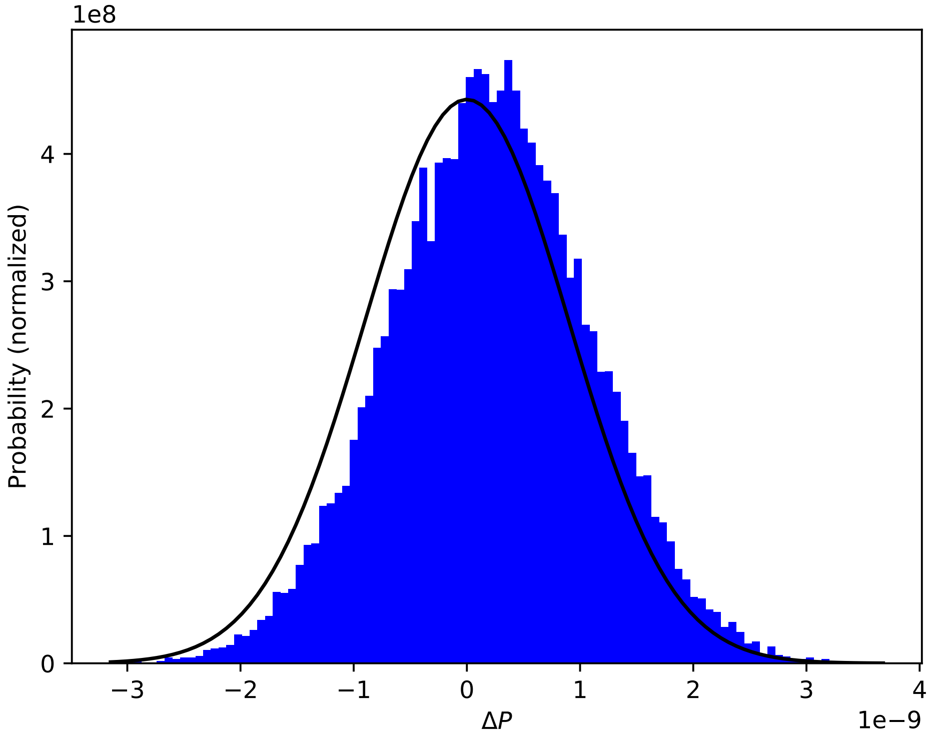


Figure 2.15: Distribution of stretch amounts ($\Delta|\boldsymbol{p}_i|$) for a sample simulation. The expected distribution (equation (2.36)) is shown as a line. The histogram of measured energies, with 100 bins, is shown in blue.

The distribution of stretch amounts (Figure 2.15) has a small amount of systematic stretching. The 3-D nature of the system means that the $\Delta|\boldsymbol{p}_i|$ is not defined exactly by equation (2.36).

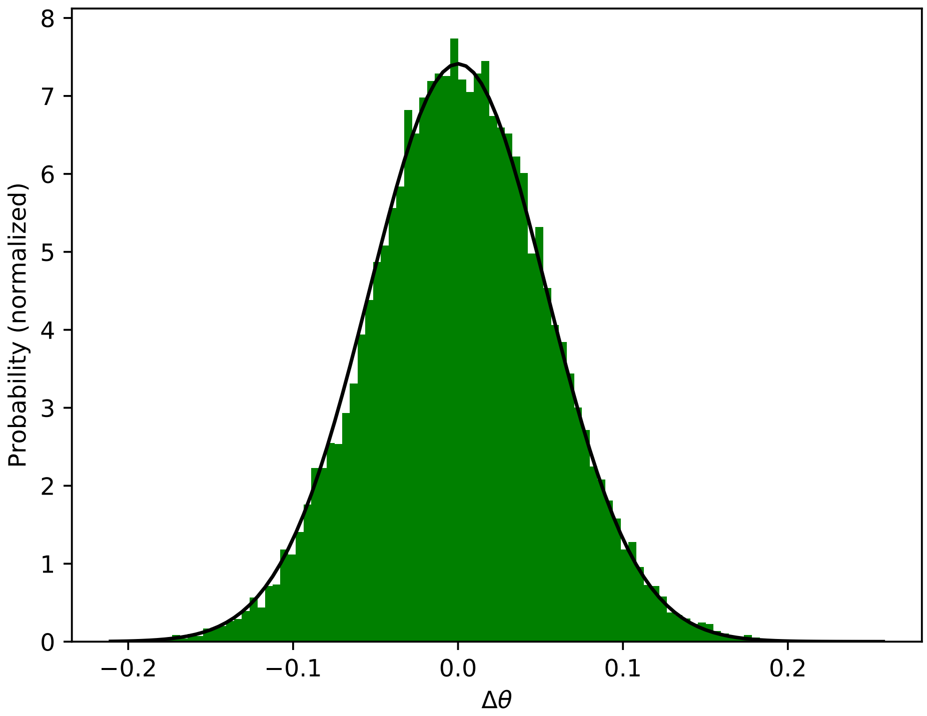


Figure 2.16: Distribution of twist amounts ($\Delta\theta$) for a sample simulation. The expected distribution (equation (2.36)) is shown as a line. The histogram of measured energies, with 100 bins, is shown in green.

For bending, we evaluate this expression for both axes of bending, as these are independent degrees of freedom:

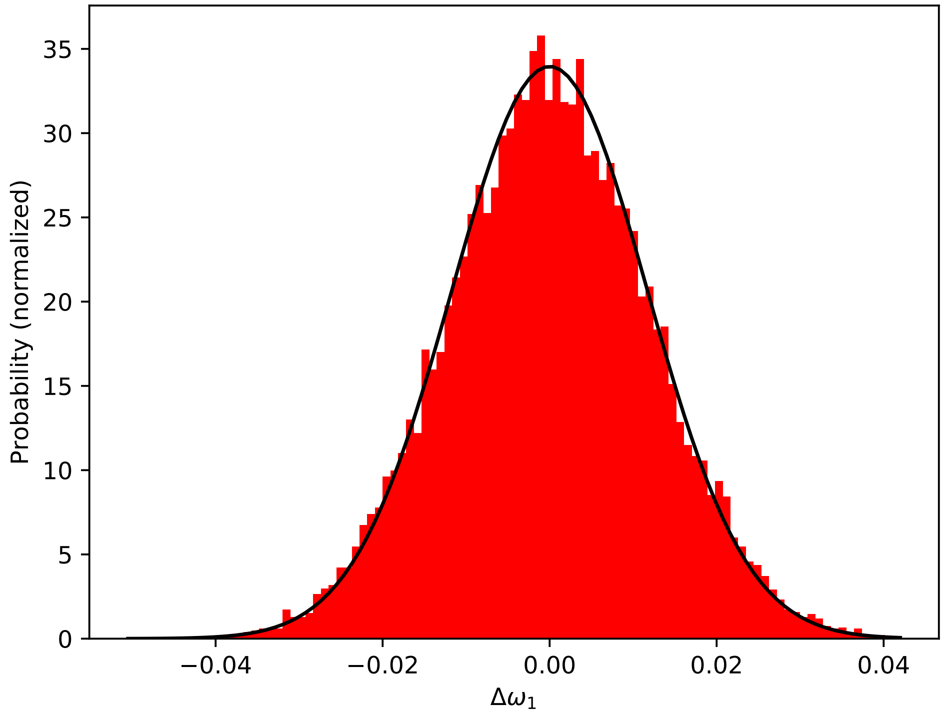


Figure 2.17: Distribution of bend amounts ($\Delta\omega_1$) for a sample simulation. The expected distribution (equation (2.36)) is shown as a line. The histogram of measured energies, with 100 bins, is shown in red.

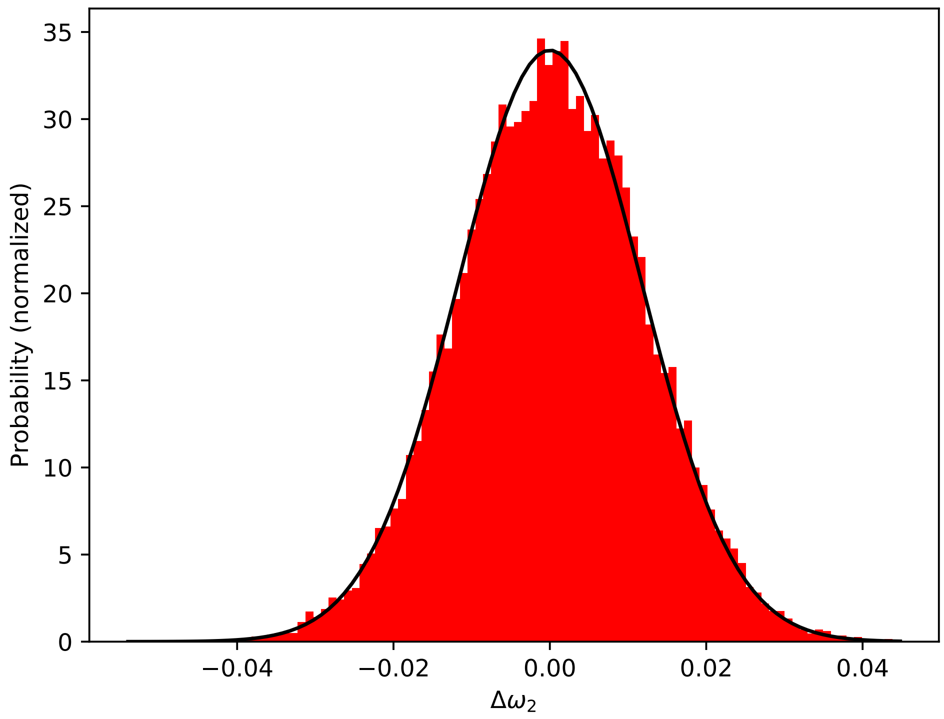


Figure 2.18: Distribution of bend amounts ($\Delta\omega_2$) for a sample simulation. The expected distribution (equation (2.36)) is shown as a line. The histogram of measured energies, with 100 bins, is shown in red.

2.4.2 Comparison with Polymer Chain Models

Like KOBRA, most polymer chain models represent long, thin objects as chains of linear elements connected by nodes. The properties of polymer chains, such as their persistence length, can be computed analytically from the elastic moduli of the rod, but they can also be computed geometrically from the trajectories of the chains. Therefore, we can use polymer chain statistics to compare analytical predictions with measured values from KOBRA simulation trajectories. If the measured values are close to the predicted ones, then the rod is behaving like a polymer chain.

The persistence length of a polymer chain is the length over which the orientation of the elements in that chain becomes uncorrelated. Stiff polymer chains will have a long persistence length, flexible ones will have a short persistence length. Analytically, the persistence length is given [102] by the expression

$$P = \frac{EI}{k_B T} = \frac{B}{k_B T} \tag{2.37}$$

Where $B = EI$ is the bending stiffness, the Young's modulus multiplied by the moment of inertia for a cylinder, k_B is Boltzmann's constant, and T is the temperature in Kelvin. For an isotropic rod, B is equal to the diagonal elements of the isotropic bending stiffness matrix $\boldsymbol{B}$.

In a worm-like chain, the rod elements cannot stretch or twist, and have very small bend angles. If the rod behaves as a worm-like chain, then the persistence length p_i^{wlc} of element i can be measured

$$p_i^{wlc} = \langle|\widetilde{\boldsymbol{p}}_i|\rangle \frac{2}{\langle acos(\theta_i)\rangle^2} \tag{2.38}$$

where

$$\theta_i = \frac{\boldsymbol{p}_i}{|\boldsymbol{p}_i|} \cdot \frac{\boldsymbol{p}_{i+1}}{|\boldsymbol{p}_{i+1}|} \tag{2.39}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ are rod elements.

In a freely-joined chain, inextensible elements are connected about nodes with fixed bend angles, but the rod can be twisted freely. If we assume instead that the rod is a freely-jointed chain, then

$$p_i^{fjc} = \langle|\widetilde{\boldsymbol{p}}_i|\rangle \frac{-1}{\langle \log(\theta_i)\rangle} \tag{2.40}$$

A KOBRA simulation was performed using the same parameters as section 2.4.1. The measured and predicted persistence lengths are compared in Figure 2.19. The analytical persistence length of the rod, calculated from the bending modulus, corresponds to the measured persistence length under the assumption that the rod is a freely-jointed chain.

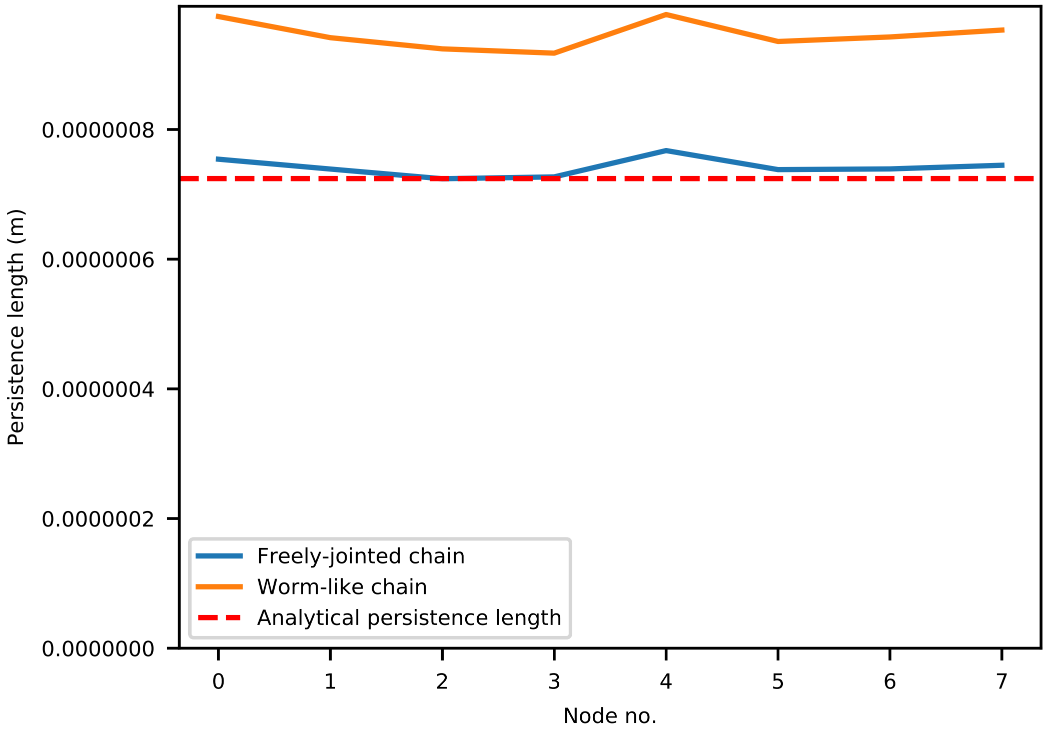


Figure 2.19: Persistence length for a sample rod trajectory.

2.4.3 Buckling Under Torsional Strain

We have observed that KOBRA rods obey the equipartition theorem and the physics of polymer chains. Because this model includes both bending and torsional strain, we can also use it to predict the buckling properties of Euler beams. The experiments corresponding to this qualitative validation can be performed in person, using a rope, bungee cord, hair bobble or the cable for an electrical appliance.

1. An elastic rod which is straight at equilibrium is clamped (with respect to both position and rotation) at both ends.

2. The element at one end of the rod is twisted away from equilibrium while the other end remains clamped.

3. The rod independently starts to kink (‘hockling’) as a result of a balance between the twisting and bending forces.

For this validation, thermal noise is disabled and the two elements at the end of the rod are fixed in place. Then, one an element at one end is twisted by π radians, and then the system is equilibrated. A sufficient number of elements are used to obtain a smooth curve. A comparison of the resulting KOBRA rod with a physical bungee cord can be seen in Figures in 2.20 and 2.21.

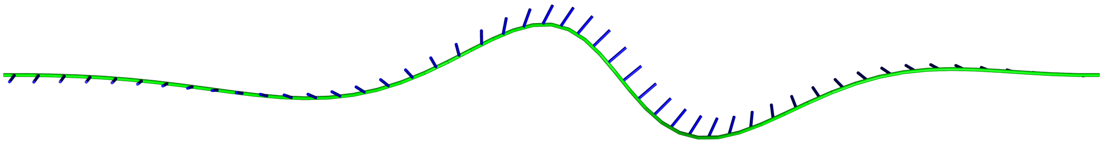


Figure 2.20: Rod fixed at both ends, twisted at one end, and equilibrated, rendered in PyMOL [16].

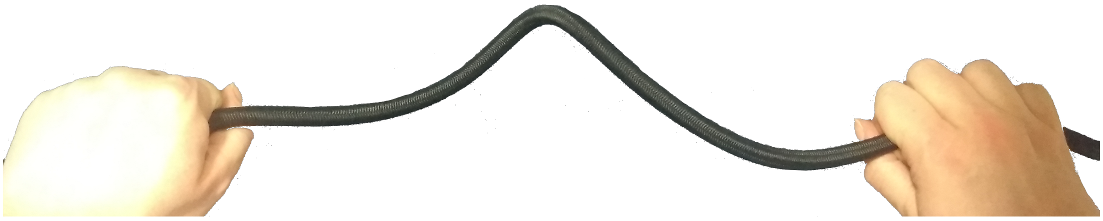


Figure 2.21: Rod (a bungee cord) fixed at both ends, twisted at one end, and equilibrated, in real life, by clammy human hands.

In twisting one end of the cord while keeping its position constant, a torsional strain is introduced into the cord. If this torsional strain is distributed evenly through both bending and twisting modes, the overall energy in the cord will be minimized. The torsional stress is relieved by increasing the *writhe* of the rod. Note that Figure 2.20 shows a large but noncontacting kink. Accurate representations of larger kinks and plectonemes are not possible without a self-contact model.

Chapter 3

Parameterisation of KOBRA Simulations from All-Atom Data

In this chapter, we apply the general model for rod-like biomolecules described in chapter 2 to the specific case of the Ndc80 protein complex (Ndc80C).

As described in section 1.2.5, Ndc80C is a long, flexible molecule comprised of two coiled-coil sections joined by an unstructured loop. This molecule has both a non-straight equilibrium shape and an inhomogeneous, anisotropic bending modulus, which will allow us to test all of the features of KOBRA. A cartoon depiction of the Ndc80C structure is shown in Figures 3.1 and 3.2. One of Ndc80Cs distinguishing features is the presence of a molecular hinge in the centre of its coiled-coil region. This hinge allows Ndc80C to enter highly kinked conformations which may be necessary in the transition from side-on to end-on attachment. Much of the content of this chapter will be centred around investigating the flexibility of Ndc80C and the degree of bending that this can lead to.

A complete all-atom structure for Ndc80C is not available, so we will first construct one using existing partial structures and protein structure prediction. Then, we will extract coarse grained rod model parameters from the dynamics of a simulation of the all-atom Ndc80C. Finally, we will consider metrics we can use to compare these models to each other, and to experimental measurements, such as negative stain EM images [18].

3.1 Creation of an All-atom Ndc80C Model

The Ndc80C protein complex is long, thin and contains disordered regions that are highly flexible. Its flexibility means no crystallographic images of the entire protein complex are available, as it does not form an ordered crystal. Consequently, a full all-atom description of Ndc80C does not exist. To run an MD simulation of Ndc80C, we need to construct a model of the structure by integrating information obtained from experimental studies.

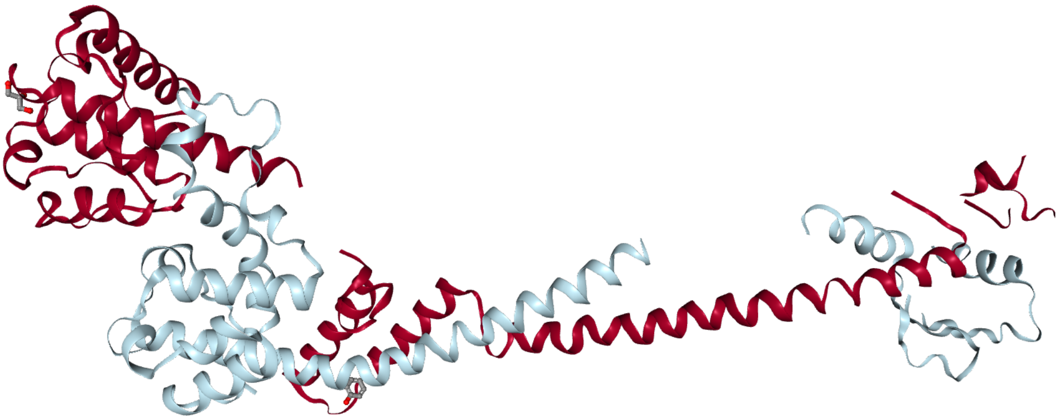


Figure 3.1: Cartoon representation of the Bonsai Ndc80C protein complex rendered in UCSF Chimera. Compare this to Figure 3.6 which contains the majority of the coiled-coil region.

As a starting point, we use the so-called 'Bonsai' Ndc80C molecule [17] (2VE7) (Figure 3.1). This molecule contains both globular domains and a short coiled-coil region 85Å in length. However, most of the coiled-coil and the unstructured loop are removed.

We will need to combine different sources of information to construct a complete all-atom representation of Ndc80C. For more information about the full Ndc80C structure, we turn to Ciferri et al. [17]. They provide the positions of cross-links between the two coiled-coils and a prediction of the entire secondary structure by type: a long coiled-coil with a small, unstructured loop at the centre. They also indicate the relative residue numbers at which these features occur. Figure 3.2 shows the connectivity of Ndc80C's four sub-units (Ndc80 in blue, Spc24 in purple, Nuf2 in yellow and Spc25 in green) and the positions and residue numbers of the cross-links.

The spatial parameters of the coiled-coil, such as the pitch, radius, crick angles and $\phi\ C_\alpha$ (angles describing the geometry of the residues and bonds), can be extracted from the 'Bonsai' Ndc80C molecule. We can use this, combined with the protein sequence [103] [104] and alignment information from Ciferri et al. [17], to construct a basic model of the full-length Ndc80C coiled-coil. We use ISAMBARD, a software package designed for modelling, analysis and parametric design of proteins [105], to generate

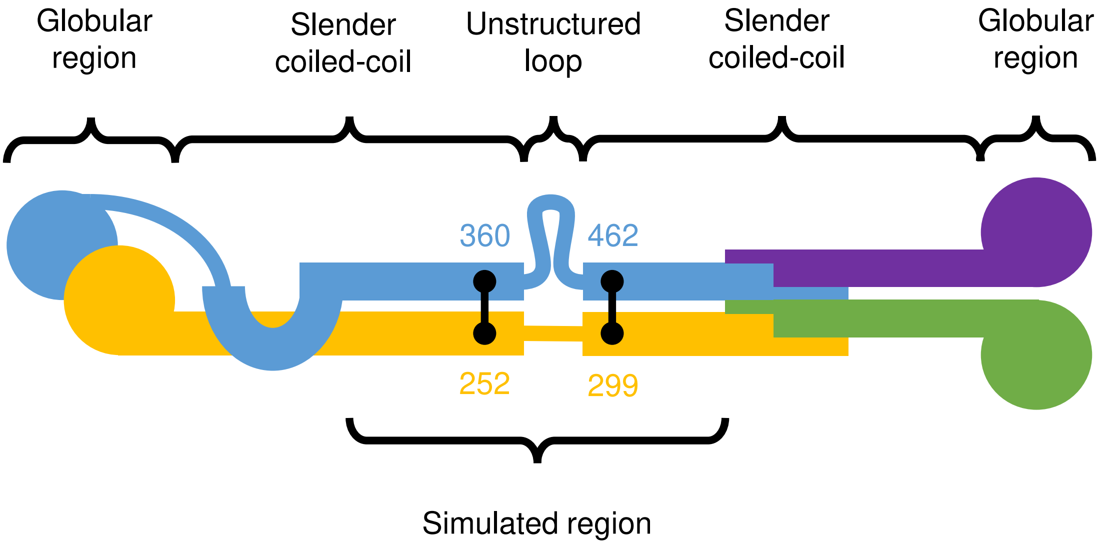


Figure 3.2: Cartoon depiction of the Ndc80 protein complex [17]. The colours distinguish the four connected proteins in the complex (Ndc80 in blue, NUF2 in yellow, and Spc24 and Spc25 in purple and green). The black lines denote cross-links at their respective residue numbers.

a coiled-coil structure from the protein sequence of the four chains in Ndc80C, with each coiled-coil being constructed from two chains, aligned by the residue numbers of their cross-links. For more information, an annotated version of the script responsible for this can be found in doi.org/10.1039/D0SM00491J under the name ndc80_build. In brief, the process is:

- Load the Bonsai Ndc80C structure
- Identify which chains are Ndc80 and Nuf2 based on the residue sequence, remove the globular domains and loop from the sequence and bonsai structure
- Use ISAMBARD to extract the parameters
- Align 2 sequences based on known adjacency of cross-links
- Using ISAMBARD, construct a new coiled-coil from the sequences with these parameters

This method yields two all-atom structures, representing the coiled-coil region on either side of the unstructured loop (but not the loop itself).

Ndc80	NUF2
12.2	-152.0
-16.0	-156.5
-10.5	-165.1
-43.1	138.1
-67.0	131.7
59.0	104.0

Table 3.1: Cα values for NUF2 and Ndc80 coiled-coils in Bonsai Ndc80C molecule.

This generated structure identifies the locations of side-chains, but they are not explicitly represented on an all-atom level. We build side chains onto the coiled-coil using SCWRL4, a protein side chain predictor [106] using SCWRL4's default settings.

We then use the QUARK *ab inito* protein structure prediction server [107] to create predictions of the structures in Ndc80C's loop region from the residue sequence, also using default settings. The results of these predictions can be found at `bitbucket.org/Robert-Welch/kobra-raw-data`, in the file `quark_results.tar.bz2`. These structures were peptide-bonded together with the two coiled-coils created with ISAMBARD using UCSF Chimera [108]. The overall length of the generated Ndc80C, which does not include the two globular domains, is $48nm$. A breakdown of the predicted structure is shown in Figure 3.3.

The resulting model was refined using ModRefiner [109], also at default settings, and minimised using the YASARA minimisation server [110]. This brings the protein, created with somewhat arbitrary bond angles, closer to its native state. The final structure comprises 10984 atoms and is $47\AA$ in length.

Simulations were then performed using AMBER 2014. The ff14SB force field [111] was used as this is AMBER's primary protein model. For computational efficiency, a Generalised Born Surface Area (GBSA) implicit solvent model was used. The sim-

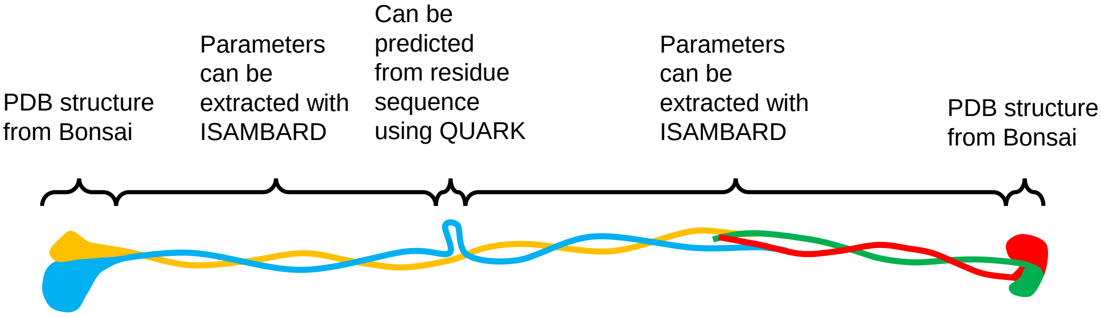

PDB structure from Bonsai
Parameters can be extracted with ISAMBARD
Can be predicted from residue sequence using QUARK
Parameters can be extracted with ISAMBARD
PDB structure from Bonsai

Figure 3.3: Different methodologies used to recreate the all-atom Ndc80C structure (shown to scale).

ulations created a trajectory of 90 nanoseconds over the course of 62 days. A more detailed look into the parameters used in AMBER can be found in table E.3, and the full AMBER trajectories can be found in doi.org/10.1039/D0SM00491J.

3.2 Rod Parameter Extraction

3.2.1 Mapping All-atom Structures Onto Rods

To construct a rod model for Ndc80C, we need to obtain suitable values for the physical parameters κ_s, β and $\boldsymbol{B}$ for each node and element. These parameters are obtained from Ndc80C's dynamics — we select rod parameters that reproduce the local fluctuations in shape from the all-atom trajectory. There is no guarantee that reproducing local fluctuations on an element-by-element basis will produce a system with identical global dynamics — for example, specific interactions between groups of residues, such as binding sites, are not accounted for. A comparison with global dynamics will therefore follow in the next section.

We first map the all-atom MD trajectory directly onto a coarse-grained trajectory for an equivalent rod model. We define the position of nodes in this rod model by averaging over the positions of small clusters of atoms.

$$\boldsymbol{r}_i^r = \frac{1}{k_{max} - k_{min}} \sum_{k=k_{min}}^{k_{max}} \boldsymbol{r}_k^a \tag{3.1}$$

where $\boldsymbol{r}^r_i$ is a rod node at index i, k_{min} and k_{max} are the indices denoting the edges of the ith cluster, and $\boldsymbol{r}^a_k$ is the position of the atom at index k. For a given value of i, the values of k_{min} and k_{max} are minimum and maximum indices of atoms, denoting the edges of that cluster. Using only small clusters means that the rod configuration is not unnaturally smoothed during coarse-graining. The rod segments, as before, are defined as end-to-end vectors such that $\boldsymbol{p}_i = \boldsymbol{r}_{i+1} - \boldsymbol{r}_i$.

The backbone of the all-atom Ndc80C model was coarse-grained to 14 nodes from 2668 atoms. The optimal length for rod segments depends on the system being discretised. It should be as coarse as possible, while being fine enough to capture information about the implicit shape and dynamics of the molecule being modelled. In the case of Ndc80C, the rod should be fine enough to resolve the molecular hinge, but each element should be no shorter than a turn of the coiled-coil. At such a resolution, it would be necessary to represent the coiled-coil explicitly using two rods, which is not possible without a rod self-contact model.

Each node was averaged from a 10-atom wide cluster according to equation (3.1) (see Figure 3.4). This number is also system-dependent, for Ndc80C it was chosen such that there is more than a factor of 10 difference between number of atoms per rod element and the number of atoms being used to set a node position.

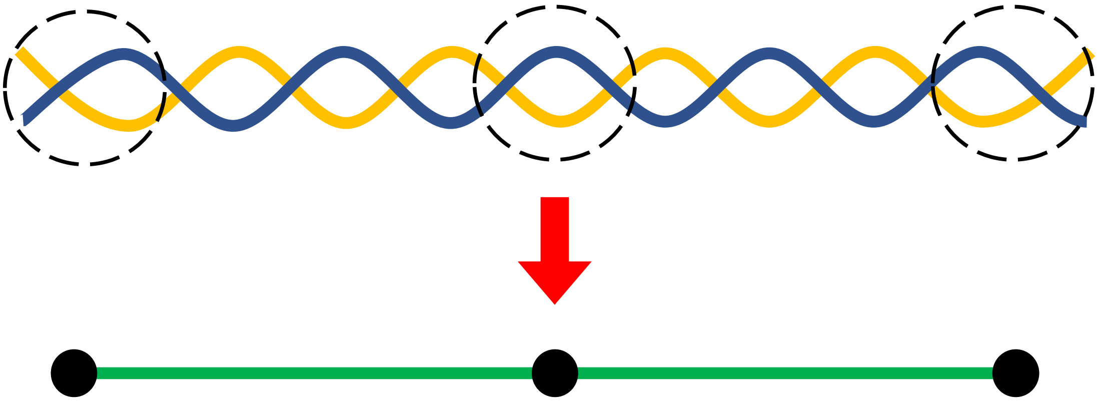


Figure 3.4: To coarse-grain the atomistic Ndc80C trajectory, the co-ordinates of groups of atoms are averaged according to clusters of atom indices.

Having defined the nodes, we now need to define the material axis of each coarse-grained segment in order to measure twist. One method for doing this is to first calculate the vector $\boldsymbol{q}_i$ between the opposite atoms in the two chains of the coiled-coil (Figure 3.5):

$$\boldsymbol{q}_j = \frac{\boldsymbol{r}_j^A - \boldsymbol{r}_j^B}{|\boldsymbol{r}_j^A - \boldsymbol{r}_j^B|}, \tag{3.2}$$

where $\boldsymbol{r}_j^A$ is the position of the jth atom in chain A, and $\boldsymbol{r}_j^B$ is the position of the atom in chain B which is closest to $\boldsymbol{r}_j^A$ at equilibrium. In practice, for each rod we select 10 atom pairs which are located halfway between the atoms defining two nodes at the end of the rod; we take the average $\boldsymbol{q}_j$ from those atom pairs, as indicated in Figure 3.5.

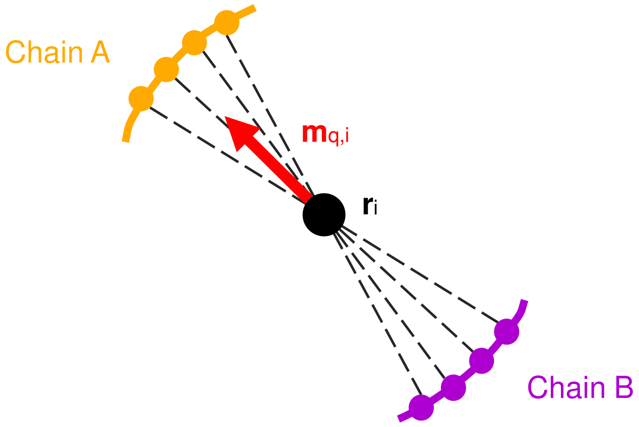


Figure 3.5: Schematic view of the coiled-coil cross-section, showing how the material axis is computed. The dashed lines are used to compute $\boldsymbol{q}_i$ for each atom, which are then averaged to $\boldsymbol{m}_q, i$. For clarity, only four atoms in each chain are depicted.

To ensure that the material axes are always normal to the elements, a material axis $\boldsymbol{m}_{\boldsymbol{q},j}$ is defined from the orthonormal projection of $\boldsymbol{q}_i$ from the tangent of the element, $\boldsymbol{l}_j$,

$$\boldsymbol{m}_{q,j} = \frac{\boldsymbol{q}_j - (\boldsymbol{q}_j \cdot \boldsymbol{l}_j)\boldsymbol{l}_j}{|\boldsymbol{q}_j - (\boldsymbol{q}_j \cdot \boldsymbol{l}_j)\boldsymbol{l}_j|}. \tag{3.3}$$

The material axes of adjacent elements obtained from equation (3.3) may point in quite different directions, and although the above equations for calculating deformation energy permit this, it is practically better (e.g. when defining the bending stiffness matrix) to maintain similar directions for the material axes of adjacent elements. The material axes of two adjacent elements can be considered to be in the 'same' direction if the material axis $\boldsymbol{m}_i$ of element i, when parallel transported onto element $i+1$, points in the direction of the material axis $\boldsymbol{m}_{i+1}$ of that element.

Consequently, we use the following procedure for calculation of material axes. We begin in the equilibrium configuration of the atomistic model. In this equilibrium configuration, we obtain the material axis $\boldsymbol{m}_0$ of the first element according to equation (3.3). We then generate the material axis for all other rod elements in the equilibrium configuration by parallel transport, by iterating along the rod.

$$\widetilde{\boldsymbol{m}_{\boldsymbol{i}}} = P\left(\widetilde{\boldsymbol{m}_{\boldsymbol{i-1}}},\ \widetilde{\boldsymbol{l}}_{\boldsymbol{i-1}},\ \widetilde{\boldsymbol{l}}_{\boldsymbol{i}}\right). \tag{3.4}$$

For each element i we can also calculate a candidate material axis $\boldsymbol{m}_{\boldsymbol{q},i}$ from the local atomistic positions according to equation (3.3). In the equilibrium configuration we may compare the material axis $\boldsymbol{m}_i$ obtained by parallel transport with the material axis $\boldsymbol{m}_{\boldsymbol{q},i}$. These two vectors will in general differ by some rotation angle $\widetilde{\Delta\theta_i^u}$, which are stored. Then, during the subsequent dynamics of the MD trajectory, at any moment we can generate the instantaneous $\boldsymbol{m}_{\boldsymbol{q},i}$ according to equation (3.3), and calculate the instantaneous material axis $\boldsymbol{m}_i$ by rotating $\boldsymbol{m}_{\boldsymbol{q},i}$ about the rod by signed angle $\widetilde{\Delta\theta_i^u}$, using the Rodrigues rotation formula (equation (2.32)).

Using this method, the all-atom Ndc80C model containing 2668 atoms (top of Figure 3.6) was coarse-grained to a rod model comprised of 14 nodes (bottom of Figure 3.6). Each node was averaged from a 10-atom wide cluster according to equation (3.1) (see Figure 3.4). The Python script that builds this model is available at `doi.org/10.1039/D0SM00491J`, and as part of the FFEA software package.

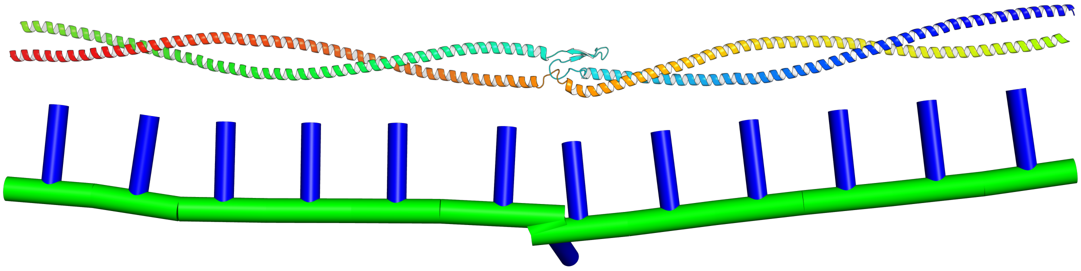


Figure 3.6: Top: Completed atomistic Ndc80C model after minimisation. Bottom: simplified rod model showing equilibrium atomistic structure. The blue lines represent the equilibrium orientation of the material axes, and their twisting indicates the coiled-coil pitch. The radius displayed is arbitrary.

3.2.2 Calculating Rod Parameters

Having mapped the MD simulations onto an equivalent rod trajectory, our strategy for model parameterisation is to select parameters for the rod elements and nodes such that the local mean square fluctuations in bending, twisting and extension will match those observed in the MD trajectory. From this trajectory, we therefore compute (for each rod, or node, i) the mean square fluctuations in rod length $\langle (|\boldsymbol{p}_i| - |\widetilde{\boldsymbol{p}}_i|)^2 \rangle$, and in twisting angle $\langle \Delta\theta_i^2 \rangle$ where $\Delta\theta_i$ is the difference between instantaneous twisting angle and its equilibrium value.

Because the rod model obeys the equipartition theorem (section 2.4), we can use the equipartition result to determine the probability distributions associated with the parameters specifying the stretching, twisting and bending of the rod. The probability distribution for the variable x in the potential $V(x)$ due to thermal energy is the Boltzmann

$$P(x) \propto \exp\left(-\frac{V(x)}{k_B T}\right). \tag{3.5}$$

If we substitute the harmonic potential $V(x) = \frac{1}{2}kx^2$, we can use the standard result, that for a normal distribution where $P(x) \propto \exp\left(\frac{-x^2}{2\delta^2}\right)$, the variance $\langle x^2 \rangle = \delta^2$. Therefore

$$P(x) \propto \exp\left(\frac{-kx^2}{2k_BT}\right) \Rightarrow k = \frac{k_BT}{\langle x^2\rangle}. \tag{3.6}$$

As an initial estimate for the model parameters, we note that the energies defined in equations (2.6), (2.9) and (2.21) are all quadratic in the respective parameters. We might therefore expect fluctuations to be distributed normally and the equipartition theorem to apply (section (2.4)). This would result in initial estimates for the parameters as

$$k_i = \frac{k_BT}{\langle(|\boldsymbol{p}_i| - |\widetilde{\boldsymbol{p}}_i|)^2\rangle} \tag{3.7}$$

$$\beta_i = \frac{L_i k_B T}{2\langle\Delta\theta_i^2\rangle}. \tag{3.8}$$

The same result can be used to compute the values of the elements of the $\boldsymbol{B}$ matrix (equation (2.15)). In the case of anisotropic rods, the bending energy is formed from the scalar product of the $\boldsymbol{B}$ matrix with the elements of the 2-vector $\boldsymbol{\omega}$. The values of the $\boldsymbol{\omega}$ 2-vector, representing the material curvature, will also be normally distributed about the equilibrium value, $\widetilde{\boldsymbol{\omega}}$. This probability distribution can be written as

$$P(\Delta\boldsymbol{\omega}_{i,\alpha}^m, \Delta\boldsymbol{\omega}_{i,\beta}^m) \propto \exp\left(\frac{-1}{2k_BT \cdot L_i} \cdot (\Delta\boldsymbol{\omega}_{i,\alpha}^m \Delta\boldsymbol{\omega}_{i,\beta}^m) \cdot \boldsymbol{B}_i \cdot \begin{pmatrix}\Delta\boldsymbol{\omega}_{i,\alpha}^m \\ \Delta\omega_{i,\beta}^m\end{pmatrix}\right) \tag{3.9}$$

Since bending is parameterised by a two-component vector $\boldsymbol{\omega}_i^m$ (see equation (2.14)), fluctuations in bending are parameterised by a covariance matrix for bending fluctuations, for node i,

$$\boldsymbol{C}_{i,\alpha\beta} = \langle\Delta\boldsymbol{\omega}_{i,\alpha}^m \Delta\boldsymbol{\omega}_{i,\beta}^m\rangle = k_BTl_iB_{i,\alpha\beta}^{-1} \tag{3.10}$$

where $\Delta\boldsymbol{\omega}^m_{i,\alpha}$ is the difference between the α component of the bending vector $\boldsymbol{\omega}^m_i$ and its time average value (where $\alpha = 1$ or 2).

We therefore derive the following expression for the elements of $\boldsymbol{B}$.

$$\boldsymbol{B}^i = k_B T L_i \cdot \boldsymbol{C}_i^{-1} \tag{3.11}$$

Note that $\widetilde{\boldsymbol{B}}_i$ is independent of the element length in a homogeneous rod. In equation (3.11), the angular fluctuations increase as the element length increases, so $\boldsymbol{C}_i$ increases. But equation 3.12 is scaled by L_i, which compensates for this. However, for a hinge, the angular fluctuations are more or less fixed, independent of the length of the elements on either side. Therefore, C_i is constant, but $\widetilde{\boldsymbol{B}}_i$ increases with L_i. In this method, this isn't a problem, as we discretise the rod before we parameterise it.

The above results also neglect the three dimensional nature of the rod model dynamics, and the non-linear geometrical effect this has on the probability distributions of rod model variables. A uniform density of states for the x-, y- and z-coordinates of node positions does not equate to a uniform density of states for rod length and bending co-ordinates.

As a result, the probability distributions for these variables are not exactly normally distributed (one can think of this as an extra entropic contribution to the free energy of fluctuations arising from the density of states in the space of the bending or rod length co-ordinates). However, in equation (3.6), we assume that they are. This means that, in going from (x, y, z) co-ordinates to co-ordinates in terms of $\boldsymbol{\omega}$, we over-estimate the energy at large bend angles, and so over-estimate the values of $\widetilde{\boldsymbol{B}}$.

If we use equation (3.11) to estimate the bending stiffness matrix, then typically the mean square fluctuations in the resulting rod model do not match those of the original simulation. However, they are sufficiently close that an iterative scheme is able to rapidly converge on the correct parameterisation. For a given node i, our aim is to reproduce the degree of bending fluctuations $\boldsymbol{C}$ as measured in the all atom MD simulations. Running a KOBRA model built using the bending energy matrix $\boldsymbol{B}'_i$, given by application of equation (3.11), will result in a trajectory with observed fluctuations $\boldsymbol{C}_i^{\text{old}}$ which will in general be different from $\boldsymbol{C}_i^{\text{target}}$ (even after reducing statistical errors by running a long trajectory). By considering the expected change in $\boldsymbol{C}$ from a small change in $\boldsymbol{B}$, we can obtain a new estimate for the bending energy matrix as

$$\boldsymbol{B}_i^{\text{new}} = k_B T L_i \left(\boldsymbol{C}_i^{\text{target}} - \boldsymbol{C}_i^{\text{old}} + \frac{k_B T L_i}{\boldsymbol{B}_i^{\text{old}}} \right)^{-1}. \tag{3.12}$$

Where $\boldsymbol{B}_i^{new}$ is the corrected value of $\boldsymbol{B}_i$, k_B is Boltzmann's constant, T is the temperature, L_i is defined in equation (2.10), $\boldsymbol{C}_i$ is the $\boldsymbol{C}$ matrix for the original trajectory, $\boldsymbol{C}_i^{\text{new}}$ is the $\boldsymbol{C}$ matrix for the new trajectory, and $\boldsymbol{B}_i^{\text{old}}$ is the bending stiffness matrix calculated for the original trajectory using equation (3.11).

A KOBRA simulation run using this new estimate at the bending energy matrix will give bending fluctuations closer to the target value. If necessary, the iteration can be repeated, but we have found that a single iteration is usually sufficient to satisfactorily reproduce the target bending fluctuations within the typical uncertainty produced from statistical sampling of trajectories.

3.2.3 Parameterisation Validation

The method for computing $\boldsymbol{B}$ from an existing trajectory was validated against a trajectory from a KOBRA rod with known values of $\boldsymbol{B}$. A simulation with 9×10^6 frames recorded at intervals of $5ns$ was analysed. This length of trajectory is required, as the method described in 3.2.2 cannot replicate the parameters needed to create a particular set of conformations without a trajectory that features those conformations. The reference rod is anisotropic, so bends more in one axis than another, and inhomogeneous, featuring a hinge in its centre. Figure 3.7 shows the results of this parameterisation. The raw trajectories are available at `doi.org/10.1039/D0SM00491J`, and more information on the simulation parameters used can be found in table E.4.

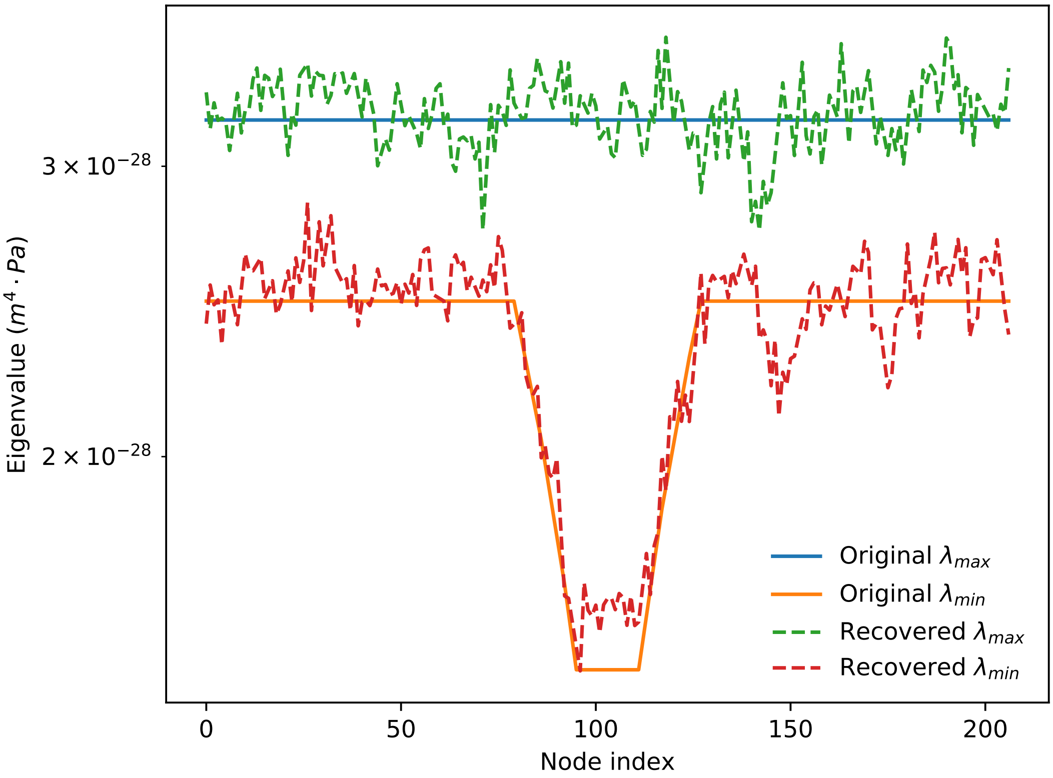


Figure 3.7: 'Recovered' values of the $\boldsymbol{B}$ matrix computed from a simulation trajectory, compared to the actual values given to the nodes in that trajectory. Here, the two lines represent the maximum and minimum eigenvalues.

The method described in section 3.2.2 only works for rods with relatively small values of $\Delta\boldsymbol{\omega}$. Figure 3.7 shows the results of parameter recovery for a rod with typical $\boldsymbol{B}_i$ values ($\begin{bmatrix} 2.5 \times 10^{-28} & 0 \\ 0 & 3.2 \times 10^{28} \end{bmatrix}$), but extremely short elements (the rod contains a total of 209 elements, and each element is $1.2nm$ in length).

For systems with larger elements, we recover the parameters iteratively. A KOBRA simulation with 49 rod elements with a known $\boldsymbol{B}_i$ was run, and the fluctuations in the bending angle were measured, giving the observed C_i. The parameters used for this simulation can be found in table E.4, and the raw data can be found online at `doi.org/10.1039/D0SM00491J`. Applying equation (3.11) results in the extracted values of $\boldsymbol{B}'_i$ shown on the left-hand side of Figure 3.8. Running a simulation with these extracted parameters gives fluctuations C_i', different to C_i. Iterating once using equation (3.12) then gives the extracted values of $\boldsymbol{B'}$ shown on the right-hand side of Figure 3.8.

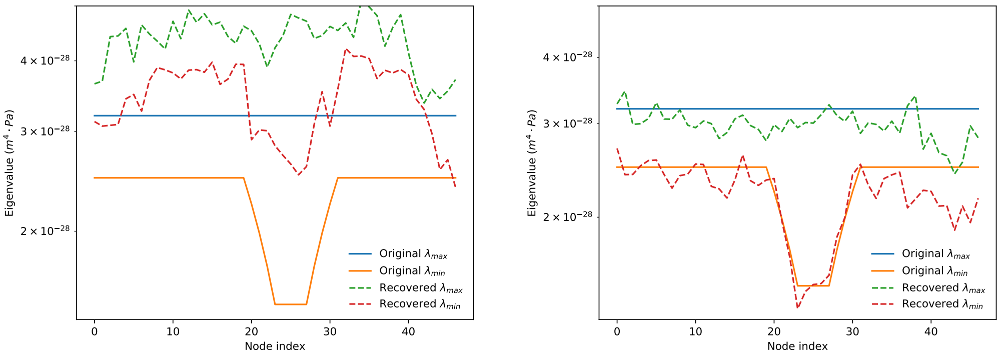


Figure 3.8: Maximum and minimum eigenvalues of the $\boldsymbol{B}$ matrix found using the iterative parameterisation scheme, compared to known reference values. The left-hand side shows the results of the parameterisation using equation (3.11). The right hand-side shows the results after one iteration of parameter recovery using equation (3.12).

The values of $\boldsymbol{B_i'}$ generated through such a parameterisation, applied to a molecular dynamics trajectory, can only represent the conformational space explored by the molecule during that trajectory. If the trajectory is short, this may produce an error as a result of undersampling.

To represent how this error scales with trajectory length, a KOBRA trajectory was created with a rod with known values of $\boldsymbol{B}_i$. For different fractions of the trajectory — the entire trajectory, up to and including frame M - the value of $\boldsymbol{B}_i'$ was calculated using equation (3.12). Figure 3.9 shows how the RMSD between the actual values and the recovered ones (the RMSE) decreases over time.

$$\delta \boldsymbol{B}_i' = \left(\frac{1}{M} \sum_{j=1}^{M} \frac{|\boldsymbol{B}_i^j - \boldsymbol{B}_i^{j,\prime}|^2}{\boldsymbol{B}_i^{j,\prime}} \right)^{\frac{1}{2}} \tag{3.13}$$

Where $\delta \boldsymbol{B}_i'$ is the fractional error in $\boldsymbol{B}_i'$, the recovered value of $\boldsymbol{B}_i$. The values of $\delta \boldsymbol{B}_i'$ are averaged over the entire rod to produce Figure 3.9.

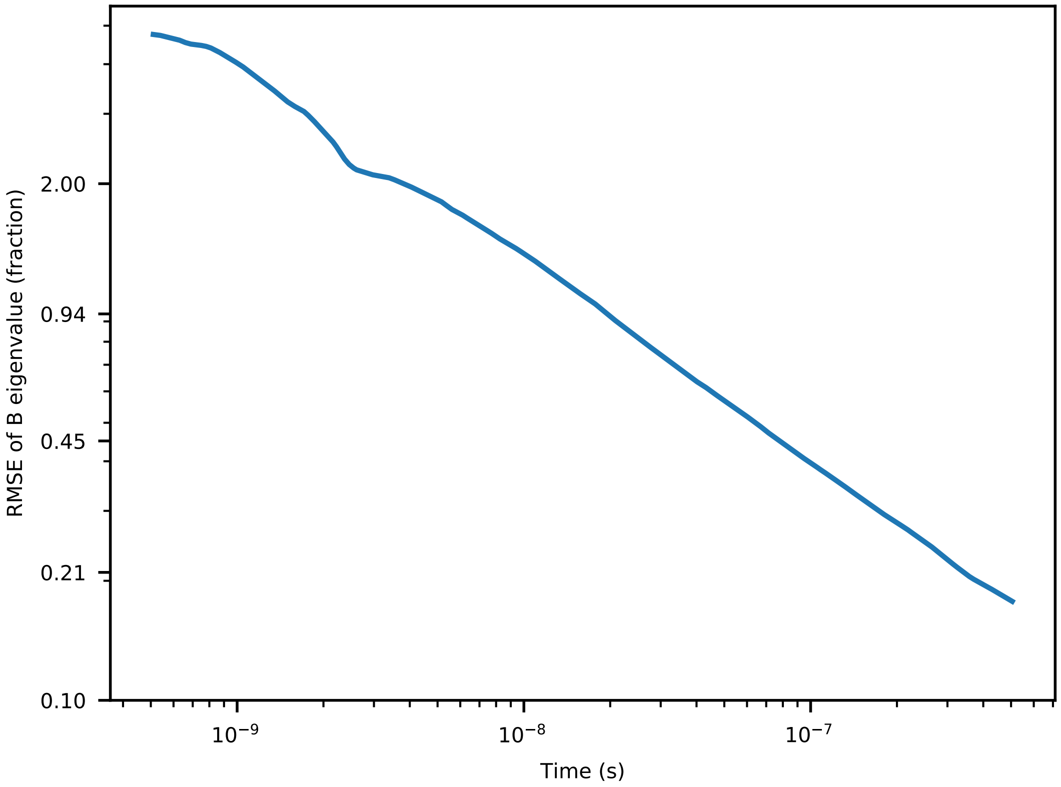


Figure 3.9: RMSE of the $\boldsymbol{B}$ eigenvalues for different fractions of the trajectory, for a test trajectory with known values of $\boldsymbol{B}$. The very start of the trajectory has been truncated to better show the scale.

Note that rod simulations are overdamped, so they explore conformational space more slowly than a comparable all-atom simulation. This means that the timescales of rod and all-atom simulation trajectories are not always directly comparable.

For values typical of coiled-coils, the other parameters (β_i and k_i) can be recovered without iteration. Figures 3.10 and 3.11 show the recovery of inhomogeneous values of β_i, using equation (3.8) for a trajectory consisting of 50000 frames and totalling $50\mu s$. For this simulation, the parameters used can be found in table E.4, and the raw data is available online at `doi.org/10.1039/D0SM00491J`.

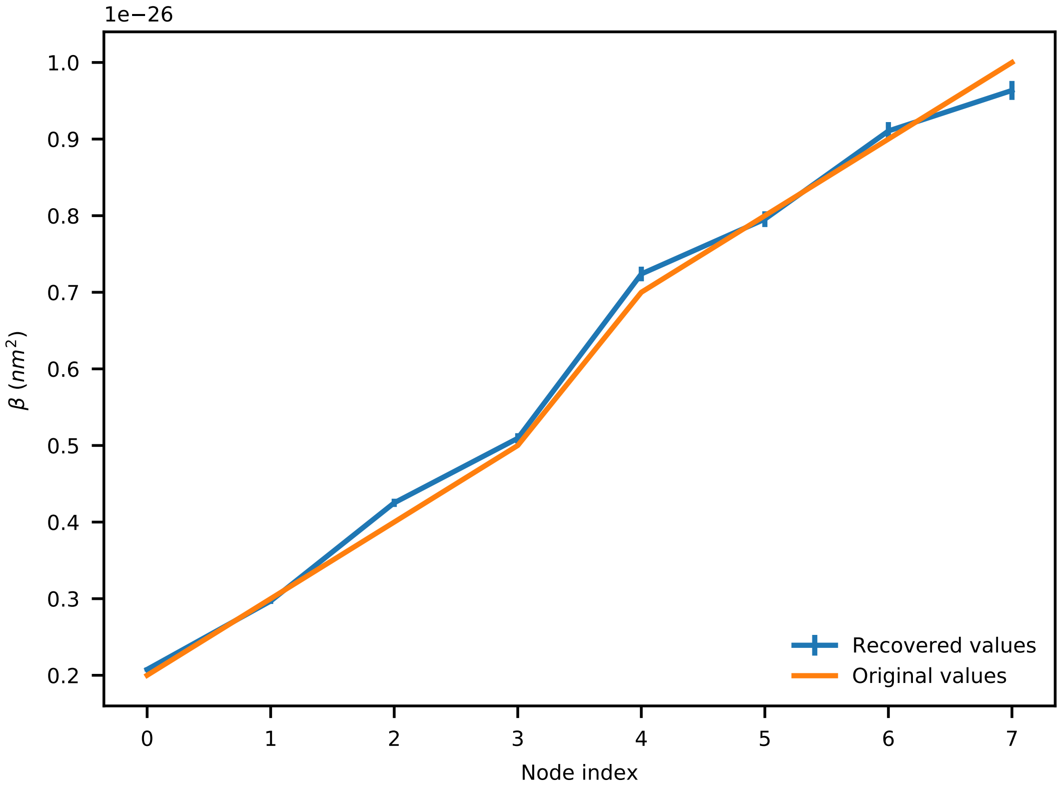


Figure 3.10: Twist recovery parameterisation for a rod with inhomogeneous twist parameters, compared to known values.

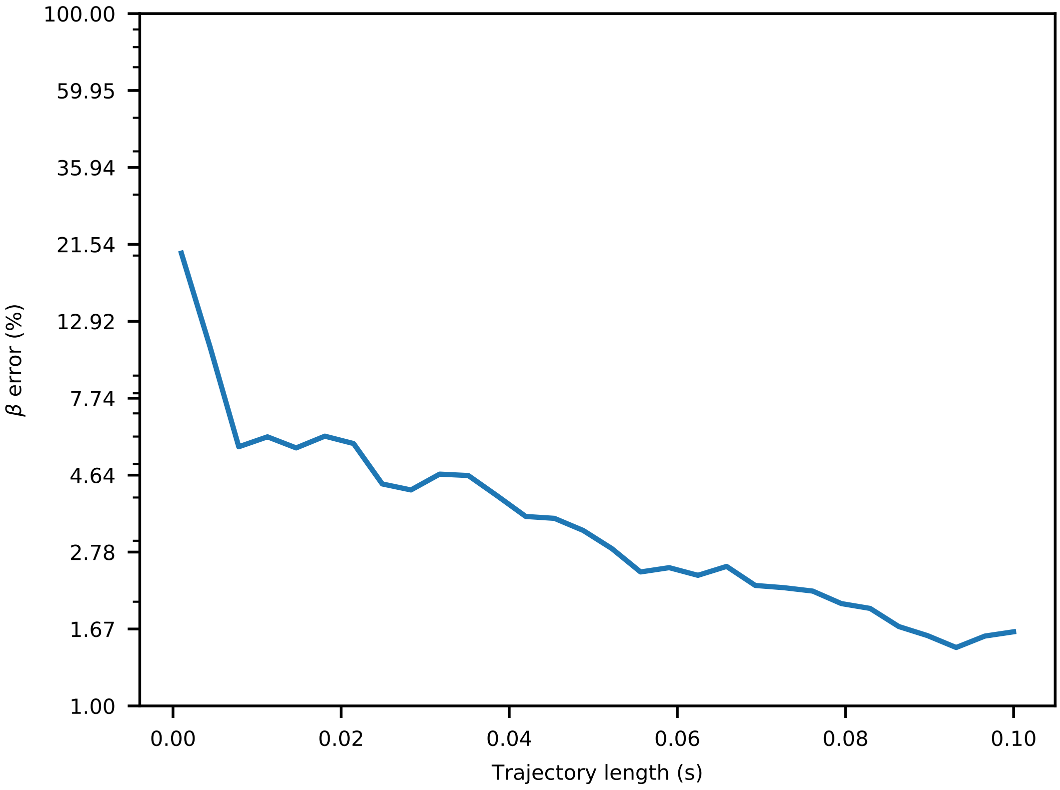


Figure 3.11: Percentage error in the recovered value of β calculated for different fractions of the trajectory length.

The recovery of κ_i is shown in Figure 3.12.

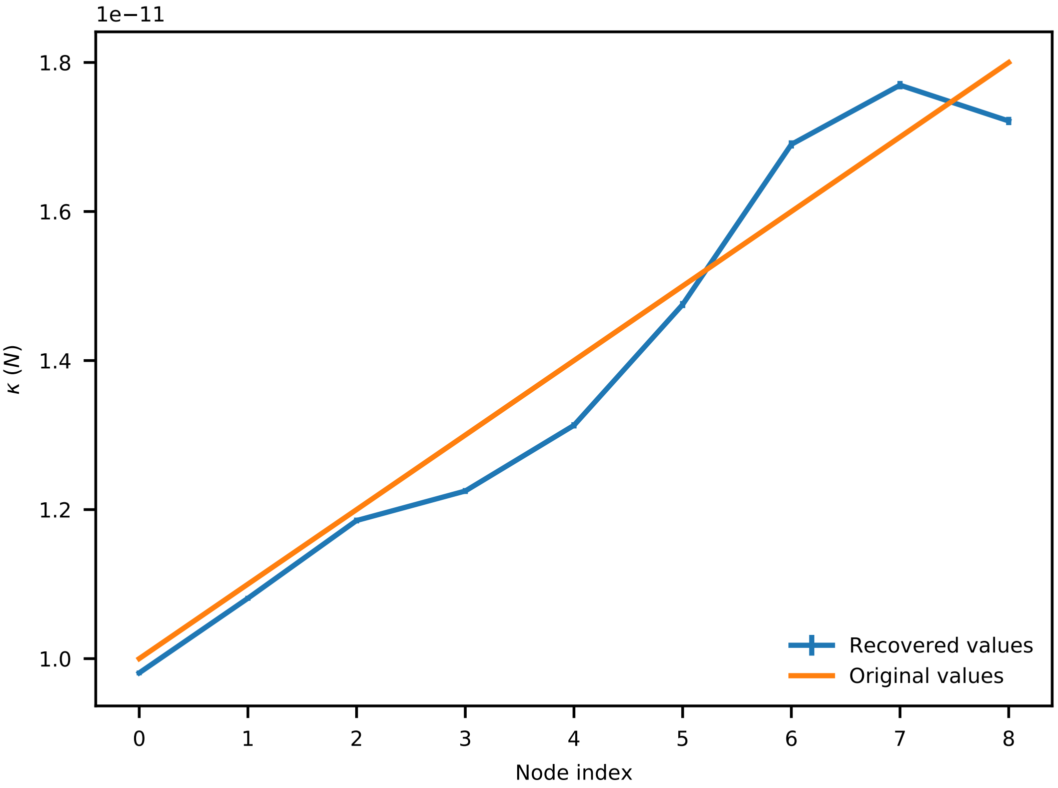


Figure 3.12: Stretch recovery parameterisation for a rod with inhomogeneous stretch parameters, compared to known values.

3.3 All-atom Parameterisation Results

Using the method described in section 3.2.2, and the molecular dynamics simulation from section 3.1, we can acquire a set of rod parameters for Ndc80C. Two rod simulations were run to provide a single iteration for parameter recovery. The simulation parameters can be found in table E.4, and the raw data at `doi.org/10.1039/D0SM00491J`. The average values of the recovered parameters for the entire molecule (including the flexible hinge) are given in table 3.3, specified per unit length (see section 2.2.3 for an example).

Parameter	Recovered Value
κ	$(1.718 \pm 0.017) \cdot 10^{-11}\ N$
$\boldsymbol{B}$	$(5.15 \pm 0.29) \cdot 10^{-31}\ m^4 \cdot Pa$
β	$(1.32 \pm 0.07) \cdot 10^{-29}\ Nm^2$

Table 3.2: Average values of rod parameters for Ndc80C. $\boldsymbol{B}$ is assumed to be isotropic in this case, so the values quoted are the diagonal elements of the $\boldsymbol{B}$ matrix.

In addition to increasing its flexibility, Ndc80C's hinge region means that it is slightly more susceptible to bending in one axis than in the other. This hinge is localised to a small region in the centre of Ndc80C, so we need to look at the parameters extracted on a per-node basis.

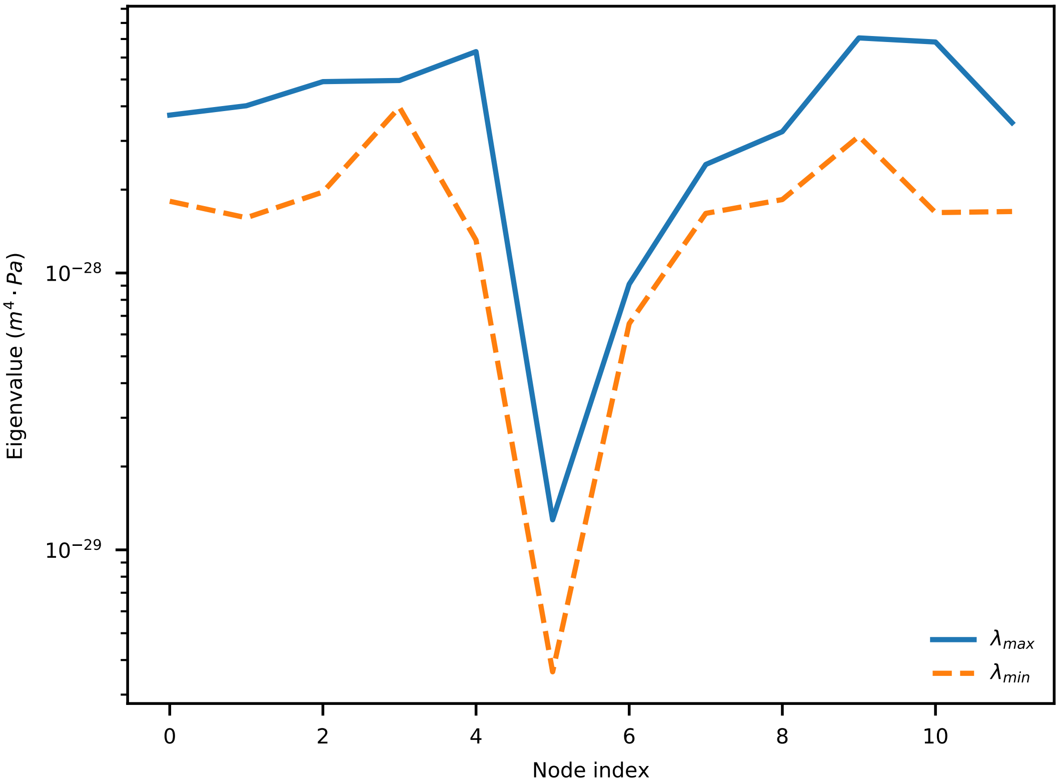


Figure 3.13: The eigenvalues of the bending energy matrix $\boldsymbol{B}^i$ for the rod calculated from the atomistic Ndc80C trajectory. Here, the two lines represent the maximum and minimum eigenvalues, not any particular axis.

Figure 3.13 indicates the two eigenvalues of $\boldsymbol{B}^i$ as a function of the node number. It shows a localised region of decreased stiffness between nodes 4 and 6 in the parameterised rod model, which corresponds to the unstructured loop region (the hinge) in the atomistic model. This hinge is also observed in the negative stain EM imaging of Ndc80C [18], which will be examined in more detail in section 3.4.1.

3.4 Comparison of KOBRA Dynamics With Atomistic Molecular Dynamics and Experimental Data

The parameterisation from section 3.3 was used to create a KOBRA rod model and run a simulation comprised of $100,000$ frames, with a timestep of $1 \times 10^{-12} s$, writing frames at 3000 step intervals, for a total simulation time of $300 \mu s$. This is almost three orders of magnitude longer than the atomistic simulations, though these simulations only ran for 24 hours on the same workstation.

3.4.1 Comparison of Molecular Kink Angles

Wang et al. [18] use negative stain EM to obtain 2D images of a variety of Ndc80C conformers (Figure 3.14). They force the Ndc80C molecules to lie on a flat carbon support, and measure the 'kink angle', defined as the angle between the two halves of the molecule, before and after the unstructured loop. This provides a distribution of angles, based on 83 observations, and so gives an experimental measurement of the flexibility of Ndc80C.

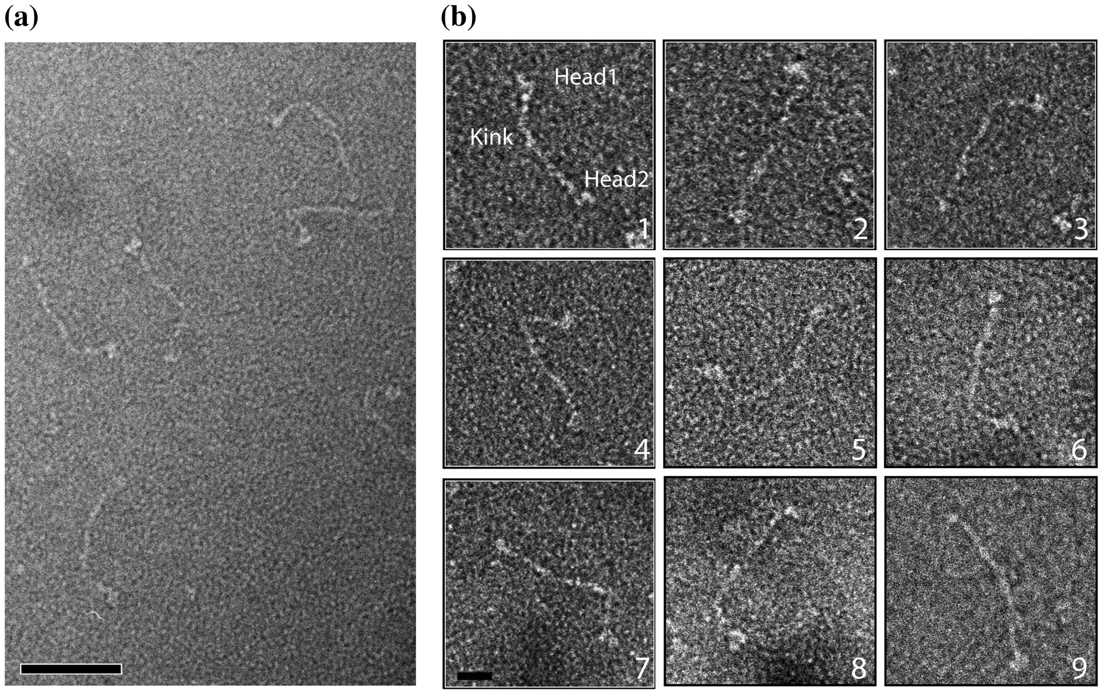


Figure 3.14: Negative stain EM images of Ndc80C [18]. The scale in the lower left is 50nm in length.

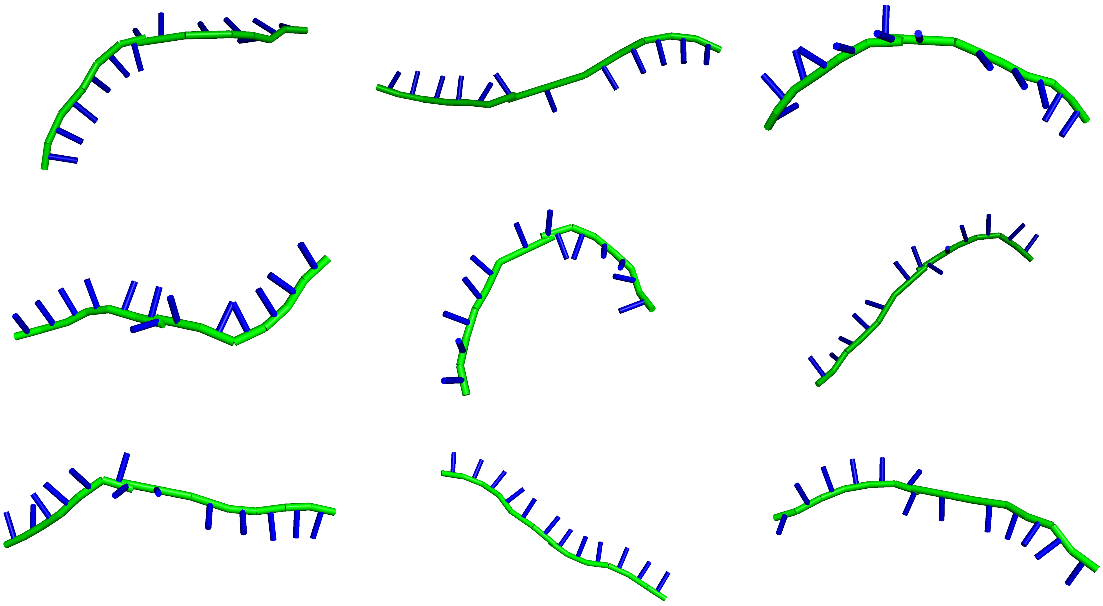


Figure 3.15: Snapshots from the Ndc80C rod trajectory rendered in PyMOL. Green lines denote elements, blue lines denote material axes.

From both the KOBRA rod (Figure 3.15) and atomistic MD trajectories, the kink angles can be measured using the end-to-end vectors $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ of the two halves of the coiled-coil region (i.e. from one end of the coiled-coil to the loop/hinge region, and then from the loop/hinge region to the far end of the coiled-coil).

$$\theta = \arccos\left(\frac{\boldsymbol{p}_a \cdot \boldsymbol{p}_b}{|\boldsymbol{p}_a||\boldsymbol{p}_b|}\right) \tag{3.14}$$

This angle is computed for each frame of the trajectory, then binned in increments of 10 degrees in order to preserve parity with the experimental data, and the resulting distribution normalised.

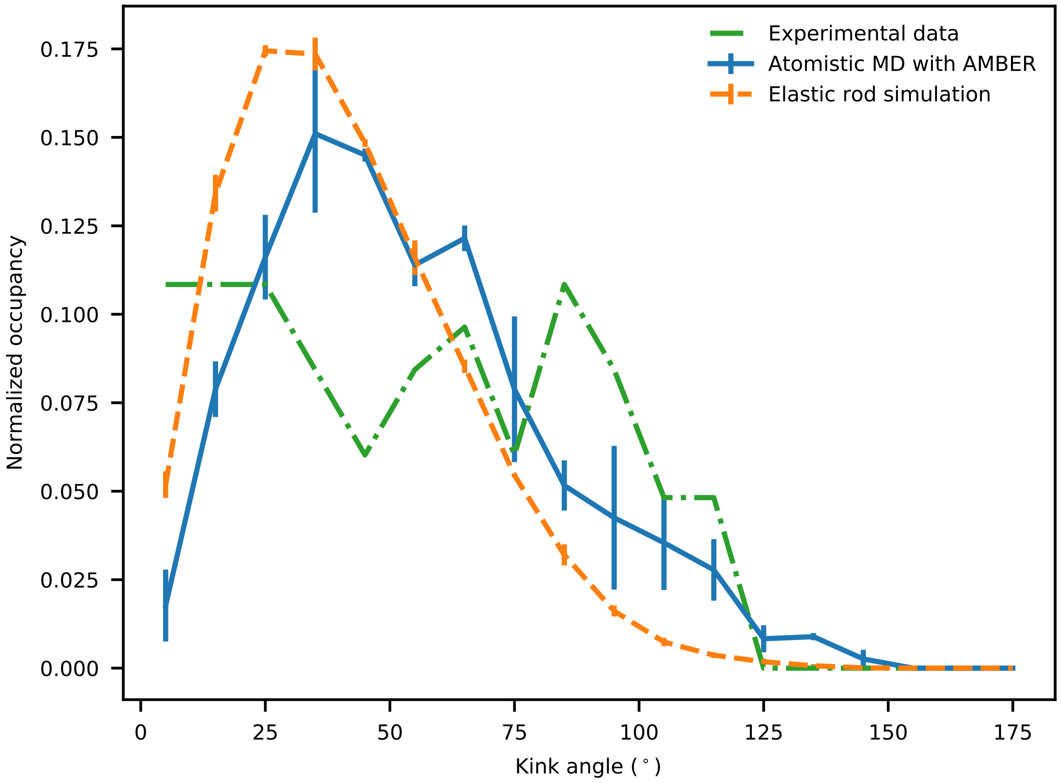


Figure 3.16: Kink angles for experimental data, rod simulation and atomistic simulation. Angles are given as absolute values. Errors are computed by finding $|Deltaf(\theta)|$, the difference in the occupancy for two halves of the trajectory.

The distribution of kink angles for the KOBRA rod and atomistic trajectories are very similar (Figure 3.16). Since the distribution of bend angles at each node from the MD trajectory was used to parameterise the KOBRA rod simulation this can be interpreted as an indication of a successful parameterisation.

The distribution of the experimental angles retains the same general features. The experimental distribution does not capture the initial maximum, as the resolution was too low to resolve bend angles less than 30 degrees - these angles were all inserted into a single bin. Both the atomistic and rod simulation find a broad maximum of bend angles between 25 and 75 degrees centred on 50 degrees, which falls off rapidly for angles above 100 degrees. The experimental data broadly replicates these features although the range of the bend angles is broader. It should also be noted that the experimental distribution is constructed from only 83 samples, compared to 10,000 frames/3μs for the rod trajectory and 50,000 frames/90ns for the atomistic trajectory.

3.4.2 Comparison of Principle Components

Although we have set the rod material parameters directly from the dynamics of the all-atom rod, it does not follow that the two models will exhibit identical dynamics, as the rod material parameters were obtained from local measurements, whereas the eigenvalues are global properties of the simulation trajectory. Furthermore, the MD simulation is still relatively short and may be undersampled. To understand which types of large-scale motion are conserved from the all-atom simulation to the KOBRA simulation, we can analyse the trajectories from the two simulations using principal component analysis (PCA) [112].

PCA is a statistical method used to reduce the dimensionality of molecular dynamics data sets, by identifying the principal eigenmodes of motion (the principle components) that result from thermal (or other) fluctuations. Each eigenmode represents a different structural deformation. The principal components are ordered from the largest to the smallest value of the associated eigenvalue (and thus, amplitude of the component). PCA was performed using the software package pyPcazip [113].

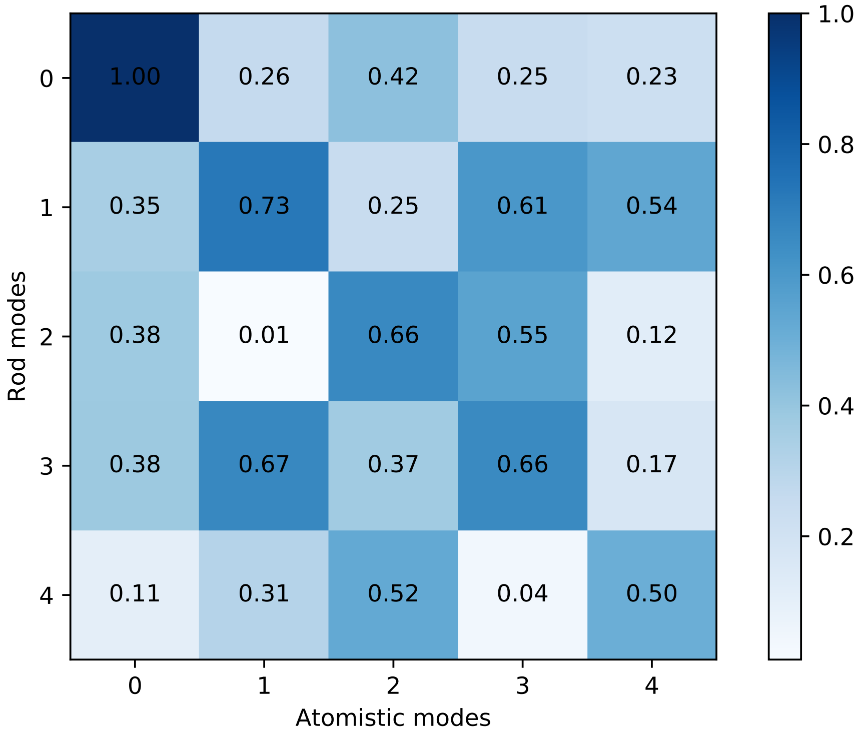


Figure 3.17: Dot product matrix of the average principal component eigenvectors. The values shown are normalised to the size ordering of the largest dot product.

Figure 3.17 shows the dot product matrix comparing the rod and all-atom simulations. Each cell shows the dot product of the eigenvector of the rod mode with the corresponding all-atom mode. A dot product close to one means that the eigenvectors are highly correlated. The eigenvectors are sorted by eigenvalue size, so if the matrix is diagonal, it also means that the relative magnitude of the various modes of bending are in the same order. The highest correlation is between the first modes, which correspond to bending about the hinge. Although diagonal elements of the matrix dominate, there are some significant off-diagonal components, which could be evidence of mode mixture, particularly between mode 3 and 4.

As KOBRA cannot capture the motions of individual atoms, any mode that relies on these motions cannot be replicated. Also note that the accuracy of the principal components is limited by the length of the trajectories and the amount of conformational space each one has explored. Finally, these eigenvectors only represent the direction of the motion, not the magnitude, so Figure 3.17 is a better reflection of the relative directions of the different components of the bending matrix than the actual value of those components.

To observe the mode mixture and visualise the motion represented by the first few modes of the PCA, we can create ‘PCA animations’, that show the range of motion for each eigenvalue. The PCA animations are created using the following formula,

$$\boldsymbol{r}_i^{anim} = \boldsymbol{r}_i + (\lambda f)\boldsymbol{v} \tag{3.15}$$

where $\boldsymbol{r}_i^{anim}$ is the animated node, $\boldsymbol{r}_i$ is the original node, λ is the eigenvalue for that node, $\boldsymbol{v}$ the eigenvector, and f is a scalar between 0 and 1. The resulting figure shows how the nodes oscillate within that principal component.

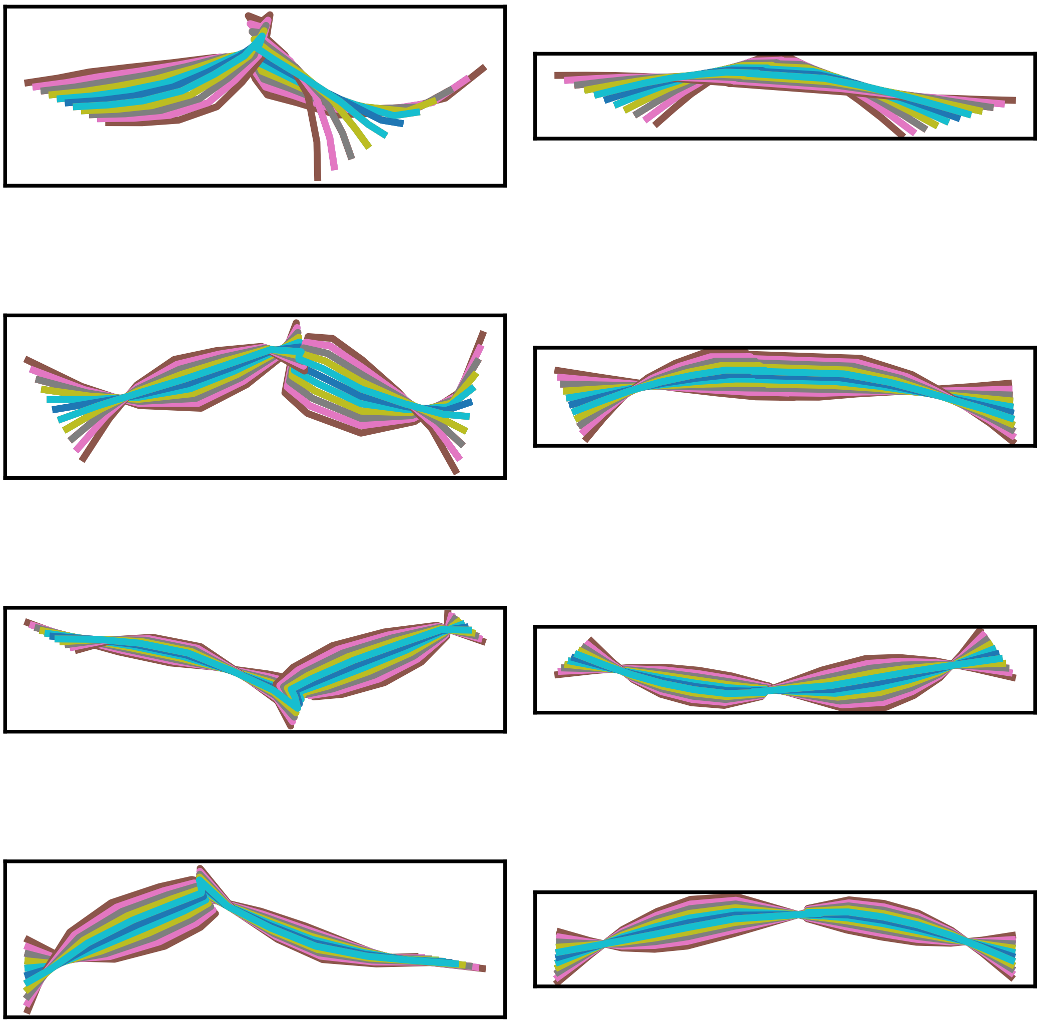


Figure 3.18: Comparison between the principal components of both trajectories. Left: original, all-atom trajectory, coarsened to 14 nodes. Right: rod simulation. Both models were projected into a plane containing the end-to-end vectors of two halves of the molecule. The principal components appear in decreasing order of size.

Figure 3.18 compares the range of motion exhibited in the first four principal component eigenmodes obtained from the rod and the all-atom MD simulations. The first mode shows a central hinge, with similar magnitudes of bending for the rod and atomistic model. The second and third all-atom modes show the fluctuation of two different coiled-coil regions. While the second and third principal components initially look dissimilar, the dot product matrix suggests there is mixing between these modes. For example, modes 1 and 2 of the atomistic trajectory show motion on opposite sides of the hinge, whereas in the rod, this motion is distributed evenly between both modes. For reference, when assessing correlations between eigenmodes from multiple all-atom trajectories, these normally display a significantly smaller degree of diagonal correlation between PCA eigenmodes from separate runs on the same model (see Figure A.2) than is shown in Figures 3.17 and 3.18.

Together, Figures 3.17 and 3.18 indicate a strong correlation between the motions involved in the atomistic MD simulation and the KOBRA rod trajectories. This level of correlation is typical (and in fact good) for comparisons of PCA between similar MD simulations, or between MD and coarse grained simulations. The reason the correlation is not perfect is because the MD trajectories are too short to sample a large number of global configurations (the large scale modes) of the molecule [114], so there is a statistical sampling error. We have used local information, such as the variation in local bend and twist angles, to parameterise the coarse grained rod model, rather than the variation in global configurations. For short MD trajectories, matching the local bend and twist variation does not necessarily translate to exactly matching the (statistically limited) variations observed in global configurations.

3.4.3 Range of Motion of the Hinge

We can compute the root-mean-square displacement (RMSD) of our simulation trajectories on a per-element basis. This gives us a profile of the rod's flexibility. We can use this metric to compare the relative flexibility of the rod and atomistic trajectories.

To compute the best-fit RMSD, we first fit the equilibrium structure to the trajectory using the iterative closest point algorithm described in section A. Then, the RMSD of the node positions is calculated in the bioinformatics style [115].

$$\mathrm{RMSD}(\boldsymbol{r}, \widetilde{\boldsymbol{r}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{r}_i - \widetilde{\boldsymbol{r}}_i\|^2} \tag{3.16}$$

The results of the best-fit RMSD calculation for atomistic and KOBRA rod structures are shown in Figure 3.19.

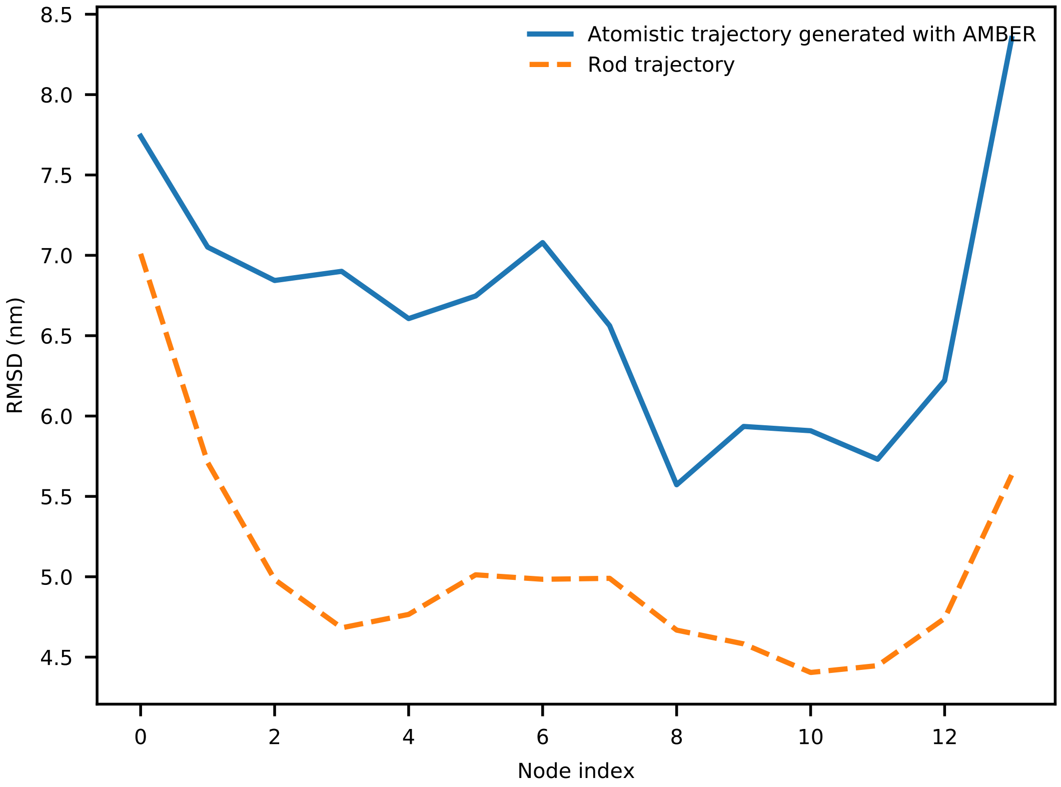


Figure 3.19: Per-element best-fit RMSD for the atomistic and rod structures. Note: the $\boldsymbol{B}$ matrix shown in Figure 3.13 is not a property of the nodes at the edge of the rod, so number of nodes and node indexing differ between that figure and this one.

The equilibrium rod structure is effectively a straight line. If we assume that the rod is bending about a single point, the per-element RMSD of the rod simulation is easy to understand: the equilibrium structure will, on average, intersect the bent structure more around nodes 3 and 10 (the minima on Figure 3.19) and will diverge more near the ends.

A similar, but asymmetric profile is visible for the atomistic simulation. Although flexibility of the two sides of the rod could be genuinely asymmetric, it's also possible that the trajectory is undersampled. This could explain the lower RMSD at each end, as it did not enter extreme enough bend angles, and also the asymmetry — one end stayed straighter than the other, so equilibrium structure aligned to it preferentially.

Chapter 4

Connecting Rods to Tetrahedra and Kinetochore-MT Binding

Chapter 3 saw the use of the KOBRA elastic rod model to investigate the dynamics of coiled-coils. This chapter will extend that model, allowing it to simulate of coiled-coils coupled to globular regions. In particular, we will return to NDC80c, this time including the small globular domains at each end. Similar ‘dumbbell’ shapes are also found in myosin, dynein and fibrinogen.

Fluctuating Finite Element Analysis (FFEA) was introduced in chapter 1 as a coarse-grained algorithm for simulating globular proteins. We will now outline a method for coupling KOBRA rods to tetrahedral FFEA elements in a way that is computationally inexpensive and allows for the construction of mesoscale systems, such as the kinetochore.

Once this algorithm has been developed, we will construct a simplified lateral attachment model (see section 1.2 for a description of lateral attachment) in order to study the microtubule-binding behaviour of Ndc80C and how this behaviour as affected by its cargo and the hinge region.

Some of the literature reviewed in chapter 1 briefly discusses coupling of elastic rods to exterior objects. Pai et al. [90] consider the system to be a standard boundary value problem and solve the equations with the assumption that the stress at one end of rod is always known, and the position and orientation of the other end is always known. These constraints are specific to their problem, and would be difficult to apply to molecules such as Ndc80C.

Bergou et al. [93], upon which the model outlined in chapter 2 is based, use a method based on the minimisation of the elastic energy, with the constraints enforced using a manifold projection. This method requires the energy to be found iteratively, and depends upon the kinetic energy of the rod, making it unsuitable for massless rod simulations.

For KOBRA, we propose an algorithm which is simple to implement, with no modification to the existing design of KOBRA or FFEA, and computationally inexpensive, able to run in linear time. After this model has been described and validated, we will use it to perform preliminary studies of Ndc80C bound to a microtubule.

4.1 Exchange of Forces

FFEA represents globular regions as 3D volumetric meshes constructed from linear tetrahedra. Each mesh is referred to in the FFEA nomenclature as a 'blob'. At the surface, the faces and nodes of these tetrahedra can be considered to be either 'surface faces/nodes' or 'internal faces/nodes'. In this example, we will consider an element with one surface face, three surface nodes, three internal faces and one internal node (also called the 'back' node), as shown in Figure 4.1. This figure also indicates the notation used for describing nodes and edges.

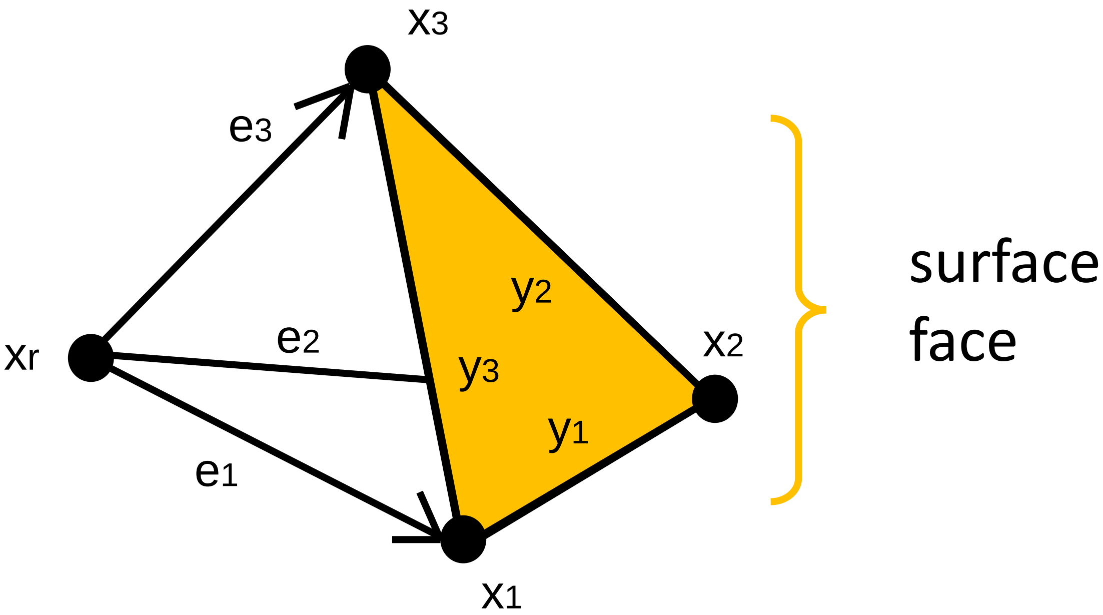


Figure 4.1: Notation to be used in describing a tetrahedron.

We wish to connect an FFEA element and a KOBRA rod together, such that forces in every degree of freedom are transmitted through the connection. These forces are mediated through the use of an intermediate rod element, called the 'attachment element'. This allows the rod to 'see' the tetrahedron as another rod element, and also allows the tetrahedron's nodes to experience forces as if they were rod nodes. This attachment will be described in terms of its three degrees of freedom - stretching, bending and twisting. For each degree of freedom, we will need a new way to translate between the geometry of the tetrahedron and that of the attachment element.

4.1.1 Attachment Node

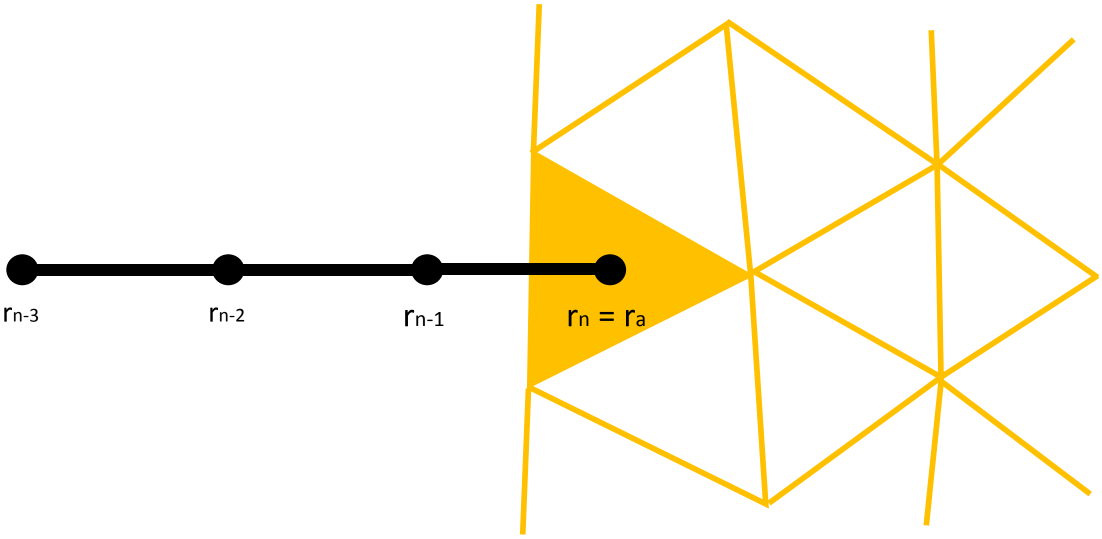


Figure 4.2: The attachment node ($\boldsymbol{r}_a$) within the attachment tetrahedron (shown in yellow).

When an attachment is created, the node at either the start or the end of the rod becomes the 'attachment node'. In these examples, the end node ($\boldsymbol{r}_N$) is the attachment node. This node is positioned at some prescribed point relative to the internal edges of the tetrahedron, as illustrated in Figure 4.2. The attachment node $\boldsymbol{r}_a$ is

$$\boldsymbol{r}_a = \sum_{i=0}^{3} w_i \boldsymbol{e}_i. \tag{4.1}$$

Where $\boldsymbol{r}_a$ is the attachment node, $\boldsymbol{e}_i$ are the edges and w_i are arbitrary weights. For example, if $w_1, w_2, w_3 = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$, the attachment node lies at the centre of the tetrahedron's face, opposite to the back node $\boldsymbol{r}_b$.

4.1.2 Attachment Element

The attachment node only transmits stretch forces. To account for bending and twisting, we define the attachment element $\boldsymbol{p}_a$ (Figure 4.3) and attachment material axis $\boldsymbol{m}_a$. In order to fix the rod direction in the tetrahedron, we first compute the inward normal to the face of the tetrahedron, $\boldsymbol{n}_{face}$.

$$\boldsymbol{n}_{face} = \frac{\boldsymbol{y}_1 \times \boldsymbol{y}_2}{|\boldsymbol{y}_1 \times \boldsymbol{y}_2|} \tag{4.2}$$

Where $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are two edges on the tetrahedron's face (see Figure 4.1). To ensure that the attachment element is always pointing into the tetrahedron[1] (toward the tetrahedron's centroid, $\boldsymbol{C}$), we multiply $\boldsymbol{p}_a$ by a direction factor d,

$$\boldsymbol{n}'_{face} = d\boldsymbol{n}_{face} \tag{4.3}$$

where

$$d = \begin{cases} 1, & \text{if } c_{path} > 0 \\ -1, & \text{if } c_{path} < 0 \end{cases} \tag{4.4}$$

where c_{path} is defined to be

$$c_{path} = \boldsymbol{n}_{face} \cdot \left(\frac{\boldsymbol{r_a} - \boldsymbol{C}}{|\boldsymbol{r_a} - \boldsymbol{C}|} \right) \tag{4.5}$$

[1] This is a detail specific to the FFEA implementation, which does not guarantee the ordering of nodes in tetrahedra.

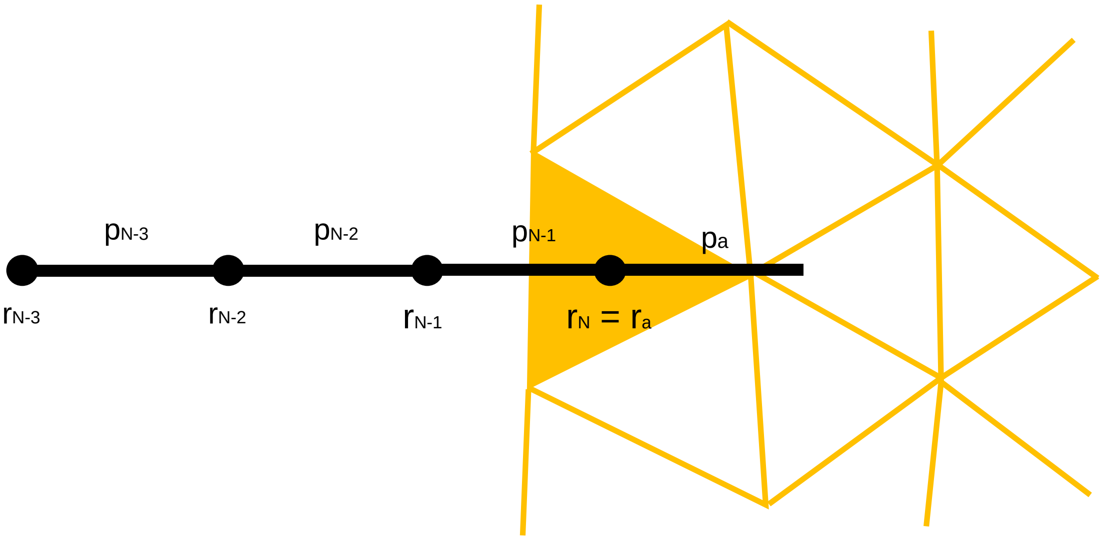


Figure 4.3: Positioning of the attachment element relative to the tetrahedron face (equations (4.2) and (4.3)).

For most purposes, the attachment element is of unit length. However, when calculating the bend energy using the mutual element (see section 2.2.3), a unit-length element would heavily weight the mutual element toward the attachment node. Therefore, it is scaled such that

$$\boldsymbol{n}''_{face} = \boldsymbol{n}'_{face}(2|\widetilde{\boldsymbol{n}}'_{\boldsymbol{face}}| - |\boldsymbol{n}'_{face}|) \tag{4.6}$$

The length of the attachment element is scaled such that it gets shorter if the end element gets longer, and vice-versa. The choice of scaling behaviour is arbitrary, but this preserves approximately correct weighting behaviour with respect to the end element.

Setting the position of the attachment element like this means it is always pointing directly out of the attachment face. In some systems, we may require it to point out of the element at an angle. So, after the orientation of the attachment element has been calculated, it can be rotated by three angles, α, β and γ, using the Tait-Bryan convention in the following order: $X_1 Z_2 Y_3$. Thus, following rotation matrix is constructed:

$$\boldsymbol{R}_a = \begin{bmatrix} \cos(\beta)\cos(\gamma) & -\sin(\beta) & \cos(\beta)\sin(\gamma) \\ \sin(\alpha)\sin(\gamma)+\cos(\alpha)\cos(\gamma)\sin(\beta) & \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta)\sin(\gamma)-\cos_(\gamma)\sin(\alpha) \\ \cos(\gamma)\sin(\alpha)\sin(\beta)-\cos(\alpha)\sin(\gamma) & \cos(\beta)\sin(\alpha) & \cos(\alpha)\cos(\gamma)+\sin(\alpha)\sin(\beta)\sin(\gamma) \end{bmatrix} \tag{4.7}$$

We then rotate $\boldsymbol{n}'_{face}$ (or $\boldsymbol{n}''_{face}$) by the rotation matrix $\boldsymbol{R}_p$ to acquire the final attachment element $\boldsymbol{p}_a$.

$$\boldsymbol{p}_a = \boldsymbol{R}_p \boldsymbol{n}'_{face} \tag{4.8}$$

4.1.3 Attachment Material Axis

The attachment material axis is needed to represent the twisting of the blob relative to the rod. Unlike the attachment node and element, this cannot be computed solely from the current positions of the tetrahedron nodes. Instead, we obtain the rotation matrix $\boldsymbol{R}$ between the original and current tetrahedron configurations, and apply that rotation to the equilibrium material axis. This equilibrium material axis is obtained by parallel transporting (section 2.2.2) the equilibrium material axis from the adjacent element onto the equilibrium attachment element,

$$\widetilde{\boldsymbol{m}}_a = P(\widetilde{\boldsymbol{m}}_{adj}, \widetilde{\boldsymbol{l}}_{adj}, \widetilde{\boldsymbol{l}}_a) \tag{4.9}$$

Where $\widetilde{\boldsymbol{m}}_a$ is the equilibrium attachment material axis, $\widetilde{\boldsymbol{m}}_{adj}$ is the adjacent equilibrium material axis (assuming the connection is at one end of the rod, this is either $i = 1$ or $i = n - 1$), $\widetilde{\boldsymbol{l}}_a$ is the normalised adjacent element at equilibrium, $\widetilde{\boldsymbol{l}}_{adj}$ is the normalised attachment element, and P is an application of parallel transport as defined in equation (4.9).

To obtain $\boldsymbol{R}$, we first compute the Jacobian matrix of the tetrahedron, which is the transformation matrix from a reference tetrahedron with nodes at $x_r = (0, 0, 0), x_1 = (1, 0, 0), x_2 = (0, 1, 0,$ and $x_3 = (0, 0, 1)$.

$$\mathbf{J} = \begin{bmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{bmatrix} \tag{4.10}$$

where $\boldsymbol{J}$ is the Jacobian, x, y and z are dimensions, and subscripts are tetrahedron node indices. We can multiply the Jacobians for the current and equilibrium state of the tetrahedron to get the deformation matrix which transforms from the equilibrium to the current tetrahedron configuration,

$$\mathbf{F_e} = (\mathbf{J}\widetilde{\mathbf{J}}^{-1})^T \tag{4.11}$$

where $\boldsymbol{J}$ is the Jacobian for the equilibrium configuration. The elements of this matrix give us the displacement, rotation and deformation of elements of the tetrahedron. To extract only the rotational component, we can perform a QR decomposition of the matrix using the Gram-Schmidt algorithm [116].

$$\boldsymbol{F} = \boldsymbol{QR} \tag{4.12}$$

This gives us two matrices. The orthogonal matrix $\boldsymbol{Q}$ defines the shape of the tetrahedron, while the triangular matrix $\boldsymbol{R}$ defines its orientation. $\boldsymbol{R}$ is the rotation matrix that rotates the equilibrium tetrahedron onto the current one, so this rotation matrix can also be used to rotate $\widetilde{\boldsymbol{m}}_a$ onto $\boldsymbol{m}_a$ (Figure 4.4),

$$\boldsymbol{m}_a = \boldsymbol{R}\widetilde{\boldsymbol{m}}_a \tag{4.13}$$

When the end/attachment node position is updated, the attachment material axis is forced to be perpendicular to the attachment element using the method described in section 2.2.5.

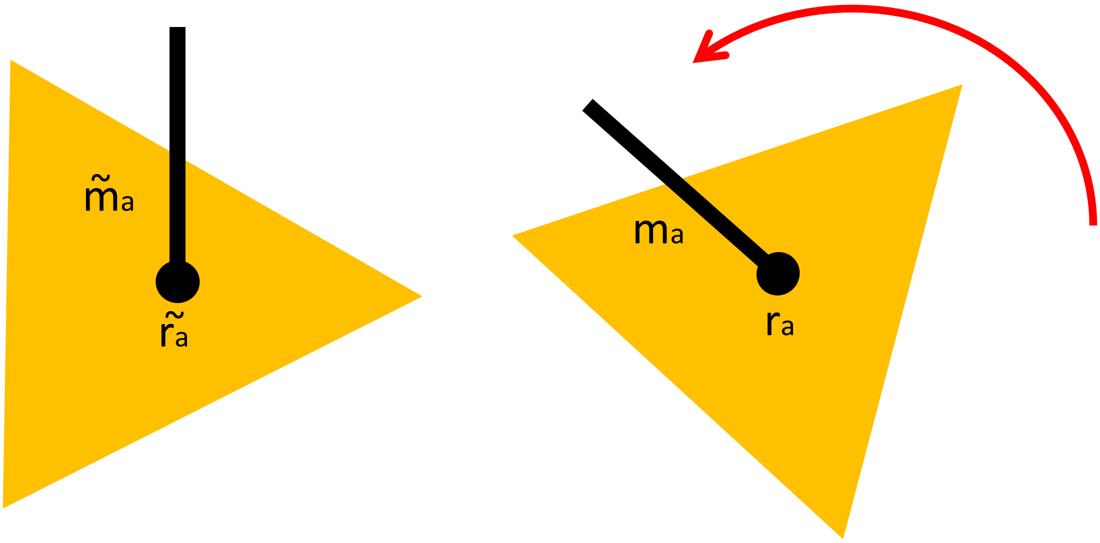


Figure 4.4: Update of the material axis from the equilibrium configuration ($\widetilde{\boldsymbol{m}}_a$ and $\widetilde{\boldsymbol{r}}_a$) to the current one ($\boldsymbol{m}_a$ and $\boldsymbol{r}_a$) as detailed in equations (4.10-4.1.3).

4.1.4 Attachment Energy

Energy type	Mediated by
Stretch	Attachment node
Bend	Attachment element
Twist	Attachment material axis

Table 4.1: Summary of the degrees of freedom in the rod-blob connection.

The KOBRA rod has now been extended to contain an extra element and material axis, representing the connection to the tetrahedron. Additionally, the end node $\boldsymbol{r}_n$ is replaced by the attachment node $\boldsymbol{r}_a$, so the position of this node is determined by the position of the connected tetrahedron, and therefore the blob is transmitting forces onto the rod.

To resolve forces from the rod to the blob, we use a method similar to that described in section 2.2.4. The nodes of the attachment tetrahedron now experience a gradient in energy with respect to their position, from the rod that is connected to them,

$$\boldsymbol{F_i} = \frac{\partial E}{\delta x_i}. \tag{4.14}$$

If we compute this energy using the equations described in section 2.2.3, updating the position of the rod elements as described in this chapter, and apply the resulting force to the tetrahedron nodes, then the rod will transmit a force back onto the blob.

4.1.5 Implementation Details

Initialisation

For numerical stability at initialisation, these attachments must start at or close to equilibrium. An attachment can be considered either ‘rod-to-blob’ or ‘blob-to-rod’. This does not affect how forces are resolved in the connection, only how the attachment is initialised.

Rod-to-blob attachments ignore the initial position and rotation of the blob. Instead, the blob is translated and rotated such that, at equilibrium, the attachment node $\boldsymbol{r}_a$ is at the end ($\boldsymbol{r}_n$) of the rod, the blob’s face normal is parallel to the attachment element $\boldsymbol{p}_a$, and there is no twist from the attachment material axis, between $\boldsymbol{p}_a$ and $\boldsymbol{p}_{n-1}$.

A blob-to-rod attachment will ignore the initial position of the rod, and translate/rotate the rod such that, at equilibrium, the attachment element $\boldsymbol{p}_a$ is parallel to the face blob normal, the first node r_0 is at the position of the attachment node r_a and there is no twist between the attachment material axis $\boldsymbol{m}_a$ and the first material axis m_0.

To rotate the rod or blob, we use the method described in 2.2.2 to compute the vector that rotates the adjacent rod element ($\boldsymbol{p}_0$ for blob-rod attachments, $\boldsymbol{p}_{n-1}$ for rod-blob attachments) onto the current attachment element $\boldsymbol{p}_a$.

For a blob-to-rod attachment, the rod is then translated by the vector $\boldsymbol{t}$, where

$$\boldsymbol{t} = \boldsymbol{r}_0 - \boldsymbol{r}_a \tag{4.15}$$

For rod-to-blob attachments, the blob is translated by $-t$ instead.

This initialisation step occurs for every attachment in the system. Each attachment has an attachment priority value that determines whether it will be initialised before or after other attachments.

Algorithm Steps

The steps taken in initialisation and one timestep are as follows:

- Initialisation
    - Initialise attachments in order of priority (section 4.1.5)
    - Position objects (and attachments) in simulation box
    - Set initial values (e.g. $\widetilde{\boldsymbol{J}}$)
- Timestep
    - Update blob boundary conditions, calculate nearest neighbours
    - Reorientate attachment (sections 4.1.1 through 4.1.3)
    - Set position of connected rod end node to attachment node
    - For each node in the attachment tetrahedron (section 4.1.4):
        - Permute node position
        - Compute new attachment element
        - Compute energy and force on node from rod
    - Calculate blob forces (internal/external)
    - Update blob positions

Unlike other algorithms, particularly those that make use of optimisation algorithms, resolving a rod-bob connection happens in O(1) time, and multiple rod-blob connections scale in linear time. Multiple connections are independent and so the algorithm as also embarrassingly parallel. The memory consumption is also negligible: a single extra node, element and material axis. The downside of this method is that its numerical stability depends upon the stability of the connected element — strong forces at the connected element can cause the element to invert, or prevent the conjugate gradient solver from converging. The easiest solution to this problem is to ensure that the size of the connected element is fairly large. For typical rod parameterisations, such as those discussed in chapter 3, the longest side should be no shorter than $14\AA$.

In the same vein as section 2.4, the correctness of this algorithm is supported by a set of unit and end-to-end tests. These tests can be found in the Appendix, section B.2.

4.2 Simulations of Larger MT-Kinetochore Systems

In this section, we will run some preliminary simulations to show the capabilities of the rod-blob connections, examine how Ndc80C can form attachments to microtubules, and study how these attachments are affected by the flexural stiffness of Ndc80C and the presence of its cargo. This section will focus solely on lateral attachment, in which Ndc80C binds to the side of a microtubule, pointing away from the microtubule surface [117], which can be examined without a dynamic microtubule model.

4.2.1 Research Questions

Prior experimental research into Ndc80C and the kinetochore suggests a number of open questions, into which simulations may provide insight.

Do stiffer Ndc80Cs bind preferentially? If stiffer, or straighter, Ndc80Cs bind preferentially with microtubules [10] [118], we might expect to see this reflected in the bend angle distributions of Ndc80C molecules bound to microtubules compared to those which are not bound. However, we might also expect bound cargo to restrict the possible bend angles that Ndc80C can take. Could it be that Ndc80Cs which are bound to microtubules bend less because of their cargo, and not that less-bent Ndc80Cs bind preferentially to microtubules?

Does the presence of cargo affect Ndc80C's MT-binding affinity? The inner kinetochore attaches to Ndc80C at the Spc24/Spc25 end, connecting it to chromosomes by the centromere. The disparate Ndc80C complexes are all brought together by a superstructure made from MIND complexes [5]. In understanding the dynamics of Ndc80C, our primary concern is how the presence of this cargo can exert forces on Ndc80C and affect its affinity to bind to the MT. There is some evidence that straight Ndc80Cs adhere more easily, but this straightening may not be an intrinsic property of Ndc80C, rather the result of tension applied by the MIND complex [119].

Effect of the flexible hinge on MT-binding As discussed in chapter 3, Ndc80C contains a central flexible hinge that accounts for the wide range of conformations that the molecule can take. The flexibility of Ndc80C can also regulate its ability to bind and unbind from microtubules [10] [118]. It is therefore worth considering how the flexible hinge might affect the profile of energies resulting from attractive forces at the surface of the microtubule. In atomistic molecular dynamics, we are used to thinking of binding interactions as a product of the binding energy between two molecules. However, the effective binding interaction between Ndc80C and the MT doesn't just depend only the specific interaction between the globular domains and the MT surface, it also depends on the free energy of the entire Ndc80C molecule, and thus the hinge parameters. Does the flexible hinge increase this binding affinity, or decrease it?

Does Ndc80C bind better in clusters? There is experimental evidence that the attachment becomes more stable as the number of Ndc80C molecules increases, with three being the minimum number to form a stable attachment [17] [43]. Would increasing the number of Ndc80C molecules increase the number of binding events per molecule?

Binding between Ndc80C's unstructured tail and the DAM1 ring Ndc80C has an unstructured tail, and there is some evidence that this greatly increases the binding affinity, not directly with the MT but with the DAM1 ring. We lack a DAM1 ring model and a way to represent this loop — so this simulation would require atomistic studies followed by a parameterisation based on point-point interactions or a rod model that can support FFEA-style rod-surface interactions (see section 5.3 for a discussion of this). A DAM1 ring model may prove necessary in the future as Ndc80C binds to DAM1 during side-on attachment [120] [121].

4.2.2 Parameterisation

In order to address some of the above questions, we have set up a series of initial simulations designed to mimic attachment of the kinetochore complex to the microtubule. The parameterisation of the system used in these simulations is based on low resolution ($4.3\mathring{A}$) structural information for laterally attached kinetochores (Figure 4.5). All-atom structures cannot be recovered from this data, and different protein complexes within the kinetochore are not easily distinguished from one another.

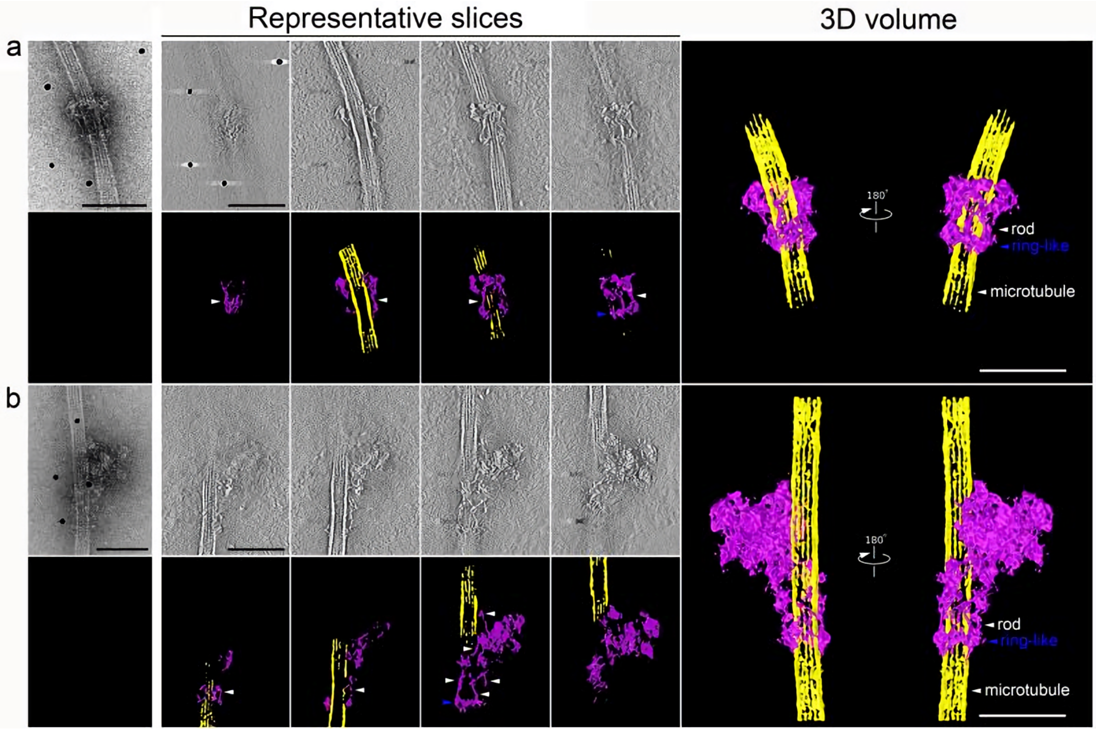


Figure 4.5: Three dimensional images of the kinetochore (pink) bound to a microtubule (yellow) captured using electron tomography [6]. Scale bars are $100nm$.

Given their resolution, the fact these tomographic images cannot be used to identify specific protein components is expected. There is insufficient information to create FFEA meshes and KOBRA rods directly. While it is clear that Ndc80C binds to the MT and connects to a larger cargo, we will need to make certain assumptions about how the system is laid out in order to set up a coupled FFEA/KOBRA simulation. We assume that:

1. Ndc80C begins perpendicular to the MT surface.

2. There are three Ndc80 complexes bound to the MT (this is the minimum number than can be bound whilst retaining the correct chromosome segregation behaviour [122]).

3. Ndc80C binds to the MT at binding sites separated by $8.5\mathring{A}$. The interatomic potentials are parameterised using the attractive part of the 6-12 Lennard-Jones potential [123],

$$U_{i,j}(r) = \epsilon_{i,j} \left[\left(\frac{\sigma_{i,j}}{r} \right)^{12} - 2 \left(\frac{\sigma_{i,j}}{r} \right)^{6} \right] \tag{4.16}$$

   with parameters of $\sigma_{i,j}$ (the equilibrium separation) $= 1 \times 10^{-15} m$, and $\epsilon_{i,j}$ (the depth of the potential well) $= -1 \times 10^{-10} J$. In previous simulations it was shown that this affinity provides a reversible binding [124].

4. The structure of the inner kinetochore does not broadly affect the MT-binding behaviour of Ndc80C, except that it imparts a viscous drag. We will represent Ndc80C's cargo as a sphere, mirroring in vitro nanobead experiments [125].

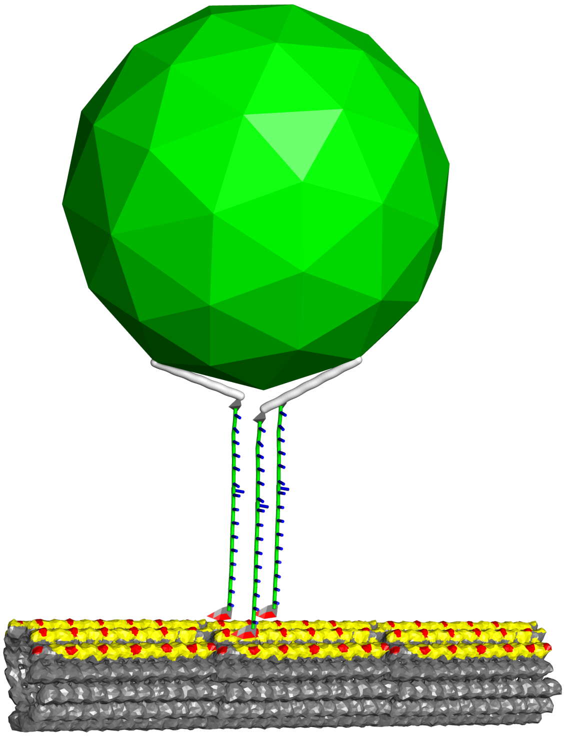


Figure 4.6: FFEA system depicting side-on MT attachment rendered in PyMOL. The sphere radius is $50nm$, the three Ndc80 complexes are parameterised as in chapter 3, and the MT is parameterised as it is in [19].

Figure 4.6 shows the assembled KT-MT system. Three Ndc80 complexes, each comprised of one rod and two small globular domains, are positioned above the surface of the MT. The rod uses the parameterisation from chapter 3. The globular domains are created from the ‘Bonsai’ Ndc80C structure [17] using the FFEA coarse-graining method [1]. Rod-blob connections between the rod and globular domains are set up using the alignment of the the ‘Bonsai’ structure as a guide. The Ndc80/Nuf2 end is at the MT surface, while the Spc24/Spc25 ends are attached by harmonic restraints (shown as grey lines) to a sphere. In Figure 4.6, the surface triangles for the meshes are coloured according to their surface-surface interaction parameters. The red-coloured triangles have a short-range Lennard-Jones-like attraction to other red-coloured triangles, so are responsible for attracting the Ndc80C to the surface of the MT. The parameters of this interaction are listed on earlier in this section, on page 108. The full surface is shown in Figure E.1. Some of the other FFEA parameters are listed in table E.5.

A set of simulations were performed using the system shown in Figure 4.6. A summary of these can be found in table 4.2.

Simulation	MT	Cargo	Hinge
Control	✓	✓	✓
No hinge	✓	✓	✗
No cargo	✓	✗	✓
No cargo, no MT	✗	✗	✓

Table 4.2: Summary of the simulations performed using the Ndc80C-MT system shown in Figure 4.6.

The 'Control' simulation is of exactly the system shown in Figure 4.6, containing the microtubule, cargo, and a flexible Ndc80C parameterised as it was in chapter 3. The 'No hinge' simulation removes Ndc80Cs flexible hinge, replacing it with a region with average bending modulus parameters calculated from the rest of Ndc80C. The 'No cargo' simulation removes the spherical cargo and harmonic restraints, but keeps the rest of the system as shown in Figure 4.6. The 'No cargo, no MT' simulation contains no MT or cargo, it features only a single Ndc80C molecule, a KOBRA rod attached to two globular domains.

4.2.3 Preliminary Results

This section will discuss the results of the Ndc80C simulations and how they relate to questions about Ndc80C's MT-binding behaviour and the stability of the MT-KT attachment. These are initial results, presented as an illustration of possible lines of analysis using KOBRA.

Does the Presence of the MT and Cargo Restrict the Possible Conformations of Ndc80C?

If Ndc80C is bound at one or both ends, we might expect that to constrain its motion, either through viscous drag (cargo) or steric repulsion (the MT), and restrict the range of conformations that it explores. This might inform us of how the presence of the cargo affects Ndc80C's MT-binding ability, and whether the observed 'stiffness' of bound microtubules might actually be a result of the tension stabilising the interaction, and not vice-versa. To test this, we compare distributions of kink angles from the systems listed in table 4.2 using the method described in section 3.4.1.

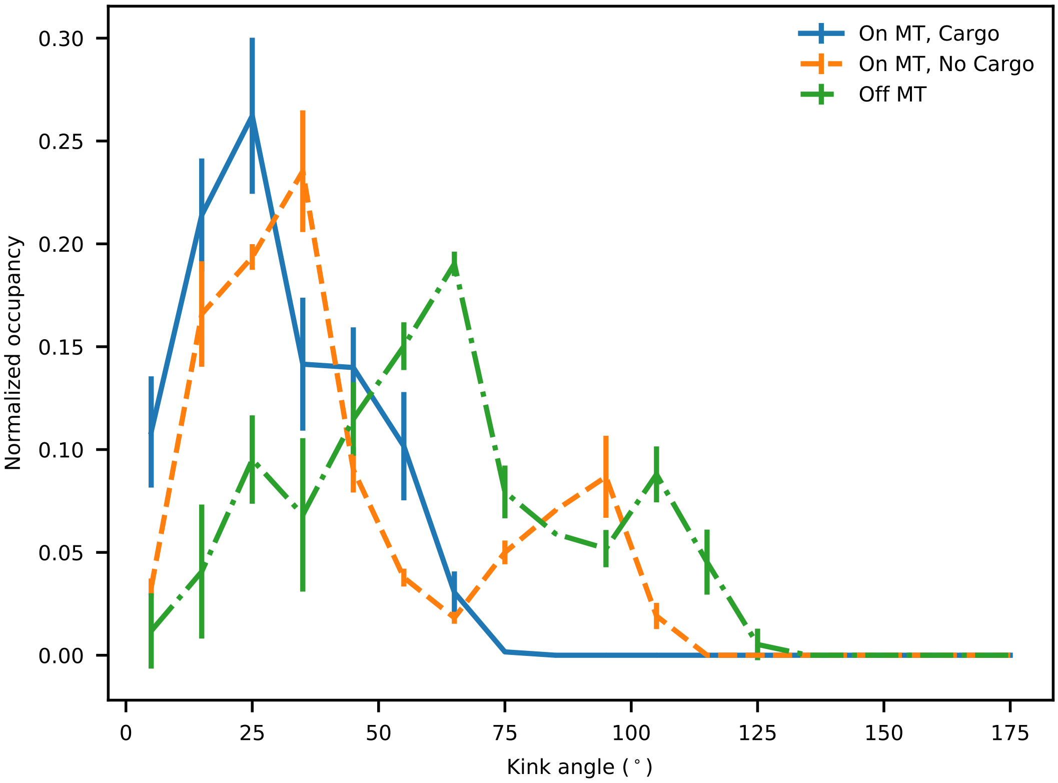


Figure 4.7: A distribution of kink angles taken by three different systems. The kink angles are computed using the method described in section 3.4.1. In simulations with multiple Ndc80Cs, the mean bend angle is used.

Figure 4.7 compares the kink angles of three different systems, one with an Ndc80C molecule on its own (No cargo, no MT), one in which three are attached to a microtubule (No cargo), and one in which three are attached to a microtubule with the spherical cargo (Control). The motion of Ndc80C is restricted by its interactions with the microtubule, and even more constrained in the presence of cargo. There are two factors behind this: first, binding Ndc80C means that the molecule is under more tension. Second, the steric interactions with the microtubule restrict the motion of the globular domain at that end. Do these factors affect Ndc80C's MT-binding properties? To find out, we can investigate MT-binding behaviour of Ndc80C more closely by looking at the magnitude of the attractive surface interactions, and the role of Ndc80C's flexible hinge.

Effects of Bound Cargo and Tension

If bound cargo (and thus, tension) affect the MT-binding behaviour of Ndc80C, we might expect to see this reflected in the energy of the short-range interactions between the globular ends of Ndc80C and the MT surface. The existing trajectories were profiled in order to observe these energies and develop a picture of Ndc80C's ability to diffuse and locate a binding site. We first compare these energies for two simulation trajectories, 'Control', with cargo, and 'No cargo', without it.

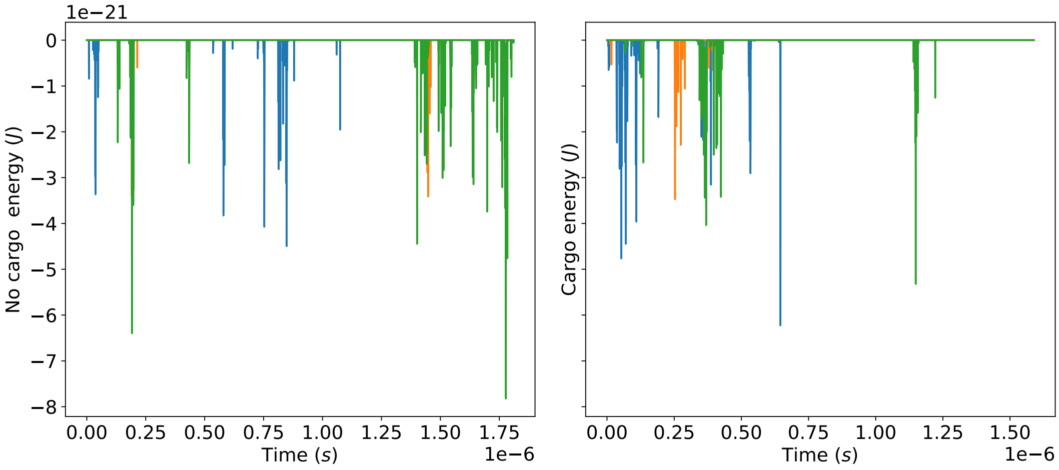


Figure 4.8: Van der Waals interaction energies between Ndc80C's globular domains and the MT surface as a function of time, comparing a trajectory with cargo (left) to one without (right). Different colours represent different Ndc80C molecules.

Figure 4.8 shows how the short-range interaction energies evolve throughout a trajectory. The lines show the instantaneous total van der Waals interaction energy for the three Ndc80C globular domains, each frame. A trough on this graph indicates that Ndc80C came close to a binding site. The energy is calculated using the 6-12 potential shown in equation (4.16).

The right-hand side of the plot shows a simulation with no bound cargo - the sphere, normally bound to the Spc24/Spc25 end of Ndc80C, is not included. They both show a similar number of binding events, though the one with cargo has slightly more. It's possible that the presence of cargo anchors the far end of Ndc80C in place, restricting the conformational space it can explore, but allowing it find a binding site more easily. However, the interaction is not strong, and Ndc80C can unbind easily. This can be observed in the second half of the 'Control' trajectory, as Ndc80C has drifted away from the microtubule entirely. This suggests that, to form a stable attachment, especially in the presence of tension, Ndc80C must interact more strongly with the MT than other MT-binding proteins such as cytoplasmic dynein. Alternatively, more than three Ndc80C molecules are required in order to form a stable attachment. The relative lack of binding events and their uneven distribution also suggest that conformational space is not well-sampled, even in a lengthy (1.8 μs) trajectory.

We can visualise the paths that Ndc80C's globular domains take throughout the trajectory by plotting the positions of their centroids in cylindrical co-ordinates, with respect to the microtubule axis, for each frame in the trajectory.

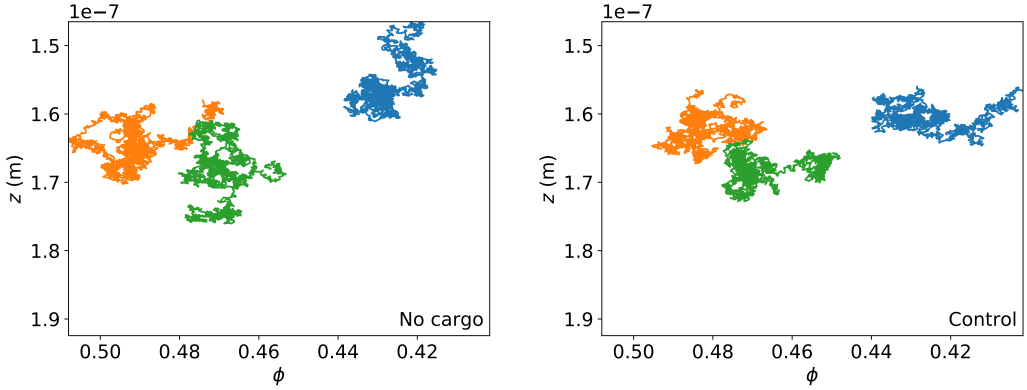


Figure 4.9: Trails of Ndc80C's globular domains, represented in cylindrical co-ordinates, the azimuthal angle ϕ and the height z. Each colour represents a different Ndc80C molecule. The 'No cargo' system is on the left, 'Control' on the right.

Figure 4.9 compares the paths of Ndc80C's globular domains for two simulations, the control (right), and the simulation run with no cargo (left). The trajectories are not far from being random walks, although clearly all three molecules show an affinity for certain regions of the MT. The rightmost Ndc80C in the no cargo trajectory also appears to find a different binding site. We can compare this interaction quantitatively by looking at the third cylindrical co-ordinate, ρ, the radius, and how it is distributed for the different trajectories.

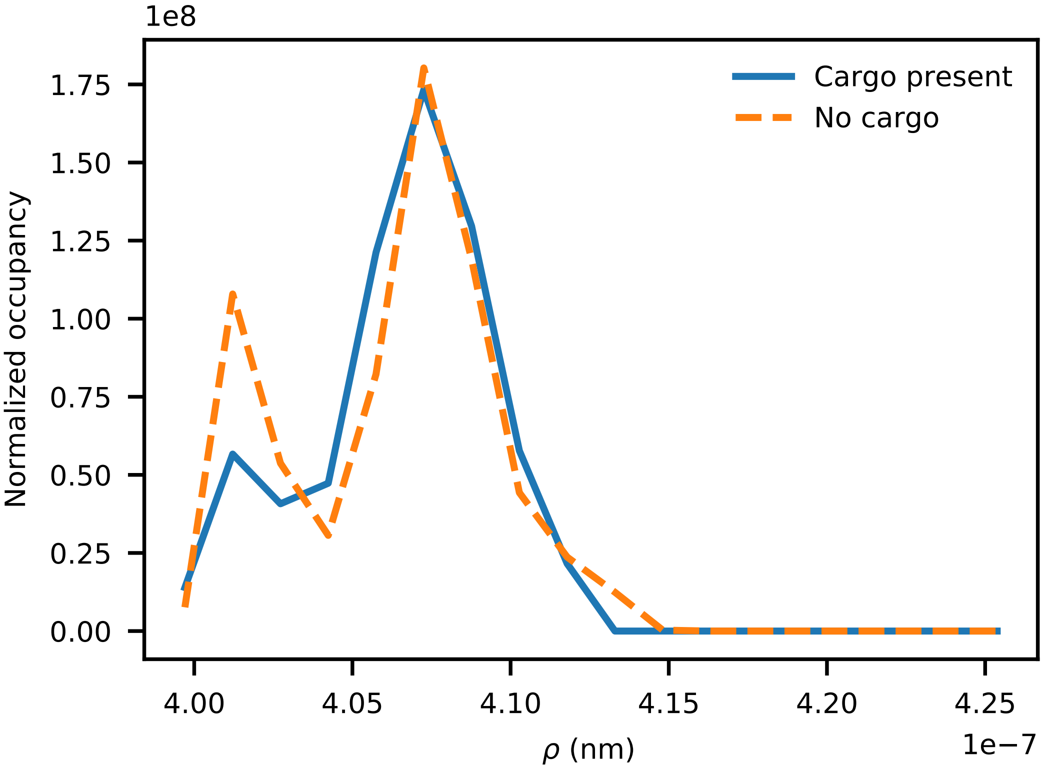


Figure 4.10: Distribution of radii of the Ndc80C Ndc80/NUF2 globular domain. The radii are given in terms of the cylindrical co-ordinate ρ, the radius from the center of the cylinder.

The cargo may affect the ability of the Ndc80C to move around the surface of the MT (Figure 4.8), but it does not inhibit its ability to move radially (Figure 4.10), which is almost identical with and without the cargo. The motion of Ndc80C in these trajectories is driven entirely by thermodynamics, but we might expect the cargo to inhibit the motion due to kinetics more so than the diffusive motion. Real cellular environments are densely-packed, there may be more to inhibit the motion of the kinetochore's cargo than just Stokes drag. Finally, conformational space may not be well-sampled, even in a trajectory 1.8μs in length.

Ndc80C's Flexible Hinge

This section will focus on the Ndc80C molecule parameterised without a flexible hinge (‘No hinge‘). This model has an identical structure to the hinged Ndc80C, but the values of the $\boldsymbol{B}$ matrix at the hinge have been set to the average for the whole molecule. The surface-surface interaction energies from the resulting trajectory are shown in Figure 4.11.

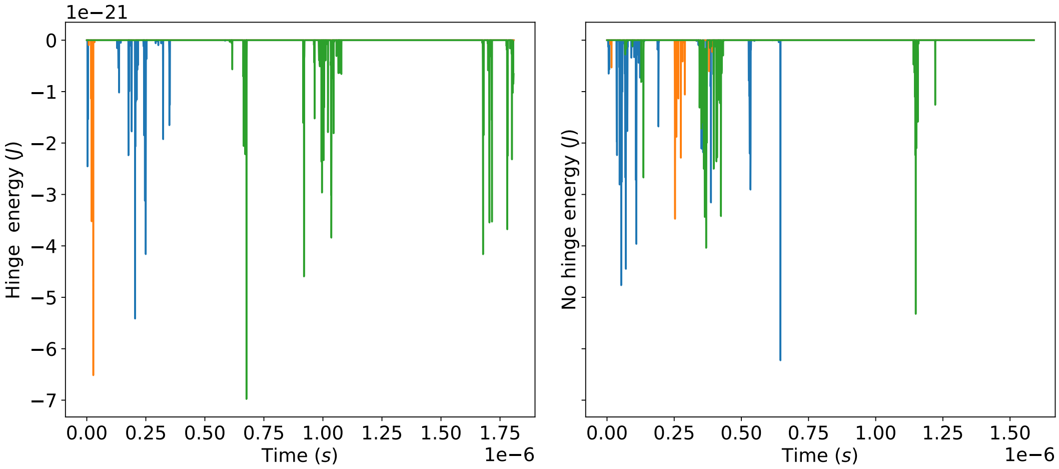


Figure 4.11: Short-range (van der Waals) energy between Ndc80C's globular domains and the MT surface as a function of time, comparing a trajectory with a hinge (left) to one without (right). Different colours once again represent different Ndc80C molecules.

The hinged and unhinged molecules show a similar number of binding events. However, the binding events for the unhinged Ndc80Cs are denser at the start of the trajectory and become more sparse as it evolves. This could suggest that the hinge-free Ndc80Cs are drifting away from the MT surface. It's possible the motion of the cargo dominates the position of the Ndc80Cs, and the force from the cargo's motion — without the hinge to mediate it — is stronger than the attraction from the binding site. We can again investigate this more closely by looking at the radius of Ndc80C's globular domains from the microtubule.

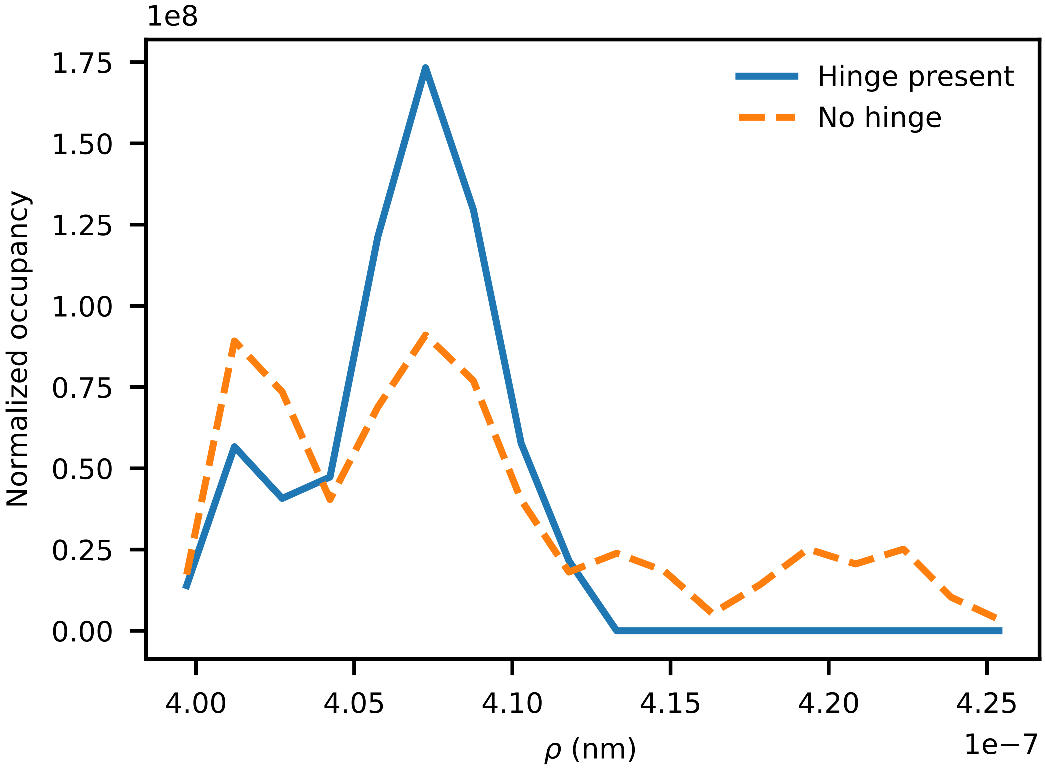


Figure 4.12: Distribution of radii of the Ndc80C Ndc80/NUF2 globular domain. The radii are given in terms of the cylindrical co-ordinate ϕ, the radius from the center of the cylinder.

Figure 4.12 shows the average radius of Ndc80C's globular domains from the center of the microtubule. The higher peak of the hinged molecule suggests a more stable attachment. On average, the globular domains are farther from the MT without the hinge. This reinforces the idea that the hinge allows Ndc80C to bind more easily to the MT surface. The distribution of radii is altered far more radically than in Figure 4.10. This suggests that the difference in Figure 4.11 is not necessarily due to forces exerted by the cargo, and it may simply be that the flexible hinge allows Ndc80C to access the binding site more easily. Does this correspond to an increase in conformational space, across the MT's surface, explored by the molecule?

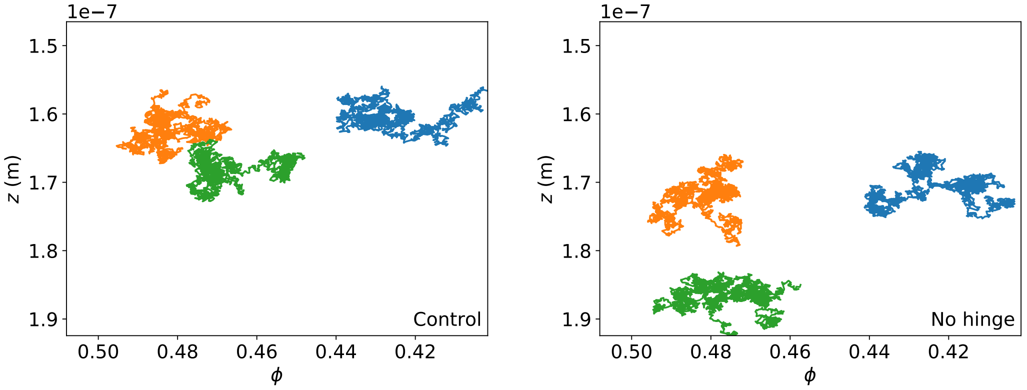


Figure 4.13: Trails of Ndc80C's globular domains, represented in cylindrical coordinates, the azimuthal angle ϕ and the height z. Each colour represents a different Ndc80C molecule. Left: control. Right: no hinge.

Figure 4.13 shows the results of this comparison. The molecule with the hinge and without the hinge do not stray any farther from their original positions. This would suggest that the flexible hinge allows Ndc80C to stay bound while experiencing tensile forces from its cargo, but it does not more easily allow it to find a binding site in the first place.

4.2.4 Future Work

Although the scope of this investigation was limited, the results do point towards several interesting research questions. Additionally, there are many unanswered questions from the literature that were not considered when building these simulations. This is partly due to the incomplete feature-set of KOBRA. Neither cluster binding nor the DAM1 ring are represented. It is not yet possible to represent the physics present in these systems using KOBRA, as it lacks a model of short-range interactions (such as Lennard-Jones and steric interactions) between rods and rods, or rods and blobs (for a discussion of these, see sections 5.2 and 5.3). Any emergent behaviour of densely-packed Ndc80C is likely driven or affected by self-interaction and self-intersection — Ndc80Cs becoming tangled together — and the results would not be meaningful without this. Similarly, the DAM1 ring's behaviour depends upon a combination of blob-blob and rod-blob interactions, and would require a fully-modelled DAM1 ring and a careful parameterisation of Ndc80Cs unstructured tail.

As a follow-up the preliminary results shown here, we propose the following:

- Multiple repeat trajectories for all simulations. As discussed in section 3.4.2 of chapter 3, the conformational space of biological molecules is extremely large [114], and a single simulation showing some binding behaviour (or the lack thereof) is not enough to establish a pattern. More simulations would also allow us to analyse binding events via a histogram.

- A more thorough investigation of Ndc80C's clustering behaviour. Varying the number of Ndc80C molecules could affect the strength of the attachment [17] [43]. This could be combined with an investigation into the MT parameters or tension to produce a phase space diagram showing when MT-binding can occur.

- Parameterisation of Ndc80C's tail. There is evidence [121] [120] that this tail increases the binding affinity with the DAM1 ring. Simulations of this would require atomistic studies of DAM1 and a tetrahedral DAM1 mesh, followed by a parameterisation based on point-point interactions, or a rod model that can support FFEA-style rod-surface interactions. We also may want to look at DAM1 next as Ndc80C does bind DAM1 in at some point during side-on attachment [39] [47].

- The sphere in these simulations could be replaced with a more accurate model of the inner kinetochore. This model could be created by combining data from partial EM structures as was done in [47].

- The parameters of the short-range interaction could be investigated more thoroughly by varying the number of Ndc80Cs, the strength of the interaction, and applying a force to the sphere to simulate the tension resulting from the inner kinetochore and cargo. This force would likely be between $2-10pN$, the lower bound is given by thermal noise; the upper bound is the magnitude of forces present in molecular motors. FRET measurements estimate it to be of order 1.36 to $7.44pN$ depending on the phase of the cell cycle [126]. Finally, note that the interaction between Ndc80C and the MT (possibly through DAM1) is much stronger after the transition to end-on attachment; side-on attachment should not be so stable that the transition to end-on attachment cannot take place.

Chapter 5

Conclusion and Future Work

In this chapter, we will summarise the progress that has been made so far, and discuss ways to extend the model and the kinds of systems that these features could be used to examine.

5.1 Summary of Progress

In chapter 2, we discussed the development of an elastic rod model that could be used to represent slender biological macromolecules. This was based on work by *Bergou et al.* [93], but heavily modified to include an extra degree of freedom (stretching), bending about a single-element hinge, intrinsic twisting, and new dynamics incorporating stochastic thermal noise and viscous drag. As we have seen, biological elastic rod models are nothing new, but KOBRA exists at a rarely-explored length scale between all-atom molecular dynamics and models for filaments and fibrils with explicitly defined radii, such as microtubules and actin. Its particular feature-set makes it ideal for simulating coiled-coils and stable alpha-helices at the mesoscale.

In chapter 3, we turned to examine Ndc80C on an all-atom scale. The creation of this Ndc80C structure was informed by a combination of X-ray crystallography and secondary structure prediction. We were also able to observe the role of the hinge region in decreasing the stiffness of Ndc80C, which had previously been theorised. We described a novel method to extract the rod parameters of Ndc80C from the magnitude of the local fluctuations in the all-atom model, and how to incorporate these measurements into a coarse-grained rod model. However, while this model was validated by comparison with all-atom simulations and negative-stain EM, the length scale of the simulations performed did not increase. While the timescales did increase, the accuracy of the dynamics was still constrained by the timescale of the all-atom simulations.

In chapter 4, we extended the rod model include connections to tetrahedral meshes such as those found in the FFEA algorithm, which can be used to create larger, more complex mesoscale systems, such as the Ndc80C-MT system, created such a system, and observed its MT-binding behaviour, and how it was affected by the presence of cargo and the parameterisation of Ndc80C's flexible hinge.

Although we have only examined Ndc80C, no aspect of the method described in this thesis is specific to Ndc80C. KOBRA is currently being used to understand both SMC complexes (Figure 5.1), involved in chromosome condensation [127], and Myosin-V, a motor protein [128].

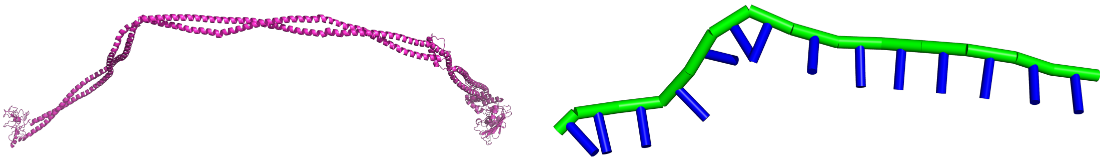


Figure 5.1: Parameterisation of the SMC complex [20] from all-atom (left) to KOBRA (right) by Samantha Coffey. Rendered in PyMOL [16].

An integral part of the work undertaken in this project — and one that is often neglected, due to the time and effort it requires — is the development of robust, well-engineered software, validated with unit and integration tests. In physics software, unit and integration tests guarantee that the software works exactly like the algorithms described in papers, theses and presentations. Low-quality software produces science that can't be tested or replicated. KOBRA includes a comprehensive set of unit and integration tests, most of which are not described in the main text of this thesis, but short summaries are available in section B. A discussion of the broader software engineering principles behind KOBRA's design can be found in section C. Other software development work undertaken as part of this project includes an FFEA/KOBRA continuous integration server (also described in B), development of the FFEA website (`ffea.bitbucket.io`), documentation for users and developers (`ffea.readthedocs.io`), improvements to the FFEA PyMOL plugin for visualisation, and educational materials for FFEA workshops.

5.2 Rod-rod Interactions

Generally speaking, the systems we have been studying have arrangements of rods that are sparse and straight enough that they are not likely to self-intersect. In cases where they do, the forces that would arise are not included in our algorithm, so we are making the assumption that such forces are not important to the overall dynamics and function of the system. In many systems, this is not the case. For example, the end-on yeast kinetochore system (section 1.2.2) is much more densely-packed, with micotubules splaying outwards instead of abruptly ending. In the human kinetochore (section 1.2.3), Ndc80C interlocks with these splayed-out microtubules. Therefore, the transfer of forces in these systems is either subject to or dependent entirely on collisions and short-range interactions.

In simulations on this scale, there are two classes of forces we wish to represent: steric repulsion, which penalises the overlap of objects, and forces that act at a distance, such as van der Waals attraction, charge and dipolar interactions. Therefore, it is necessary to calculate both the distance between objects and the extent of their overlap. Normally, the van der Waals force is computed using the attractive part of a 6-12 Lennard-Jones potential. For steric repulsion, it is normally not possible to represent the extreme hardness of the potential from the Pauli exclusion principle. FFEA uses a softer steric repulsion force proportional to the overlapping volume of two tetrahedra, so this is also the method we will examine in relation to KOBRA.

Detection and resolution of collisions between 3D objects is generally broken into three steps. First, subdivision of space. Then, calculation of distance or volume overlap, then calculation of force.

A naive collision detection algorithm will scale as $O(N^2)$, where N is the number of colliding objects in the system. Every time a new object is added, a collision test must occur between it and every other object. To mitigate this, collision detection algorithms subdivide space into cells, only computing collisions between adjacent cells. Normally this comes in the form of either octrees[1] or evenly-spaced cuboids, of which FFEA uses the latter. In systems of highly variable density, octrees give better performance, but the systems targeted by FFEA usually do not contain large voids.

The solution to the second problem, calculating the distance or volume intersection between two objects, depends upon the geometric primitives used to represent the objects.

[1] An octree is a data structure in which space is recursively subdivided into cubes. Each level of recursion subdivides one cube into eight. Octrees are particularly useful for representing three-dimensional volumes of variable density, as fewer recursions can be used to represent less dense areas.

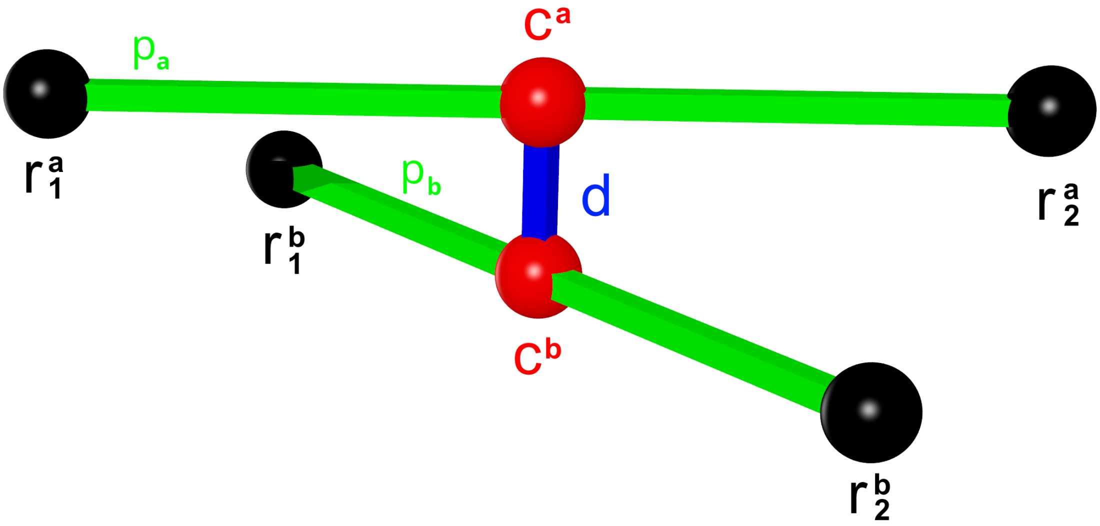


Figure 5.2: Shortest distance between two skew lines.

For example, KOBRA rod elements could be considered to be infinitely thin skew lines (Figure 5.2), in which case the distance between them is analytically solvable [129], and they have no volume, so cannot intersect. The shortest distance, d, between two skew lines, is given by

$$d = \left| \left(\frac{\boldsymbol{p}_a}{|\boldsymbol{p}_a|} \times \frac{\boldsymbol{p}_b}{|\boldsymbol{p}_b|} \right) \cdot (\boldsymbol{r}_b - \boldsymbol{r}_a) \right|, \tag{5.1}$$

where $\boldsymbol{p}^a$ and $\boldsymbol{p}^b$ are the elements, and $\boldsymbol{r}^a$ and $\boldsymbol{r}_b$ are the nodes. Hence, it would be straightforward to introduce a soft interaction that depended on the distance between rod elements.

If we consider the rod elements to have a non-zero volume, distances and volume intersections are more difficult to compute. For example, we could consider the rods to be a series of spheres distributed along the rod axis, whose intersection volume and distance apart can be computed analytically. However, in order to prevent rods from passing through one another, each rod element would need to be constructed from many spheres, which would greatly increase the computational cost of collision detection, even if it saves time in resolving the collisions.

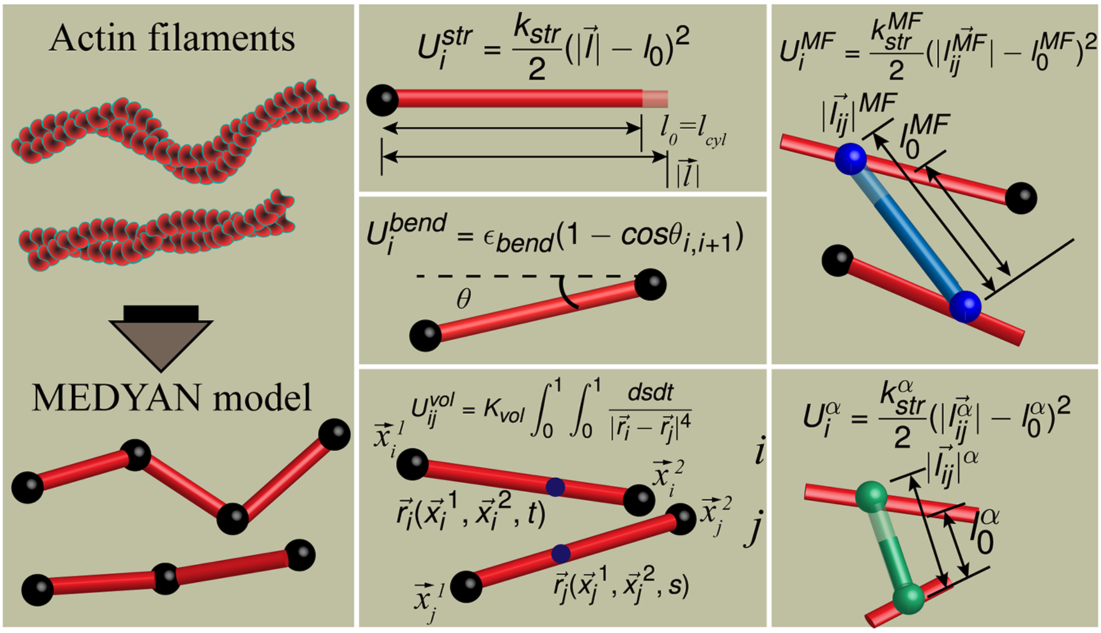

Actin filaments
MEDYAN model
$U_i^{str} = \frac{k_{str}}{2}(|\vec{l}| - l_0)^2$
$l_0 = l_{cyl}$
$|\vec{l}|$
$U_i^{bend} = \epsilon_{bend}(1 - cos\theta_{i,i+1})$
θ
$U_{ij}^{vol} = K_{vol}\int_0^1\int_0^1 \frac{dsdt}{|\vec{r}_i - \vec{r}_j|^4}$
$\vec{x}_i^{\,1}$
$\vec{x}_i^{\,2}$
$\vec{r}_i(\vec{x}_i^{\,1}, \vec{x}_i^{\,2}, t)$
i
$\vec{x}_j^{\,2}$
j
$\vec{x}_j^{\,1}$
$\vec{r}_j(\vec{x}_j^{\,1}, \vec{x}_j^{\,2}, s)$
$U_i^{MF} = \frac{k_{str}^{MF}}{2}(|\vec{l}_{ij}^{MF}| - l_0^{MF})^2$
$|\vec{l}_{ij}|^{MF}$
l_0^{MF}
$U_i^{\alpha} = \frac{k_{str}^{\alpha}}{2}(|\vec{l}_{ij}^{\alpha}| - l_0^{\alpha})^2$
$|\vec{l}_{ij}|^{\alpha}$
l_0^{α}

Figure 5.3: Elastic rods used to represent actin filaments in the MEDYAN cytoskeleton model [21]. The excluded volume potential U_{ij}^{vol} is shown at the bottom.

In the MEDYAN cytoskeleton model [21] (Figure 5.3), actin filaments are represented as cylinders, and collisions are handled by an analytical excluded volume potential given by the equation

$$U_{ij}^{vol} = K_{vol} \int_0^1 \int_0^1 \frac{dsdt}{|\boldsymbol{p}_i - \boldsymbol{p}_j|^4} \tag{5.2}$$

Where U is the energy resulting from the excluded volume, K_{vol} is a constant, and

$$\boldsymbol{p}_i^1 = \boldsymbol{r}_i^1 + t(\boldsymbol{r}_i^2 - \boldsymbol{r}_i^1) \tag{5.3}$$

Where $\boldsymbol{x}$ are the node positions, with the subscripts i and j denoting which filament the node belongs to, 1 and 2 denoting the index of the node within the rod, and t is a parameter denoting the length of the element.

This double integral could be avoided by using capsules instead of cylinders. If we enclose the rod elements by capsules with radius r, then any point on the surface of the capsule is always a distance r away from the rod element. This means that distances between capsules can be computed using a modified version of equation (5.1), the skew line formula

$$d = \left| \left(\frac{\boldsymbol{p}_a}{|\boldsymbol{p}_a|} \times \frac{\boldsymbol{p}_b}{|\boldsymbol{p}_b|} \right) \cdot (\boldsymbol{r}_b - \boldsymbol{r}_a - R_a - R_b) \right| \tag{5.4}$$

Where R_a and R_b are the radii of rod elements a and b.

This distance formula allows for the attractive part of the Lennard-Jones potential to be computed easily. As this is a distance between rods and not between nodes, the forces in question must be mapped back onto the rod nodes. The simplest solution is to divide the forces evenly between the two adjacent nodes, but this could introduce torque into the system. To keep the line of action for the forces the same, the following formula can be used to compute the points $\boldsymbol{c}_a$ and $\boldsymbol{c}_b$ that form the line segment joining the rod elements a and b,

$$\boldsymbol{c}_a = \boldsymbol{r}_a + \frac{(\boldsymbol{r}_b - \boldsymbol{r}_a) \cdot \boldsymbol{n}_b^p}{\frac{\boldsymbol{p}_a}{|\boldsymbol{p}_a|} \cdot \boldsymbol{n}_b^p} \frac{\boldsymbol{p}_a}{|\boldsymbol{p}_a|} \tag{5.5}$$

and

$$\boldsymbol{c}_b = \boldsymbol{r}_b + \frac{(\boldsymbol{r}_a - \boldsymbol{r}_b) \cdot \boldsymbol{n}_a^p}{\frac{\boldsymbol{p}_b}{|\boldsymbol{p}_b|} \cdot \boldsymbol{n}_a^p} \frac{\boldsymbol{p}_b}{|\boldsymbol{p}_b|} \tag{5.6}$$

where

$$n_j = \frac{\boldsymbol{p}_j}{|\boldsymbol{p}_j|} \times \left(\frac{\boldsymbol{p}_a}{|\boldsymbol{p}_a|} \times \frac{\boldsymbol{p}_b}{|\boldsymbol{p}_b|} \right). \tag{5.7}$$

The forces on the rod nodes can then be linearly interpolated according to the nearest points:

$$w_1 = \frac{|c_1 - r_1^a|}{|p_1|}, \; w_2 = \frac{|c_1 - r_2^a|}{|p_1|} \tag{5.8}$$

These skew line formulae actually find the closest point between two infinite lines. To apply this to finite rod segments, we check to see if the points $\boldsymbol{c}_a$ and $\boldsymbol{c}_b$ lie upon $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ [130]. If

$$\boldsymbol{p} \cdot (\boldsymbol{c} - \boldsymbol{r}_1) > 0 \tag{5.9}$$

and

$$\boldsymbol{p} \cdot (\boldsymbol{c} - \boldsymbol{r}_1) < \boldsymbol{p}^2 \tag{5.10}$$

then the point c lies on $\boldsymbol{p}$. If not, the closest node (either $\boldsymbol{r_1}$ or $\boldsymbol{r}_2$) can be used instead.

If the result of equation (5.4) is negative, its value is an ‘intersection radius’ that we can use to calculate the steric repulsion. The disadvantage of this method is that it would create a much harder potential for interactions at the hemispherical caps of the capsules — however, this is only a problem at the ends of rods, as within rods, the nth rod element cannot collide with the $n+1$th or $n-1$th elements — interactions between the those elements are already accounted or via bending stiffness of the rod. The direction and magnitude of the repulsive force could then be computed by evaluating $\frac{\partial r_i}{\partial x}$, the change in intersection radius with respect to node position, just as the energy due to internal rod forces is computed in section 2.2.4.

5.3 Rod-blob Interactions

Chapter 4 discuses the exchange of forces between rods and blobs made of FFEA tetrahedra by anchoring them to one another, but this description of rod-blob interactions does not include contact forces between the two. In FFEA, distance interactions between blobs and blobs (called surface-surface interactions) are handled by a summation over short-range interactions (mostly from van der Waals forces) of points that lie on those surfaces [1].

$$O_p^{LJ} = \sum_S \sum_T A_S A_T \sum_{k=1}^{N_G} \sum_{l=1}^{N_G} W_k W_l \psi_\alpha \left(s_k\right) f_i \left(s_k, t_l\right) \tag{5.11}$$

Where $\boldsymbol{s}$ and $\boldsymbol{t}$ are points on two surfaces, $f(\boldsymbol{s},\boldsymbol{t})$ is the force due to a point-point short-range interaction, $\boldsymbol{O}_p$ is the force vector formed from the external interaction, S and T are the total number of pairs of triangular surface faces, ψ are the surface shape functions, N_G is the number of points for quadrature, W_k and W_L are the weights for the sums in the Gaussian quadrature, and A_s and A_t are the areas of the surfaces.

For distance interactions between rods and blobs, the same method could be applied, but this time using a single a summation over short-range interactions of points that lie on the rod.

For steric repulsion, FFEA does not use the repulsive part of the Lennard-Jones potential, for the reasons discussed in section 5.2. Instead, it applies a force proportional to the volume intersection between two tetrahedra [131]. The volume overlap of two tetrahedra is calculable analytically [132], but this is not the case for tetrahedra with any rod primitive (e.g. cylinders, capsules). This computation could be substituted for an ‘intersection depth’ similar to equation (5.4).

5.4 Other Features

A more complete rod connection system would allow for rods to connect to other rods, to create networks of rods closer to polymers, and circular rods which can be used to represent cyclic proteins or DNA plasmids.

The stability of rod-blob interactions could be improved further by the addition of connections to multiple tetrahedra. The numerical stability of a rod-blob interaction is determined by the stability of the connected tetrahedron, and forces that are too large could cause that tetrahedron to invert, or cause the conjugate gradient solver for the equation of motion to fail to converge. Connections to multiple elements, while not necessarily physical, could increase the stability of the system, particularly for larger systems in which larger forces are transmitted across rod-blob connections.

5.5 Final Remarks

The end goal of this project is to build a model with enough features to represent the complete kinetochore/microtubule system, and other similar complexes. Currently, we have only looked at side-on attachment, and only used a subset of the proteins involved in this process. Neither the exact process of force transmission through DAM1 or the role of kinesin have been studied *in silico*. Meanwhile, the simulation of human kinetochores will prove to be far more difficult and computationally expensive than yeast.

After four years and 13,312 lines of code, Ndc80C and the kinetochore are still filled with secrets. The examination of these molecules at the mesoscale has raised many new and interesting questions but has failed to address many of those that already existed. In that sense, it is probably not correct to speculate on whether molecular biology will ever 'end'. Rather, biological processes can be understood at many different scales, at many different levels of detail, and through many different disciplines. A mesoscale representation of Ndc80C and the kinetochore is still very new, and experimental data at this scale is sparse. This project, while it has not 'solved' the kinetochore, has helped to bring our *in silico* understanding of it up to parity with experimental data.

Until the mid-00s, Moore's law ensured the continual shrinking of transistors and improvements to single-threaded CPU performance. Over the last 15 years, CPU and GPU parallelism have allowed for increasingly parallel computations. One day, we will also reach the limits of software parallelisation [133] [134], and will have to seek out new technologies and computing paradigms. Computational biophysics, however, should not let the speed and availability of computing power set the upper bound on the size and complexity of molecular dynamics systems. We cannot not depend on the exponential growth of computing speed to solve ever more difficult problems [31]. As the systems studied in molecular biology become larger and more complex, we must strive toward building and using more ambitious coarse-grained algorithms combined with top-down parametrisations, toward a truly multiscale, multidisciplinary vision of molecular biology, biophysics and biochemistry.

Appendix A

FFEA Method Validation

The KOBRA and FFEA algorithms were developed in tandem and are, in some ways, one and three-dimensional manifestations of the same idea. In 2018 the FFEA software publication [1] was released. This publication contained both a description and validation of the FFEA method, including a comparison of FFEA simulation trajectories with all-atom molecular dynamics.

Two molecules were selected for comparison, Arfaptin and Xylanase. The former is long and thin, the latter's cross section and length are approximately equal. These two molecules were converted into FFEA tetrahedral meshes. The resulting meshes are shown in Figure A.1.

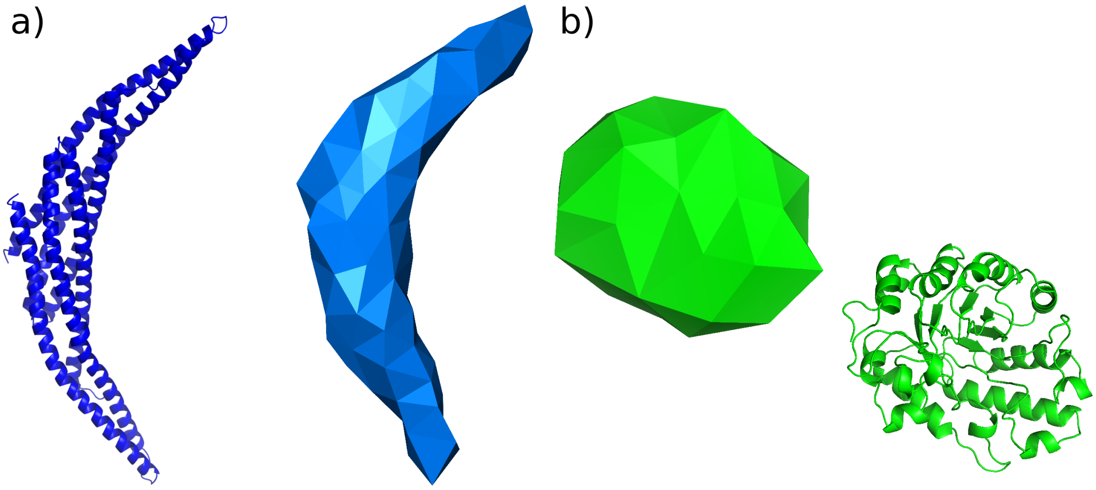


Figure A.1: All-atom cartoon vs FFEA tetrahedral mesh representations of test molecules Arfaptin (A) and Xylanase (B).

FFEA simulations of both were run with simulation parameters listed in table A.1.

Parameter	Arfaptin	Xylanase
E	$1GPa$	$1GPa$
Poisson Ratio	0.35	0.35
Shear/Bulk/solvent viscosities	$\mu = 1 \times 10^{-3} Pa \cdot s$	$\mu = 1 \times 10^{-3} Pa \cdot s$
Runtime	$100ns$	$760ns$

Table A.1: FFEA simulation parameters for validation simulations.

These were compared against simulation trajectories from the MoDEL database [135]. However, the MoDEL trajectories had differing starting positions and rotations to the FFEA simulations. In order to create comparable trajectories, the iterative closest point (ICP) algorithm [136] was used to align the starting positions of both trajectories.

In the ICP algorithm, the average distance between points in two point clouds.

$$E(R,t) = \frac{1}{N_p} \sum_{i=1}^{N_p} ||x_i - Rp_i - t||^2 \tag{A.1}$$

is minimised. Here, R and t are the translation vector and rotation matrix, N_p is the number of points, i is the point index, and X and P are the two point sets

$$X = \{x_1, ..., x_n\} \tag{A.2}$$

$$P = \{p_1, ..., p_n\} \tag{A.3}$$

The positions of the atoms and FFEA nodes were used for points. Once the translation vector and rotation matrix that aligned the FFEA blob onto the all-atom structure were found for the first frame, they were applied to every subsequent frame. The principal components of the two trajectories were then compared using the method described in section 3.4.2. The results of this comparison are shown in Figure A.2.

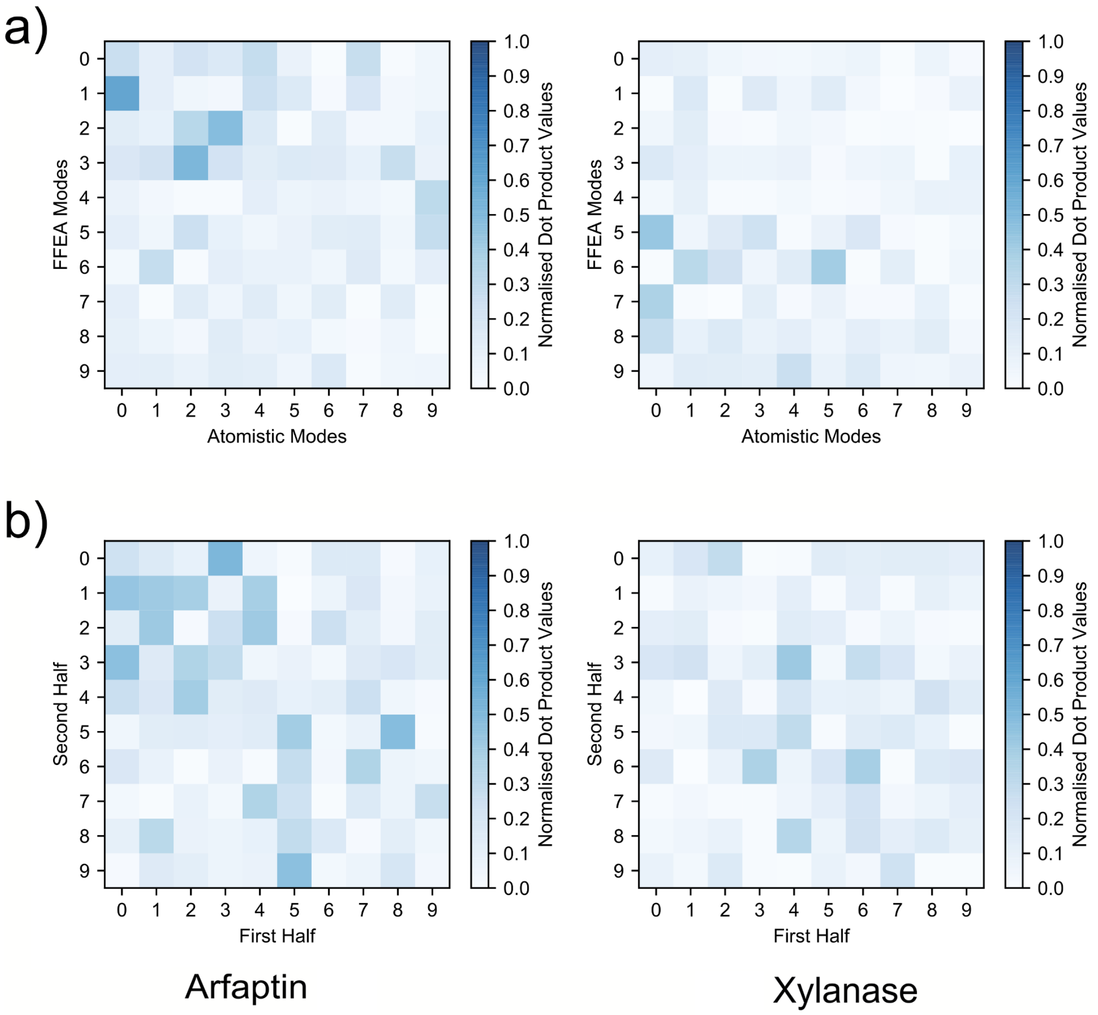


Figure A.2: Principal component dot product matrices for the FFEA and all-atom trajectories of Arfaptin and Xylanase. The top row is a comparison between all-atom and FFEA trajectories, the bottom row is a comparison between two halves of the all-atom trajectories. The modes are ordered from 0 to 9, with 0 being the largest. A perfectly-correlated matrix would be a blue diagonal line.

Neither molecule, for the all-atom simulation, shows a high degree of eigenspace overlap between the first and second halves of the trajectory, suggesting that conformational space is heavily undersampled. However, there is a degree of eigenspace overlap for Arfaptin (an average dot product of 0.6) which is comparable to that between the two halves of the all-atom trajectories. Xylanase performs less well (an average dot product of 0.4). This discrepancy is likely because the motion of Arfaptin is dominated by fluctations about its long axis, which is well-represented by FFEA method, whereas Xylanase is dominated by smaller-scale motion associated with its side chains and internal structure which FFEA does not represent. For an example of a more correlated set of principal components, see Figure 3.17.

Appendix B

Automated Tests

This section will offer brief descriptions of the tests included with the software package implementing the KOBRA algorithm.

Automated testing is a methodology that allows us to ensure that the results of scientific software are reliable, reproducible, and correct. This thesis offers a description of an algorithm which is also implemented in software. Using automated tests, we can verify that

1. The software is a correct and error-free implementation of the algorithm described in the text.
2. The algorithm produces the physics that we expect.
3. Future updates made to the software do not introduce errors into previously tested part of the software (regressions).

These tests come in two varieties: unit and integration. Unit tests check the output of the software at an atomic level - that a given mathematical function outputs the correct values, for example. Some of the unit tests in KOBRA verify the output of mathematical functions in C++ against values computed in Wolfram Mathematica or Python. Some are less specific, testing the signs, dot and cross products of vectors that should be pointing particular ways.

Integration tests, meanwhile, are 'end-to-end' tests that verify the output of the entire program. They are much broader in scope, and can test the physics that the algorithm is supposed to describe, as well as the correctness of the software. Integration tests can also be used to check emergent properties of the algorithm - features that have not been programmed in specifically, but which arise as a natural consequence of the design of the physics being simulated. The equipartition and polymer chain tests described in chapter 3 are an example of this.

These tests are built programmatically and are entirely automated. They are run nightly by an FFEA continuous integration test server running Jenkins. Every night, the server pulls the most recent version of the FFEA code from the development repository, runs all the tests, and creates a report. This process allows regressions (changes which cause a test to stop working) to be spotted quickly, and ensures that updates to the FFEA software do not cause any aspect of the software (or the physics) to stop working. The details of failing tests also allow the bug to be found and fixed more easily.

The names of the tests (in **bold**) correspond to the test names used internally by `ctest`, the test framework used by FFEA and KOBRA. To run these tests, compile the FFEA software, and then run `ctest` from the FFEA build directory.

B.1 Unit Tests

arbitrary equilibrium bend and **arbitrary equilibrium twist**: For an intrinsically bent and intrinsically twisted rod, check the energies (equations (2.9) and (2.21)) against known values for a single conformation. The test will fail if the values do not match.

bend test: for a set of exiting conformations, check that increasing the bend angle will increase the bend energy (equation (2.21)). If $E_{bend} < \widetilde{E}_{bend}$, the test will fail.

connection test: create a rod-blob interface to a single tetrahedron, at equilibrium, and apply an arbitrary rotation matrix $\boldsymbol{R}$ to that tetrahedron. Update the interface state, then apply the same rotation matrix to the interface element $\widetilde{\boldsymbol{p}}_a$. If $\boldsymbol{R} \cdot \widetilde{\boldsymbol{p}}_a \neq \boldsymbol{p}_a$, the test will fail. This tests the portion of the algorithm described in sections 4.1.3, 4.1.2 and 4.1.1.

connection energy: create a rod-blob interface at a single tetrahedron. Compute the energy at the interface. If $\widetilde{E}_a \neq 0$ - the energy about the interface node is non-zero at equilibrium - the test will fail. This tests the portion of the algorithm described in section 4.1.5.

connection identify face: create a rod-blob interface on a sphere. Find the dot product of the attachment element with the face edges. If $\boldsymbol{p}_a \cdot \Delta \boldsymbol{x}_{face} \neq 0$, the test will fail. When a connection is initialised, the face nodes given are absolute, but the rod-blob interface requires the node ids to be relative to the tetrahedron, so this test will also fail if the face node indices cannot be identified.

connection orientation: this test will fail if the rod-blob connection crashes on initialisation.

parallel transport test: for an intrinsically straight rod at equilibrium, compute $\widetilde{E}_{twist}$, the twist energy (equation (2.9)) at equilibrium. Then, move one node, and recompute the orientations of the rod elements and material axes (section 2.2.5). Then, recompute the twist energy. The test will fail if the movement of a node introduces a twist energy, e.g. if $E_{twist} > 0$.

recover normal: for a rod-blob interface at equilibrium with no rotation applied to the attachment element $\boldsymbol{p_a}$, compute the orientation of the attachment element from the equilibrium Jacobian of the tetrahedron (see equation (4.1.3)). Then, compute the normal to the surface face of the tetrahedron, $(\boldsymbol{x}_1 - \boldsymbol{x_3}) \times (\boldsymbol{x}_2 - \boldsymbol{x_3})$. The test will fail if they are not equal.

rodrigues test: for a straight rod, perform a rodrigues rotation (equation (2.32)) of the material axis by π radians. If $\boldsymbol{m}' \neq -\boldsymbol{m}$, the test will fail.

test bend against mathematica and **test mutual bend against mathematica**: for an arbitrary rod conformation, compute the bend energy E_{bend} about one node. Compare it to a value computed by evaluating the bending energy formula in Wolfram Mathematica. If they are not equal, the test will fail. There are two versions of this test, using the normal (equation (2.15)) and mutual element (equation (2.21)) bending energy formulae.

test bend for non-straight equilibrium: the same as the above test, but for a rod with a non-straight equilibrium structure.

parallel transport test: for a rod with an arbitrary (and bent) conformation, parallel transport (equation (4.9)) a material axis to its neighbouring element. If $P(\boldsymbol{m}_1, \boldsymbol{l}_1, \boldsymbol{l}_2) \cdot \boldsymbol{m}_2 \neq 0$, the test will fail.

translation and rotation test: initialise a rod and rotate the rod using the built-in rod rotation methods. Then, apply the same rotation to a single element from the rod in isolation. If the two elements are not equal, the test will fail. This tests the entire rod translation and rotation functions used during initialisation.

twist bend independence: for a straight rod conformation, twist the material axes (equation (2.32)) and compute the twist energy (equation (2.9)). Then compute the bend energy (equation (2.21)). The test will fail if the bend energy $E_{bend} \neq 0$, or if the twist energy $E_{twist} = 0$.

twist bend independence 2: this test is the same as **parallel transport test**, but now, it computes the energies of the 5-node block used in the numerical integration, rather than on a single node.

twisted stretch test: for a rod with a straight equilibrium, but a non-zero stretch energy, apply a Rodrigues rotation (equation (2.32)) to one of the material axes. If applying the twist has increased the stretch energy (equation (2.6)) of the rod, the test will fail. This doesn't just test the stretch energy, but also the normalisation of the elements during energy calculations.

B.2 Integration Tests

The tests for polymer chain physics, equipartition and formation of hockles (section 2.4) are all examples of integration tests, and the equipartition test is present as an automated test, named **equipartition test**.

B.2.1 Symmetry

If we compute the energy and dynamics of a structure which is symmetrical about one axis, those energies and dynamics should also be symmetrical about that axis. In early versions of KOBRA, a forward difference was used to compute the forces, instead of the central difference (equation (2.2.4)), and this test was developed to compare the two integration schemes.

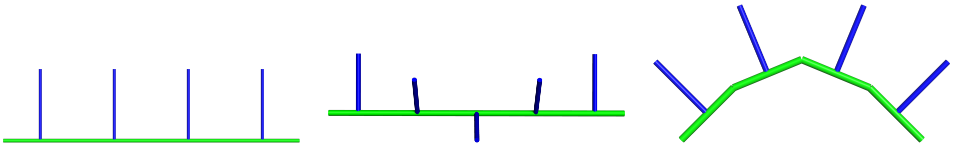


Figure B.1: Starting structures for symmetry tests in each degree of freedom (stretch, twist and bend).

The equilibrium structure for each trajectory is straight and untwisted. In each symmetry test, the structure gradually relaxes back to equilibrium at $T = 0$. Partway through this relaxation (after 10^{-11} seconds) the energies and dynamics are then tested for symmetry.

The following tests are performed on the energies and dynamics of nodes 2 and 4. Each test will fail if the equality in the equation is not true,

$$E_2 = E_4 \tag{B.1}$$

$$\Delta r_{2,x} = -\Delta r_{4,x} \tag{B.2}$$

where $\Delta r_{i,x}$ is only the x-component of Δr, which in this case is along the rod. Additionally, for node 3 (the central node), in the bending trajectory, the following test is performed.

$$\Delta \boldsymbol{r}_3 \cdot [0, -1, 0] = 1 \tag{B.3}$$

For the stretching and twisting trajectories, the following test is performed instead,

$$\Delta \boldsymbol{r}_3 = 0 \tag{B.4}$$

Finally, in the stretching trajectory, the following test is performed,

$$\boldsymbol{l}_i \cdot [1,0,0] = 1 \tag{B.5}$$

This test can be addressed within the FFEA software package under the name **symmetry test**.

B.2.2 Connection propagation

The only real integration test that can be performed on a rod-blob connection is checking to see if translations and rotations are preserved across the connection. These tests are performed for each degree of freedom, and in each direction (blob-to-rod and rod-to-blob) for a total of 6 tests.

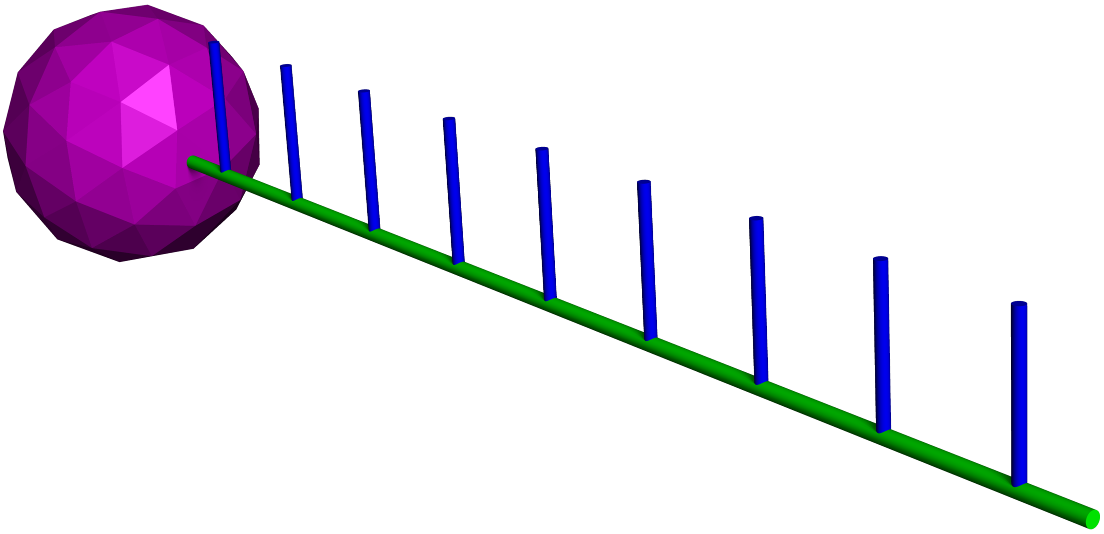


Figure B.2: System used in the connection propagation test — a 10-element rod, 90nm in length, connected to a tetrahedral sphere.

Before each test, the dot product between some arbitrary edge vector in the sphere and the element at the far end of the rod is computed,

$$a = \boldsymbol{p}_{end} \cdot \boldsymbol{y} \tag{B.6}$$

Each simulation trajectory is created at $T = 0$. Over the course of $50ns$, one end of the connection (the rod or the blob) is rotated by $\frac{1}{2}\pi$ radians. Then the system is equilibrated for $850ns$. At the end of the simulation, the dot product is computed again. If $\boldsymbol{a} \neq \boldsymbol{a}'$, the test fails.

This test can be addressed within the FFEA software package under the name **connection propagation test**.

Appendix C

KOBRA Code Style

The KOBRA algorithm is implemented in a combination of C++ and Python, and can be found online via bitbucket at ffea.bitbucket.io. The three files implementing KOBRA are `rod_math_v9.cpp`, `rod_structure.cpp` and `rod_blob_interface.cpp`. `rod_math` contains all of the pure math functions, implementing the rod energies and dynamics, while `rod_structure` implements data structures, I/O and initialisation. `rod_blob_interface` contains both the mathematical formulae and the data structures responsible for the rod-blob interface, the former in the first half of the file and the latter in the second half.

The style conventions used in these files are as follows:

- The entire contents of each file are declared within the namespace '`::rod`'.
- Maths functions with single return values simply return values. Functions that return multiple values or arrays have their return values initialised outside the scope of the function and passed as parameters. In the function definition, an empty macro (`OUT`) signifies where the parameters end and the return values begin.
- Rod methods return `*this` to allow for method chaining.
- Human readable indexes are defined in `rod_math_v9.h`. `x`, `y` and `z` correspond to the 0th, 1st and 2nd elements of an array, and any time they are used, they refer to dimensions. `im2, im1, i` and `ip1` refer to the indices of segments. Here, they are given relative to the node at the beginning of the ith segment.
- Only single-precision floating point numbers are used where possible to allow for easy GPU parallelisation in the future (and better performance).[1]
- The `vec3d` macro is used instead of a for loop for adding, subtracting, multiplying and dividing 3-element arrays, which is extremely common throughout the codebase.
- Memory for the rods is allocated on the heap at initialisation. Thereafter, heap allocation is not used, all `rod_math` work variables are allocated on the stack.

[1] There is one exception to this rule, a precise normalisation is used for the material axes in the twist energy. This is due to a quirk of `std::acos` that occurs using FFEA's default compiler settings. If the program is compiled without `-ffast-math`, this can be safely replaced with a regular normalize.

- If the static const `debug_nan` is defined to be true, the program will dump intermediate values to the console during runtime.

- The function `not_simulation_destroying` is used to check if values are equal to `NaN` or `inf`. The program will be forcibly aborted if any values are `NaN` or `inf`. FFEA's compiler options do not permit testing for `NaN` or `inf`, so `boost::math` is used instead. FFEA's compiler options also do not permit aborting with `std::abort` so less scrupulous methods are used.

- Data structures are kept as flat as possible. This reduction in indirection is an intentional choice made for cache and memory optimisation.

Appendix D

Roadmap for the KOBRA Code

Chapter 5 had some aspects of a summary and a conclusion, but it was mostly comprised of a discussion on future KOBRA features and how they might be implemented. This section complements that chapter, but this time looking at KOBRA from a software engineering perspective. The proposals in this section are not suggestions for enhancements or new features, instead they are the minimum that will have to be done to stave off code rot and keep FFEA and KOBRA usable in the future. If you are working on KOBRA (the code, not just the mathematics of it), then you might find this interesting. If not, it can be safely ignored, like a dead body at an otherwise picturesque beach.

D.1 Python 2 to 3

In 2008, Python creator Guido van Rossum created a new programming language, similar to his previous project, Python. This new language was, confusingly, called Python, and was slightly better, but different enough that everybody would have to rewrite their programs.

FFEAtools doesn't run in Python 3. If Python 2 is first on the system path, it can still run from the terminal. Even if distros stop providing Python 2, Anaconda have pledged to support it for years to come. But a lot of applications are now treating Python 3 as the default, and it's a fight to get them to even use Python 2. In particular, cmake has this behaviour, which means that FFEA's unit tests fail on new distros, as cmake prioritises Python 3 even if Python 2 is first in the path. At some point, ffeatools needs to be ported over to this new language. Again, I would recommend a progressive approach: import from future, slowly change over print statements, checking that I/O still works in Python 3 — I suspect that unicode strings will create a few regressions. Ranges and integer division will probably break some functionality as well. All of the KOBRA code should run in Python 3 — in fact, there's a few small pieces of it that only run in Python 3, because they depend on ISAMBARD.

Of all the housekeeping discussed in this section, migrating to Python 3 is probably the most important. Not being able to run ffeatools or the tests will stop potential users from even trying the software out.

D.2 Visualisation

FFEA writes trajectories to text files — .traj and .rodtraj are their file extensions — which are then read and interpreted by a plugin for PyMOL, a molecular dynamics visualiser written in Python. This isn't optimal for a few reasons. First, there are standard file formats for most of the types of data that FFEA uses, but FFEA rolls its own formats for every single type of data, which makes it inoperable with lots of other tools[1]. Second of all, PyMOL isn't a good fit for either FFEA or molecular dynamics. It's very slow, has a limited feature set compared to VMD, and FFEA piggybacks onto PyMOL features that are buggy, incomplete, and definitely weren't designed to be used to load volumetric meshes.

For example, the way FFEA sends display lists of triangles to be drawn by PyMOL is as follows: it reads in the data from a very large text file into a rabbit warren of objects containing lists of objects containing lists of objects containing lists of NumPy arrays, it serializes that data into PyMOL's custom CGO format, then it sends the CGO display list to PyMOL via PyMOL's CGO API. This is impressively slow and can use many gigabytes of memory to load a display list that would optimally be comprised of no more than a few hundred megabytes. It's also liable to segfault PyMOL and run out of memory while loading longer trajectories. PyMOL's internal memory management is such that it normally crashes or runs out of memory when loading more than 100 objects. FFEA simulations routinely contain more than 100 objects. Even worse, once those triangles (or cylinders, in the case of rods) are loaded into PyMOL, there's no API to interact with them. This is why the FFEA viewer has an option to load 'Supportive Fake Atoms' onto objects - those atoms, positioned on nodes or faces, can then be interacted with via PyMOL's API or user interface. This is the only way to interact with FFEA objects in PyMOL, even for trivial functions such as taking measurements.

Another issue with FFEA's current approach to visualisation is that all the initialisation code has to be duplicated. Even though the FFEA codebase is split down the middle between Python and C++, there is no interface between the two. If FFEA

[1] With the exception of rodtraj, which is just a CSV file, a lot of these formats are also incredibly hard to parse.

was written in C, this would be easily achievable with ctypes, but C++ would require quite a lot of glue code. This isn't necessarily a bad thing, as the C++ and Python have very different purposes and there's little duplication of functionality outside of basic I/O. For rods, having some duplication helped, as it allows for tests to compare python versions of calculations to their C++ counterparts. However, this all falls apart in initialisation. FFEA systems are stored as a collection of structure and parameter files for individual objects, all referenced in a single .ffea script file, which describes how these objects are positioned inside the simulation box. It's a very elegant design, as users can spawn multiple objects from a single structure file, and the software will write separate trajectories for each instance. The problem is that the way FFEA positions objects inside the simulation box is very complex and has a lot of quirks. I will list some quirks now, for future reference.

- The center of the box isn't at 0,0,0. It's at $\frac{1}{2}w$, $\frac{1}{2}h$, $\frac{1}{2}d$. Objects entering the simulation box are positioned relative to this point, not 0,0,0.
- The dimensions of the simulation box are not stated in the .ffea file but are instead determined by the vdw cutoff distance multiplied by the number of cells for the nearest neighbour lookup table.
- The units for translations of objects are not SI units OR mesounits, they are scaled with the scale factor for that object.

All of this happens the first time you run an FFEA simulation (in the FFEA executable, written in C++) and the resulting positions of all the nodes are then written to the trajectory files. This means that, if you're preparing a simulation and want to see what it looks like, the visualiser needs to be able to replicate all of this incredibly quirky behaviour. KOBRA doesn't do this, and not for lack of trying, but because that list of quirks is far from complete. It's like trying to float a `<div>` in Internet Explorer 6, there is simply no telling where an object will end up. Instead, I would recommend writing a dummy frame to the trajectory and then loading in the system with the trajectory.

The main reason PyMOL was chosen was to allow FFEA to be loaded side-by-side with PDB files or volumetric data. At the time, there weren't many alternatives that were extensible and had this functionality. I would not recommend trying to 'fix' or 'improve' the FFEA visualisation experience beyond a certain point, because you're always going to brush up against hard-coded limits in PyMOL. In a previous iteration of this appendix, I had recommended rewriting FFEA to output Paraview-compatible files, however, this approach also has some drawbacks. Paraview doesn't have the capacity to parse or render every type of FFEA data, which would neccessitate the development of a FFEA ParaView extension - far more work than the FFEA PyMOL viewer. ParaView's ability to visualise molecular dyanmics data is extremely limited, to the extent that it's not really usable to compare FFEA and MD systems. Finally, ParaView is not easy to learn, and it may present too steep a learning curve for many in the molecular dynamics community.

D.3 Selecting Optimal Parallelisation

KOBRA has a single parallelisation mode — within-rod parallelisation, equivalent to FFEA's within-blob parallelisation. This means that, for a system with multiple rods, the dynamics for each rod will be computed serially, and the dynamics of each node and material axis are computed in parallel. This works fine for the systems described in this thesis, but if the number of rods you have approaches the number of CPU cores on the system, it might be better to parallelise between rods instead.

Currently, KOBRA always runs at the same parallelisation level as FFEA. This is not always optimal. For example, say you have a rod of 14 nodes, and you wish to run FFEA on 8 threads. KOBRA will run on 8 threads even though it would've run faster on 7. It would be a worthwhile improvement to manually calculate the optimal number of CPU threads to run KOBRA on and use this number if it is faster than the total number of FFEA CPU threads.

Another note on parallelisation: KOBRA would parallelise very well on GPUs. I would recommend either the newer versions of OpenMP or OpenACC. The big problem would be getting FFEA to run on them, not KOBRA. FFEA's parallelisation scheme depends on a labyrinthine series of preprocessor macros, which have accumulated over time and are in desperate need of refactoring.

D.4 FFEA Collision Detection Data Structures

FFEA handles collisions by subdividing space into cuboids and checking for collisions between each cuboid and its 27 nearest neighbours. This approach is best for systems of uniform density, and while FFEA systems right now are on the sparse side, they're likely to get denser in the future. Unfortunately FFEA's implementation of this algorithm is messy, cache-thrashing, contains a huge amount of duplicated code, and even though it's heavily templated, is also deeply coupled to the data structures of FFEA blobs. There's no realistic way to drop KOBRA rods in there and make it work. Adding a separate nearest neighbour lookup table but for rods might seem like unnecessary duplication, but a highly abstract collision detection system that's completely data-structure agonistic, would be a huge amount of work, and would introduce a huge amount of complexity to a very performance-sensitive part of the code.

The way FFEA solves collisions is FFEA-specific, but the way it detects for collisions is not. For each face, all that needs to be known is 'what are the indices of other faces in collision range'? That is a solved problem, and I would recommend using an existing solution (of which you'll find many on github) rather than creating a new one.

D.5 Structure Alignment Tools

FFEAtools contains a tool called node_pdb_align, which was written with the sole purpose of aligning single FFEA objects with PDB files representing single molecules. This tool will become less relevant in FFEA's future as it doesn't align rods, nor does it align entire scripts, only individual blobs. For aligning rods to blobs, some extremely basic functionality is provided in the form of `rod_creator.get_euler_angles_from_pdb`. However, as the systems FFEA is used for get larger, correctly positioning objects will only become more labour-intensive. node_pdb_align uses the iterative closest point method, which is not ideal for individual objects, but would work better with large systems of connected rods and blobs, although it is liable to get stuck in local minima very easily. The importance of new structure alignment tools depends a lot on how easy it is to implement graphical tools to position, rotate and connect objects - all of which is currently done in scripts. Aligning objects with different topologies in 3-d space is not a solved problem, and the human brain is arguably better at it than any hastily-written script. If possible, then, it should be left to brains.

D.6 Structure of ffeatools and Python Scripts

FFEAtools was originally created by some very talented programmers who had the patience for C++ but not the time. Therefore, FFEAtools, the FFEA initialisation and analysis library, was written in Python out of necessity and not out of any particular love for Python or desire to use its idioms. Traditionally, Python programs are packaged with distutils into self-contained distributions which are installed with setuptools. These distributions are installed into Python's site-packages folder, dependencies are resolved, and the Python package can run special code during installation. FFEAtools did not initially do this. Instead, it dumped an array of Python files into a folder in `/opt` and then added this folder to the system path. This approach has a number of disadvantages, but chief among them is that the Python scripts can't be structured hierarchically like most python projects are, because individual Python scripts running out of some arbitrary directory can't do relative module imports like Python modules can.

The only way to fix this is to restructure FFEAtools to be a normal Python distribution, and here a start has been made. You can indeed install FFEAtools using setup.py, but it's still a work in progress. There are some FFEAtools which are actually C binaries that are (poorly) wrapped. Many FFEAtools still use the script imports (`import FFEA_thing`) instead of the module imports (`from FFEAtools.modules import FFEA_thing`). There are still a few FFEAtools that don't have code encapsulated into functions, run by looking at `sys.argv` and end by `sys.exit`. If you're working on the various tools in FFEAtools, please add support for the python module installation as you go, and add deprecation warnings where appropriate. There is some KOBRA functionality that requires the python module structure, because FFEA_rod.py was getting enormous and needed to be broken into sub-modules. The only way to do that without polluting the system and python paths was to create a sub-folder and use a relative import. In the future I would envisage the rest of the ffeatools, which currently occupy a separate folder, be moved into subdirectories of the modules folder, so that modules can import from them and reduce duplication. I would also recommend separating out the rod loader, rod analysis, rod math and rod creator python tools into their own files.

D.7 Periodic Boundary Conditions

KOBRA don't support any boundary conditions, even hard wall boundary conditions. This could be problematic for very large systems, particularly because it runs in singe-precision. The extent of KOBRA's stability with respect to systems a long way from the origin has not been tested. FFEA does not handle periodic boundaries within objects, only between them, which could create problems in the long axis of rods, and also for structures made from connected rods and blobs. Therefore, implementing periodic boundaries for rods is an extremely dangerous manoeuvrer and should not be attempted, excepted by highly skilled and highly-paid research software engineers.

D.8 Caching of Results

One of the most obvious performance improvements in KOBRA would be the caching of equilibrium values such as $\Delta\boldsymbol{\omega}$ and $\Delta\theta$. This could theoretically halve the amount of time it takes to calculate the energies. However, implementing this introduces a lot of state into the ffea_rod object that previously wasn't present. Updating the material parameters or equilibrium state of the rod would also necessitate an update to these cached values, and would therefore require the addition of getters, setters and private data structures.

With cached values, an almost equally large but far easier performance gain would be to replace the symmetric Euler method with an asymmetric one. This was a change made while trying to improve the numerical stability of the algorithm, but ultimately it didn't do much to improve it, and ate up a lot of CPU cycles. It could be argued that any optimisation of KOBRA is premature optimisation, because it's much faster than FFEA. Optimisation might become a more pertinent problem if someone wants to implement KOBRA into their code.

Appendix E

Simulation Parameters

Constant	Value
Element radius a	$5nm$
Dynamic viscosity μ	$0.6913MPa \cdot s$ [137]
Bending constant B	$3 \cdot 10^{-25}\ m^4 \cdot Pa$
Twisting constant β	$1.43 \cdot 10^{-26}\ Nm^2$
Stretching constant κ	$1 \cdot 10^{11}\ N$
Temperature	$300K$

Table E.1: Values of constants for the equipartition simulation in section 2.4.1 and the polymer chain tests in section 2.4.2. All material constants are homogeneous, and the bending stiffness is isotropic, with B being the diagonal elements of the bending stiffness matrix $\boldsymbol{B}$.

Constant	Value
Element radius a	$5nm$
Dynamic viscosity μ	$0.6913MPa \cdot s$ [137]
Bending constant B	$2 \cdot 10^{-27}\ m^4 \cdot Pa$
Twisting constant β	$3 \cdot 10^{-26}\ Nm^2$
Stretching constant κ	$3 \cdot 10^{11}\ N$
Temperature	$0K$

Table E.2: Values of constants for buckling test in section 2.4.3. All material constants are homogeneous, and the bending stiffness is isotropic, with B being the diagonal elements of the bending stiffness matrix $\boldsymbol{B}$.

	Minimising	Heating	Relaxing (1/2)	Relaxing (2/2)	MD
ntf		2	2	2	2
nt		2	2	2	2
ntb	0	0	0	0	0
cut	30	20	20	20	20
ntp	0				
nstlim		40000	200000	200000	
dt		0.00025	0.0005	0.001	0.002
tempi		0			
temp0		300	300	300	300
ntt		1	1	1	1
restraint_wt		150	70	10	
imin	1	0	0	0	0
ntx		1	5	5	5
maxcyc	20000				
ncyc	2500				
igb	1	1	1	1	1
gbsa		1	1	1	1
ntr	0	1	1	1	0
saltcon	0.1	0.1	0.1	0.1	0.1
restraintmask		@CA,C,O,N	@CA,C,O,N	@CA,C,O,N	

Table E.3: AMBER parameters for molecular dynamics simulation of Ndc80C used in Chapter 3.

Simulation	Section	N	Timestep	Check	Frames	Radius	β (Nm^2)	κ (N)	B $(m^4 \cdot Pa)$
$\boldsymbol{B}$ recovery	3.2.3	209	5×10^{-12}	1000	2000	5×10^{9}	4.7×10^{-29}	3.4×10^{-11}	$\sim$
Iterative $\boldsymbol{B}$	3.2.3	53	5×10^{-12}	1000	4651	5×10^{9}	4.7×10^{-29}	3.4×10^{-11}	$\sim$
Twist recovery	3.2.3	10	5×10^{-11}	20	50000	5×10^{9}	$\sim$	1×10^{-11}	1×10^{-27}
Stretch recovery	3.2.3	10	5×10^{-11}	20	50000	5×10^{9}	5×10^{-27}	$\sim$	1×10^{-27}
Ndc80C source	3.3	14	2×10^{-15}	500	50476	n/a	n/a	n/a	n/a
Ndc80C iter0	3.3	14	1×10^{-12}	3000	100000	5×10^{9}	$\sim$	$\sim$	$\sim$
Ndc80C iter1	3.3	14	1×10^{-12}	3000	100000	5×10^{9}	$\sim$	$\sim$	$\sim$

Table E.4: KOBRA/FFEA parameters for FFEA simulations used in chapter 3. For all simulations, the thermal energy kT was 4.11×10^{-21}. For more information on these simulations, raw data can be found at `doi.org/10.1039/D0SM00491J` and `bitbucket.org/Robert-Welch/kobra-raw-data`.

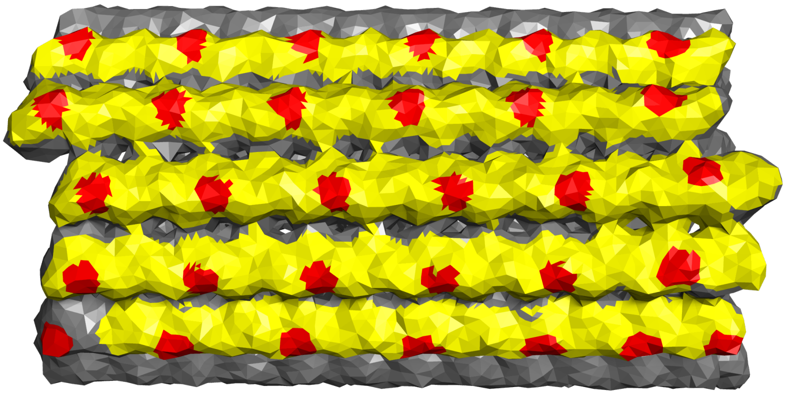


Figure E.1: A closer look at the microtubule track used in the simulations in chapter 4. Binding sites (red) have an affinity to Ndc80C's globular domains.

Constant	Value
Temperature	$300K$
Timestep	$3 \times 10^{-13}s$
Check	2000
Stokes viscosity	$1 \times 10^{-3} Pa \cdot s$
Van der Waals cutoff distance	$3.5 \times 10^{-8}m$
Spring constants	$1.771 \times 10^{-4},\ 7.143 \times 10^{-5},\ 7.355 \times 10^{-5}\ N/m$
Spring lengths	$5.645 \times 10^{-9},\ 1.4 \times 10^{-8},\ 1.257 \times 10^{-8}\ m$
Stokes radius (FFEA)	$1.1295 \times 10^{7}\ m$
Mesh density (FFEA)	$1 \times 1.5^{3}\ kg/m^{3}$
Internal bulk viscosity	$1.5 \times 10^{-3}\ Pa \cdot s$
Internal shear viscosity	$1.5 \times 10^{-3}\ Pa \cdot s$
Internal shear modulus	$5.5 \times 10^{8}\ Pa$
Internal bulk modulus	$2.2 \times 10^{9}\ Pa$

Table E.5: Values of constants for the FFEA simulations performed in chapter 4. This is not an exhaustive list, and rod parameters are not included (see section 3.3 for those).

Appendix F

Walrus

Figure F.1: FFEA walrus, drawn by A. Ghesquiere. Suggested names: Tet, FluFEA, Meso-unit

Bibliography

[1] Albert Solernou, Benjamin S. Hanson, Robin A. Richardson, Robert Welch, Daniel J. Read, Oliver G. Harlen, and Sarah A. Harris. Fluctuating Finite Element Analysis (FFEA): A continuum mechanics software tool for mesoscale simulation of biomolecules. *PLOS Computational Biology*, 14(3):e1005897, March 2018. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005897. URL http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005897.

[2] Robert Welch, Sarah A. Harris, Oliver G. Harlen, and Daniel J. Read. KOBRA: a fluctuating elastic rod model for slender biological macromolecules. *Soft Matter*, 16(32):7544–7555, August 2020. ISSN 1744-6848. doi: 10.1039/D0SM00491J. URL https://pubs.rsc.org/en/content/articlelanding/2020/sm/d0sm00491j.

[3] Sandrine Ruchaud, Mar Carmena, and William C. Earnshaw. Chromosomal passengers: conducting cell division. *Nature Reviews Molecular Cell Biology*, 8(10):798–812, October 2007. ISSN 1471-0080. doi: 10.1038/nrm2257. URL https://www.nature.com/articles/nrm2257.

[4] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips. A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. *Nature*, 181(4610):662–666, March 1958. ISSN 1476-4687. doi: 10.1038/181662a0. URL https://www.nature.com/articles/181662a0.

[5] Yoana N. Dimitrova, Simon Jenni, Roberto Valverde, Yadana Khin, and Stephen C. Harrison. Structure of the MIND Complex Defines a Regulatory Focus for Yeast Kinetochore Assembly. *Cell*, 167(4):1014–1027.e12. ISSN 0092-8674. doi: 10.1016/j.cell.2016.10.011. URL http://dx.doi.org/10.1016/j.cell.2016.10.011.

[6] Shane Gonen, Bungo Akiyoshi, Matthew G Iadanza, Dan Shi, Nicole Duggan, Sue Biggins, and Tamir Gonen. The structure of purified kinetochores reveals multiple microtubule attachment sites. *Nature structural & molecular biology*, 19 (9):925–929, August 2012. ISSN 1545-9985. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3443262/.

[7] Maria Kalantzaki, Etsushi Kitamura, Tongli Zhang, Akihisa Mino, Béla Novák, and Tomoyuki U. Tanaka. Kinetochore-microtubule error correction is driven by differentially regulated interaction modes. *Nature Cell Biology*, 17(4):421–433, April 2015. ISSN 1476-4679. doi: 10.1038/ncb3128.

[8] Stefan Westermann, Hong-Wei Wang, Agustin Avila-Sakar, David G. Drubin, Eva Nogales, and Georjana Barnes. The Dam1 kinetochore ring complex moves processively on depolymerizing microtubule ends. *Nature*, 440(7083):565–569, March 2006. ISSN 1476-4687. doi: 10.1038/nature04409.

[9] Zsolt Bertalan, Caterina A. M. La Porta, Helder Maiato, and Stefano Zapperi. Conformational Mechanism for the Stability of Microtubule-Kinetochore Attachments. *Biophysical Journal*, 107(2):289–300, July 2014. ISSN 0006-3495. doi: 10.1016/j.bpj.2014.06.004. URL https://www.cell.com/biophysj/abstract/S0006-3495(14)00606-7.

[10] Emily Anne Scarborough, Trisha N Davis, and Charles L Asbury. Tight bending of the Ndc80 complex provides intrinsic regulation of its binding to microtubules. *eLife*, 8:e44489, May 2019. ISSN 2050-084X. doi: 10.7554/eLife.44489. URL https://doi.org/10.7554/eLife.44489.

[11] Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-Grained Protein Models and Their Applications, June 2016. URL https://pubs.acs.org/doi/full/10.1021/acs.chemrev.6b00163.

[12] EMStats: EMDB statistics, . URL https://www.ebi.ac.uk/pdbe/emdb/statistics_main.html/.

[13] Sébastien Neukirch, Alain Goriely, and Andrew C. Hausrath. Chirality of coiled coils: elasticity matters. *Physical Review Letters*, 100(3):038105, January 2008. ISSN 0031-9007. doi: 10.1103/PhysRevLett.100.038105.

[14] Byoungwon Choe, Min Gyu Choi, and Hyeong-Seok Ko. Simulating Complex Hair with Robust Collision Handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, pages 153–160, New York, NY, USA, 2005. ACM. ISBN 978-1-59593-198-6. doi: 10.1145/1073368.1073389. URL http://doi.acm.org/10.1145/1073368.1073389.

[15] A. Theetten, L. Grisoni, C. Andriot, and B. Barsky. Geometrically exact dynamic splines. *Computer-Aided Design*, 40(1):35–48, January 2008. ISSN 0010-4485. doi: 10.1016/j.cad.2007.05.008. URL http://www.sciencedirect.com/science/article/pii/S0010448507001091.

[16] Schrödinger, LLC. The PyMOL Molecular Graphics System, Version 1.8. November 2015. URL https://pymol.org.

[17] Claudio Ciferri, Sebastiano Pasqualato, Emanuela Screpanti, Gianluca Varetti, Stefano Santaguida, Gabriel Dos Reis, Alessio Maiolica, Jessica Polka, Jennifer G De Luca, Peter De Wulf, and others. Implications for kinetochore-microtubule attachment from the structure of an engineered Ndc80 complex. *Cell*, 133(3): 427–439, 2008. doi: 10.1016/j.cell.2008.03.020. URL http://www.sciencedirect.com/science/article/pii/S0092867408004017.

[18] Hong-Wei Wang, Sydney Long, Claudio Ciferri, Stefan Westermann, David Drubin, Georjana Barnes, and Eva Nogales. Architecture and flexibility of the yeast Ndc80 kinetochore complex. *Journal of Molecular Biology*, 383(4):894–903, November 2008. ISSN 1089-8638. doi: 10.1016/j.jmb.2008.08.077.

[19] Benjamin S Hanson. *Mesoscale Modelling of Cytoplasmic Dynein using Fluctuating Finite Element Analysis.* PhD Thesis, University of Leeds, 2018. URL http://etheses.whiterose.ac.uk/19398/.

[20] Stefan Gruber. SMC Complex PDB structure (private commuication).

[21] Aravind Chandrasekaran, Arpita Upadhyaya, and Garegin A. Papoian. Remarkable structural transformations of actin bundles are driven by their initial polarity, motor activity, crosslinking, and filament treadmilling. *PLOS Computational Biology*, 15(7):e1007156, July 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007156. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007156.

[22] Aussie Suzuki, Benjamin L. Badger, Julian Haase, Tomoo Ohashi, Harold P. Erickson, Edward D. Salmon, and Kerry Bloom. How the kinetochore couples microtubule force and centromere stretch to move chromosomes. *Nat Cell Biol*, 18(4):382–392, April 2016. ISSN 1465-7392. URL http://dx.doi.org/10.1038/ncb3323.

[23] Franz Schrader. The kinetochore or spindle fibre locus in amphiuma tridactylum. *The Biological Bulletin*, 70(3):484–498, June 1936. ISSN 0006-3185. doi: 10.2307/1537304. URL https://www.journals.uchicago.edu/doi/abs/10.2307/1537304.

[24] Franz Schrader. Kinetochore or Centromere? *Nature*, 143(3629):860–860, May 1939. ISSN 1476-4687. doi: 10.1038/143860a0. URL https://www.nature.com/articles/143860a0.

[25] Jane S Richardson. The anatomy and taxonomy of protein structure. In *Advances in protein chemistry*, volume 34, pages 167–339. Elsevier, 1981.

[26] Rebecca F. Thompson, Matthew G. Iadanza, Emma L. Hesketh, Shaun Rawson, and Neil A. Ranson. Collection, pre-processing and on-the-fly analysis of data for high-resolution, single-particle cryo-electron microscopy. *Nature Protocols*, 14 (1):100–118, 2019. ISSN 1750-2799. doi: 10.1038/s41596-018-0084-8.

[27] Sjors H.W. Scheres. RELION: Implementation of a Bayesian approach to cryo-EM structure determination. *Journal of Structural Biology*, 180(3):519–530, December 2012. ISSN 1047-8477. doi: 10.1016/j.jsb.2012.09.006. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3690530/.

[28] Allan M. Jordan. Artificial Intelligence in Drug Design—The Storm Before the Calm? *ACS Medicinal Chemistry Letters*, 9(12):1150–1152, November 2018. ISSN 1948-5875. doi: 10.1021/acsmedchemlett.8b00500. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6295848/.

[29] Ron O. Dror, Robert M. Dirks, J.P. Grossman, Huafeng Xu, and David E. Shaw. Biomolecular Simulation: A Computational Microscope for Molecular Biology. *Annual Review of Biophysics*, 41(1):429–452, 2012. doi: 10.1146/annurev-biophys-042910-155245. URL https://doi.org/10.1146/annurev-biophys-042910-155245. _eprint: https://doi.org/10.1146/annurev-biophys-042910-155245.

[30] Carol V. Robinson, Andrej Sali, and Wolfgang Baumeister. The molecular sociology of the cell. *Nature*, 450(7172):973–982, December 2007. ISSN 1476-4687. doi: 10.1038/nature06523. URL https://www.nature.com/articles/nature06523. Number: 7172.

[31] Joanna Leng, Massa Shoura, Tom C. B. McLeish, Alan N. Real, Mariann Hardey, James McCafferty, Neil A. Ranson, and Sarah A. Harris. Securing the future of research computing in the biosciences. *PLoS Computational Biology*, 15(5), May 2019. ISSN 1553-734X. doi: 10.1371/journal.pcbi.1006958. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6521984/.

[32] R. Bruce Nicklas. How Cells Get the Right Chromosomes. *Science*, 275(5300): 632–637, 1997. ISSN 0036-8075. doi: 10.1126/science.275.5300.632. URL https://science.sciencemag.org/content/275/5300/632.

[33] J. H. Hayden, S. S. Bowser, and C. L. Rieder. Kinetochores capture astral microtubules during chromosome attachment to the mitotic spindle: direct visualization in live newt lung cells. *The Journal of cell biology*, 111(3):1039–1045, September 1990. ISSN 0021-9525 1540-8140 0021-9525. doi: 10.1083/jcb.111.3.1039.

[34] Kozo Tanaka, Naomi Mukae, Hilary Dewar, Mark van Breugel, Euan K. James, Alan R. Prescott, Claude Antony, and Tomoyuki U. Tanaka. Molecular mechanisms of kinetochore capture by spindle microtubules. *Nature*, 434(7036):987–994, April 2005. ISSN 1476-4687. doi: 10.1038/nature03483. URL https://www.nature.com/articles/nature03483.

[35] Günther Woehlke and Manfred Schliwa. Kinesin Kar3 and Vik1 Go Head to Head. *Cell*, 128(6):1033–1034, March 2007. ISSN 0092-8674. doi: 10.1016/j.cell.2007.03.003. URL http://www.sciencedirect.com/science/article/pii/S0092867407003108.

[36] T L Hill. Theoretical problems related to the attachment of microtubules to kinetochores. *Proceedings of the National Academy of Sciences of the United States of America*, 82(13):4404–4408, July 1985. ISSN 0027-8424. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC391109/.

[37] Jonathan W. Armond and Matthew S. Turner. Force Transduction by the Microtubule-Bound Dam1 Ring. *Biophysical Journal*, 98(8):1598–1607, April

2010. ISSN 0006-3495. doi: 10.1016/j.bpj.2010.01.004. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2856180/.

[38] G. Civelekoglu-Scholey, D. J. Sharp, A. Mogilner, and J. M. Scholey. Model of Chromosome Motility in Drosophila Embryos: Adaptation of a General Mechanism for Rapid Mitosis. *Biophysical Journal*, 90(11):3966–3982, June 2006. ISSN 0006-3495. doi: 10.1529/biophysj.105.078691. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1459506/.

[39] Fabienne Lampert, Christine Mieck, Gregory M. Alushin, Eva Nogales, and Stefan Westermann. Molecular requirements for the formation of a kinetochore–microtubule interface by Dam1 and Ndc80 complexes. *The Journal of Cell Biology*, 200(1):21–30, January 2013. ISSN 0021-9525, 1540-8140. doi: 10.1083/jcb.201210091. URL http://jcb.rupress.org/content/200/1/21.

[40] Jae Ook Kim, Alex Zelter, Neil T. Umbreit, Athena Bollozos, Michael Riffle, Richard Johnson, Michael J. MacCoss, Charles L. Asbury, and Trisha N. Davis. The Ndc80 complex bridges two Dam1 complex rings. *eLife*, 6, February 2017. ISSN 2050-084X 2050-084X. doi: 10.7554/eLife.21069.

[41] Kozo Tanaka, Etsushi Kitamura, Yoko Kitamura, and Tomoyuki U. Tanaka. Molecular mechanisms of microtubule-dependent kinetochore transport toward spindle poles. *The Journal of Cell Biology*, 178(2):269–281, July 2007. ISSN 0021-9525. doi: 10.1083/jcb.200702141.

[42] J. Richard McIntosh, Ekaterina L. Grishchuk, Mary K. Morphew, Artem K. Efremov, Kirill Zhudenkov, Vladimir A. Volkov, Iain M. Cheeseman, Arshad Desai, David N. Mastronarde, and Fazly I. Ataullakhanov. Fibrils connect microtubule tips with kinetochores: a mechanism to couple tubulin dynamics to chromosome motion. *Cell*, 135(2):322–333, October 2008. ISSN 1097-4172. doi: 10.1016/j.cell.2008.08.038.

[43] Gregory M. Alushin, Vincent H. Ramey, Sebastiano Pasqualato, David A. Ball, Nikolaus Grigorieff, Andrea Musacchio, and Eva Nogales. The Ndc80 kinetochore complex forms oligomeric arrays along microtubules. *Nature*, 467(7317):805–810, October 2010. ISSN 1476-4687. doi: 10.1038/nature09423. URL https://www.nature.com/articles/nature09423.

[44] Evgeni V. Sokurenko, Viola Vogel, and Wendy E. Thomas. Catch bond mechanism of force-enhanced adhesion: counter-intuitive, elusive but ... widespread? *Cell host & microbe*, 4(4):314–323, October 2008. ISSN 1931-3128. doi: 10.1016/j.chom.2008.09.005. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2610669/.

[45] Hironori Funabiki. Correcting aberrant kinetochore microtubule attachments: a hidden regulation of Aurora B on microtubules. *Current Opinion in Cell Biology*, 58:34–41, 2019. ISSN 1879-0410. doi: 10.1016/j.ceb.2018.12.007.

[46] Bungo Akiyoshi, Krishna K. Sarangapani, Andrew F. Powers, Christian R. Nelson, Steve L. Reichow, Hugo Arellano-Santoyo, Tamir Gonen, Jeffrey A. Ranish, Charles L. Asbury, and Sue Biggins. Tension directly stabilizes reconstituted kinetochore-microtubule attachments. *Nature*, 468(7323):576–579, November 2010. ISSN 0028-0836. doi: 10.1038/nature09594. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3108429/.

[47] Simon Jenni and Stephen C. Harrison. Structure of the DASH/Dam1 complex shows its role at the yeast kinetochore-microtubule interface. *Science*, 360(6388):552–558, 2018. ISSN 0036-8075. doi: 10.1126/science.aar6436. URL https://science.sciencemag.org/content/360/6388/552.

[48] Tomoyuki U. Tanaka, Lesley Clayton, and Toyoaki Natsume. Three wise centromere functions: see no error, hear no break, speak no delay. *EMBO reports*, 14(12):1073–1083, December 2013. ISSN 1469-3178. doi: 10.1038/embor.2013.181.

[49] Veronica Krenn and Andrea Musacchio. The Aurora B Kinase in Chromosome Bi-Orientation and Spindle Checkpoint Signaling. *Frontiers in Oncology*, 5:225, 2015. ISSN 2234-943X. doi: 10.3389/fonc.2015.00225.

[50] Ngang Heok Tang and Takashi Toda. Ndc80 Loop as a protein-protein interaction motif. *Cell Division*, 8(1):2, March 2013. ISSN 1747-1028. doi: 10.1186/1747-1028-8-2.

[51] Jerry F. Tien, Neil T. Umbreit, Alex Zelter, Michael Riffle, Michael R. Hoopmann, Richard S. Johnson, Bryan R. Fonslow, John R. Yates, Michael J.

MacCoss, Robert L. Moritz, Charles L. Asbury, and Trisha N. Davis. Kinetochore Biorientation in Saccharomyces cerevisiae Requires a Tightly Folded Conformation of the Ndc80 Complex. *Genetics*, 198(4):1483–1493, December 2014. ISSN 0016-6731, 1943-2631. doi: 10.1534/genetics.114.167775. URL https://www.genetics.org/content/198/4/1483.

[52] Karen M. May and Kevin G. Hardwick. The spindle checkpoint. *Journal of Cell Science*, 119(20):4139–4142, 2006. ISSN 0021-9533. doi: 10.1242/jcs.03165. URL http://jcs.biologists.org/content/119/20/4139.

[53] Iain M. Cheeseman and Arshad Desai. Molecular architecture of the kinetochore-microtubule interface. *Nature Reviews. Molecular Cell Biology*, 9(1):33–46, January 2008. ISSN 1471-0080. doi: 10.1038/nrm2310.

[54] Jean-François Maure, Shinya Komoto, Yusuke Oku, Akihisa Mino, Sebastiano Pasqualato, Kayo Natsume, Lesley Clayton, Andrea Musacchio, and Tomoyuki U. Tanaka. The Ndc80 loop region facilitates formation of kinetochore attachment to the dynamic microtubule plus end. *Current biology: CB*, 21(3):207–213, February 2011. ISSN 1879-0445. doi: 10.1016/j.cub.2010.12.050.

[55] Jacob D. Durrant and J. Andrew McCammon. Molecular dynamics simulations and drug discovery. *BMC biology*, 9:71, October 2011. ISSN 1741-7007. doi: 10.1186/1741-7007-9-71.

[56] David J. Huggins, Philip C. Biggin, Marc A. Dämgen, Jonathan W. Essex, Sarah A. Harris, Richard H. Henchman, Syma Khalid, Antonija Kuzmanic, Charles A. Laughton, Julien Michel, Adrian J. Mulholland, Edina Rosta, Mark S. P. Sansom, and Marc W. van der Kamp. Biomolecular simulations: From dynamics and mechanisms to computational assays of biological activity. *WIREs Computational Molecular Science*, 9(3):e1393, 2019. ISSN 1759-0884. doi: 10.1002/wcms.1393. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1393. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1393.

[57] Enrico Fermi, P Pasta, S Ulam, and M Tsingou. Studies of the nonlinear problems. Technical report, Los Alamos Scientific Lab., N. Mex., 1955.

[58] Michael Levitt and Arieh Warshel. Computer simulation of protein folding.

Nature, 253(5494):694–698, February 1975. ISSN 1476-4687. doi: 10.1038/253694a0. URL https://doi.org/10.1038/253694a0.

[59] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilard Pall, Jeremy C. Smith, Berk Hess, and Erik Lindahl. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1-2:19 – 25, 2015. ISSN 2352-7110. doi: http://dx.doi.org/10.1016/j.softx.2015.06.001. URL http://www.sciencedirect.com/science/article/pii/S2352711015000059.

[60] David A. Pearlman, David A. Case, James W. Caldwell, Wilson S. Ross, Thomas E. Cheatham, Steve DeBolt, David Ferguson, George Seibel, and Peter Kollman. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications*, 91(1):1 – 41, 1995. ISSN 0010-4655. doi: https://doi.org/10.1016/0010-4655(95)00041-D. URL http://www.sciencedirect.com/science/article/pii/001046559500041D.

[61] B R Brooks, C L Brooks, 3rd, A D Mackerell, Jr, L Nilsson, R J Petrella, B Roux, Y Won, G Archontis, C Bartels, S Boresch, A Caflisch, L Caves, Q Cui, A R Dinner, M Feig, S Fischer, J Gao, M Hodoscek, W Im, K Kuczera, T Lazaridis, J Ma, V Ovchinnikov, E Paci, R W Pastor, C B Post, J Z Pu, M Schaefer, B Tidor, R M Venable, H L Woodcock, X Wu, W Yang, D M York, and M Karplus. CHARMM: the biomolecular simulation program. *Journal of computational chemistry*, 30 (10):1545–1614, July 2009. ISSN 1096-987X. doi: 10.1002/jcc.21287. URL https://www.ncbi.nlm.nih.gov/pubmed/19444816.

[62] Siewert J Marrink, Alex H De Vries, and Alan E Mark. Coarse grained model for semiquantitative lipid simulations. *The Journal of Physical Chemistry B*, 108 (2):750–760, 2004.

[63] Adam Liwo, Maciej Baranowski, Cezary Czaplewski, Ewa Gołaś, Yi He, Dawid Jagiela, Pawel Krupa, Maciej Maciejczyk, Mariusz Makowski, Magdalena A Mozolewska, and others. A unified coarse-grained model of biological macromolec-

ules based on mean-field multipole–multipole interactions. *Journal of molecular modeling*, 20(8):2306, 2014.

[64] Andrzej Koliński and others. Protein modeling and structure prediction with a reduced representation. *Acta Biochimica Polonica*, 51, 2004.

[65] Steve Plimpton. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics*, 117(1):1–19, March 1995. ISSN 0021-9991. doi: 10.1006/jcph.1995.1039. URL http://www.sciencedirect.com/science/article/pii/S002199918571039X.

[66] Catherine L. Lawson, Matthew L. Baker, Christoph Best, Chunxiao Bi, Matthew Dougherty, Powei Feng, Glen van Ginkel, Batsal Devkota, Ingvar Lagerstedt, Steven J. Ludtke, Richard H. Newman, Tom J. Oldfield, Ian Rees, Gaurav Sahni, Raul Sala, Sameer Velankar, Joe Warren, John D. Westbrook, Kim Henrick, Gerard J. Kleywegt, Helen M. Berman, and Wah Chiu. EMDataBank.org: unified data resource for CryoEM. *Nucleic Acids Research*, 39 (suppl_1):D456–D464, January 2011. ISSN 0305-1048. doi: 10.1093/nar/gkq880. URL https://academic.oup.com/nar/article/39/suppl_1/D456/2508272.

[67] Hang Dou, Derek W. Burrows, Matthew L. Baker, and Tao Ju. Flexible Fitting of Atomic Models into Cryo-EM Density Maps Guided by Helix Correspondences. *Biophysical Journal*, 112(12):2479–2493, June 2017. ISSN 1542-0086. doi: 10.1016/j.bpj.2017.04.054.

[68] Michael G. Rossmann. Fitting atomic models into electron-microscopy maps. *Acta Crystallographica Section D*, 56(10):1341–1349, October 2000. doi: 10.1107/S0907444900009562. URL https://doi.org/10.1107/S0907444900009562.

[69] Konrad Hinsen, Nathalie Reuter, Jorge Navaza, David L. Stokes, and Jean-Jacques Lacapère. Normal Mode-Based Fitting of Atomic Structure into Electron Density Maps: Application to Sarcoplasmic Reticulum Ca-ATPase. *Biophysical Journal*, 88(2):818–827, February 2005. ISSN 0006-3495. doi: 10.1529/biophysj.104.050716. URL http://www.sciencedirect.com/science/article/pii/S0006349505731581.

[70] H. Nyquist. Thermal Agitation of Electric Charge in Conductors. *Phys. Rev.*,

32(1):110–113, July 1928. doi: 10.1103/PhysRev.32.110. URL http://link.aps.org/doi/10.1103/PhysRev.32.110.

[71] B. Hanson, S. Harris, D. Read, and O. Harlen. Fluctuating Finite Element Analysis: Development and Applications to Cytoplasmic Dynein. *Biophysical Journal*, 110:329a–330a, February 2016. doi: 10.1016/j.bpj.2015.11.1771.

[72] Glenn Stuart Peter Carrington. *The flexibility of myosin 7a*. phd, University of Leeds, September 2018. URL http://etheses.whiterose.ac.uk/22504/.

[73] Robin A. Richardson, Konstantinos Papachristos, Daniel J. Read, Oliver G. Harlen, Michael Harrison, Emanuele Paci, Stephen P. Muench, and Sarah A. Harris. Understanding the apparent stator-rotor connections in the rotary ATPase family using coarse-grained computer modeling. *Proteins: Structure, Function, and Bioinformatics*, 82(12):3298–3311, 2014. ISSN 1097-0134. doi: 10.1002/prot.24680. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.24680. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.24680.

[74] Thijs van der Heijden. *Torsion profiling of antibody proteins through experiment and molecular simulation*. Student thesis: Master, Eindhoven University of Technology, 2013. URL https://research.tue.nl/en/persons/thijs-wg-van-der-heijden/studentTheses/.

[75] W. A. Oldfather, C. A. Ellis, and Donald M. Brown. Leonhard Euler's Elastic Curves. *Isis*, 20(1):72–160, 1933. ISSN 0021-1753. URL https://www.jstor.org/stable/224885.

[76] G. Kirchoff. Ueber das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes. *Journal für die reine und angewandte Mathematik*, 1859(56):285–313, 2009. ISSN 1435-5345. doi: 10.1515/crll.1859.56.285. URL https://www.degruyter.com/view/j/crll.1859.issue-56/crll.1859.56.285/crll.1859.56.285.xml.

[77] Ellis Harold Dill. Kirchhoff's theory of rods. *Archive for History of Exact Sciences*, 44(1):1–23, March 1992. ISSN 1432-0657. doi: 10.1007/BF00379680. URL https://doi.org/10.1007/BF00379680.

[78] F Frenet. Sur les courbes a double courbure. *Journal de mathématiques pures et appliquées*, 17:437–447, 1852. ISSN 0021-7874. URL http://eudml.org/doc/233946.

[79] J-A Serret. Sur quelques formules relatives à la théorie des courbes à double courbure. *Journal de mathématiques pures et appliquées*, 16:193–207, 1851. ISSN 0021-7874. URL http://eudml.org/doc/235002.

[80] Kevin Linka and Mikhail Itskov. Mechanics of collagen fibrils: A two-scale discrete damage model. *Journal of the Mechanical Behavior of Biomedical Materials*, 58:163–172, May 2016. ISSN 1751-6161. doi: 10.1016/j.jmbbm.2015.08.045. URL http://www.sciencedirect.com/science/article/pii/S1751616115003549.

[81] Jian-Shan Wang, Yu-Hong Cui, Takahiro Shimada, Hua-Ping Wu, and Takayuki Kitamura. Unusual winding of helices under tension. *Applied Physics Letters*, 105(4):043702, July 2014. ISSN 0003-6951. doi: 10.1063/1.4892014. URL http://aip.scitation.org/doi/abs/10.1063/1.4892014.

[82] Christopher Prior. Helical Birods: An Elastic Model of Helically Wound Double-Stranded Rods. *Journal of Elasticity*, 117(2):231–277, December 2014. ISSN 1573-2681. doi: 10.1007/s10659-014-9472-7. URL https://doi.org/10.1007/s10659-014-9472-7.

[83] David Swigon, Bernard D. Coleman, and Irwin Tobias. The Elastic Rod Model for DNA and Its Application to the Tertiary Structure of DNA Minicircles in Mononucleosomes. *Biophysical Journal*, 74(5):2515–2530, May 1998. ISSN 0006-3495. doi: 10.1016/S0006-3495(98)77960-3. URL http://www.sciencedirect.com/science/article/pii/S0006349598779603.

[84] M. L. Smith and T. J. Healey. Predicting the onset of DNA supercoiling using a non-linear hemitropic elastic rod. *International Journal of Non-Linear Mechanics*, 43(10):1020–1028, December 2008. ISSN 0020-7462. doi: 10.1016/j.ijnonlinmec.2008.07.001. URL http://www.sciencedirect.com/science/article/pii/S0020746208001364.

[85] Yang Yang, Irwin Tobias, and Wilma K. Olson. Finite element analysis of DNA supercoiling. *The Journal of Chemical Physics*, 98(2):1673–1686, January 1993.

ISSN 0021-9606. doi: 10.1063/1.464283. URL https://aip.scitation.org/doi/10.1063/1.464283.

[86] Prashant K. Purohit. Plectoneme formation in twisted fluctuating rods. *Journal of the Mechanics and Physics of Solids*, 56(5):1715–1729, May 2008. ISSN 0022-5096. doi: 10.1016/j.jmps.2007.12.008. URL http://www.sciencedirect.com/science/article/pii/S0022509607002293.

[87] S. Goyal, N. C. Perkins, and C. L. Lee. Nonlinear dynamics and loop formation in Kirchhoff rods with implications to the mechanics of DNA and cables. *Journal of Computational Physics*, 209(1):371–389, October 2005. ISSN 0021-9991. doi: 10.1016/j.jcp.2005.03.027. URL http://www.sciencedirect.com/science/article/pii/S002199910500183X.

[88] Enrico Skoruppa, Michiel Laleman, Stefanos K. Nomidis, and Enrico Carlon. DNA elasticity from coarse-grained simulations: The effect of groove asymmetry. *The Journal of Chemical Physics*, 146(21):214902, June 2017. ISSN 0021-9606. doi: 10.1063/1.4984039. URL https://aip.scitation.org/doi/10.1063/1.4984039.

[89] Raymond E. Goldstein and Stephen A. Langer. Nonlinear Dynamics of Stiff Polymers. *Physical Review Letters*, 75(6):1094–1097, August 1995. doi: 10.1103/PhysRevLett.75.1094. URL https://link.aps.org/doi/10.1103/PhysRevLett.75.1094.

[90] Dinesh K. Pai. STRANDS: Interactive Simulation of Thin Solids using Cosserat Models. *Computer Graphics Forum*, 21(3):347–352, 2002. ISSN 1467-8659. doi: 10.1111/1467-8659.00594. URL http://dx.doi.org/10.1111/1467-8659.00594.

[91] Sunil Hadap. Oriented Strands: Dynamics of Stiff Multi-body System. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 91–100, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. ISBN 978-3-905673-34-0. URL http://dl.acm.org/citation.cfm?id=1218064.1218077.

[92] Jonas Spillmann and Matthias Teschner. C o R d E: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proceedings of the*

2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 63–72. Eurographics Association, 2007.

[93] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. In *ACM transactions on graphics (TOG)*, volume 27, page 63. ACM, 2008.

[94] P. Grassia and E. J. Hinch. Computer simulations of polymer chain relaxation via Brownian motion. *Journal of Fluid Mechanics*, 308:255–288, February 1996. ISSN 1469-7645, 0022-1120. doi: 10.1017/S0022112096001474. URL https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/computer-simulations-of-polymer-chain-relaxation-via-brownian-motion/939902CEBD38EC6DB76B6ECD0DBD29EF.

[95] David C. Morse. Theory of Constrained Brownian Motion. In *Advances in Chemical Physics*, pages 65–189. 2004. ISBN 978-0-471-48423-3. doi: 10.1002/0471484237.ch2. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/0471484237.ch2. Section: 2 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471484237.ch2.

[96] Tanniemola B. Liverpool. Dynamics of inextensible semiflexible filaments. *Physical Review E*, 72(2):021805, August 2005. doi: 10.1103/PhysRevE.72.021805. URL https://link.aps.org/doi/10.1103/PhysRevE.72.021805.

[97] Artem Zhmurov, Olga Kononova, Rustem I Litvinov, Ruxandra I Dima, Valeri Barsegov, and John W Weisel. Mechanical Transition from alpha-Helical Coiled-Coils to beta-Sheets in Fibrin(ogen). *Journal of the American Chemical Society*, 134(50):20396–20402, December 2012. ISSN 0002-7863. doi: 10.1021/ja3076428. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3526676/.

[98] Olinde Rodrigues. *Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace: et de la variation des cordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire.* 1840.

[99] Jur van den Berg (https://math.stackexchange.com/users/91768/jur-van-den berg). *Calculate Rotation Matrix to align Vector A to Vector B in 3d?* URL https://math.stackexchange.com/q/476311. Via Mathematics Stack Exchange.

[100] PG Saffman. Brownian motion in thin sheets of viscous fluid. *Journal of Fluid Mechanics*, 73(4):593–602, 1976.

[101] Ron Milo, Paul Jorgensen, Uri Moran, Griffin Weber, and Michael Springer. BioNumbers—the database of key numbers in molecular and cell biology. *Nucleic Acids Research*, 38(Database issue):D750–D753, January 2010. ISSN 0305-1048. doi: 10.1093/nar/gkp889. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2808940/.

[102] L. D. Landau and E. M. Lifshitz. Chapter I - the Fundamental Principles of Statistical Physics. In L.D. Landau and E.M. Lifshitz, editors, *Statistical Physics (Third Edition, Revised and Enlarged)*, pages 1 – 33. Butterworth-Heinemann, Oxford, third edition, revised and enlarged edition, 1980. ISBN 978-0-08-057046-4. doi: 10.1016/B978-0-08-057046-4.50008-7. URL https://www.sciencedirect.com/science/article/pii/B9780080570464500087.

[103] Tae Yeon Yoo, Jeong-Mo Choi, William Conway, Che-Hang Yu, Rohit V. Pappu, and Daniel J. Needleman. Data from: Measuring NDC80 binding reveals the molecular basis of tension-dependent kinetochore-microtubule attachments, 2019. URL http://datadryad.org/stash/dataset/doi:10.5061/dryad.14rr125. type: dataset.

[104] kinetochore protein Nuf2 [Homo sapiens] - Protein - NCBI, . URL https://www.ncbi.nlm.nih.gov/protein/NP_113611.2.

[105] Christopher W Wood, Jack W Heal, Andrew R Thomson, Gail J Bartlett, Amaurys Ã Ibarra, R Leo Brady, Richard B Sessions, and Derek N Woolfson. ISAMBARD: an open-source computational environment for biomolecular analysis, modelling and design. *Bioinformatics*, 33(19):3043–3050, 2017. doi: 10.1093/bioinformatics/btx352. URL http://dx.doi.org/10.1093/bioinformatics/btx352.

[106] Georgii G. Krivov, Maxim V. Shapovalov, and Roland L. Dunbrack. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795. doi: 10.1002/prot.22488. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.22488.

[107] Dong Xu and Yang Zhang. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Structure, Function, and Bioinformatics*, 80(7):1715–1735, 2012. ISSN 1097-0134. doi: 10.1002/prot.24065. URL http://dx.doi.org/10.1002/prot.24065.

[108] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Gregory S Couch, Daniel M Greenblatt, Elaine C Meng, and Thomas E Ferrin. UCSF Chimera - a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, 2004.

[109] Dong Xu and Yang Zhang. Improving the Physical Realism and Structural Accuracy of Protein Models by a Two-Step Atomic-Level Energy Minimization. *Biophysical Journal*, 101(10):2525–2534, November 2011. ISSN 0006-3495. doi: 10.1016/j.bpj.2011.10.024. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3218324/.

[110] Elmar Krieger, Keehyoung Joo, Jinwoo Lee, Jooyoung Lee, Srivatsan Raman, James Thompson, Mike Tyka, David Baker, and Kevin Karplus. Improving physical realism, stereochemistry, and side-chain accuracy in homology modeling: Four approaches that performed well in CASP8. *Proteins*, 77 Suppl 9:114–122, 2009. ISSN 1097-0134 0887-3585. doi: 10.1002/prot.22570.

[111] James A. Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E. Hauser, and Carlos Simmerling. ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *Journal of Chemical Theory and Computation*, 11(8):3696–3713, August 2015. ISSN 1549-9618. doi: 10.1021/acs.jctc.5b00255. URL https://doi.org/10.1021/acs.jctc.5b00255.

[112] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720. URL https://doi.org/10.1080/14786440109462720.

[113] Ardita Shkurti, Ramon Goni, Pau Andrio, Elena Breitmoser, Iain Bethune, Modesto Orozco, and Charles A. Laughton. pyPcazip: A PCA-based toolkit for compression and analysis of molecular simulation data. *SoftwareX*, 5:44–

50, January 2016. ISSN 2352-7110. doi: 10.1016/j.softx.2016.04.002. URL http://www.sciencedirect.com/science/article/pii/S2352711016300036.

[114] Xiaohu Hu, Liang Hong, Micholas Dean Smith, Thomas Neusius, Xiaolin Cheng, and Jeremy C. Smith. The dynamics of single protein molecules is non-equilibrium and self-similar over thirteen decades in time. *Nature Physics*, 12(2):171–174, February 2016. ISSN 1745-2481. doi: 10.1038/nphys3553. URL https://www.nature.com/articles/nphys3553.

[115] Irina Kufareva and Ruben Abagyan. Methods of protein structure comparison. *Methods in molecular biology (Clifton, N.J.)*, 857:231–257, 2012. ISSN 1064-3745. doi: 10.1007/978-1-61779-588-6_10. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4321859/.

[116] Erhard Schmidt. Zur Theorie der linearen und nicht linearen Integralgleichungen Zweite Abhandlung. *Mathematische Annalen*, 64(2):161–174, 1907.

[117] Go Itoh, Masanori Ikeda, Kenji Iemura, Mohammed Abdullahel Amin, Sei Kuriyama, Masamitsu Tanaka, Natsuki Mizuno, Hiroko Osakada, Tokuko Haraguchi, and Kozo Tanaka. Lateral attachment of kinetochores to microtubules is enriched in prometaphase rosette and facilitates chromosome alignment and bi-orientation establishment. *Scientific Reports*, 8(1):3888, March 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-22164-5. URL https://www.nature.com/articles/s41598-018-22164-5.

[118] Roberto Valverde, Jessica Ingram, and Stephen C. Harrison. Conserved Tetramer Junction in the Kinetochore Ndc80 Complex. *Cell Reports*, 17(8):1915–1922, 2016. ISSN 2211-1247. doi: 10.1016/j.celrep.2016.10.065.

[119] Emily M. Kudalkar, Emily A. Scarborough, Neil T. Umbreit, Alex Zelter, Daniel R. Gestaut, Michael Riffle, Richard S. Johnson, Michael J. MacCoss, Charles L. Asbury, and Trisha N. Davis. Regulation of outer kinetochore Ndc80 complex-based microtubule attachments by the central kinetochore Mis12/MIND complex. *Proceedings of the National Academy of Sciences*, 112(41):E5583–E5589, October 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1513882112. URL https://www.pnas.org/content/112/41/E5583. ISBN: 9781513882116 Section: PNAS Plus.

[120] Stephanie A. Miller, Michael L. Johnson, and P. Todd Stukenberg. Kinetochore Attachments Require an Interaction between Unstructured Tails on Microtubules and Ndc80Hec1. *Current Biology*, 18(22):1785–1791, November 2008. ISSN 0960-9822. doi: 10.1016/j.cub.2008.11.007. URL http://www.sciencedirect.com/science/article/pii/S096098220801484X.

[121] Geoffrey J. Guimaraes, Yimin Dong, Bruce F. McEwen, and Jennifer G. DeLuca. Kinetochore-Microtubule Attachment Relies on the Disordered N-Terminal Tail Domain of Hec1. *Current Biology*, 18(22):1778–1784, November 2008. ISSN 0960-9822. doi: 10.1016/j.cub.2008.08.012. URL http://www.sciencedirect.com/science/article/pii/S096098220801052X.

[122] Vladimir A Volkov, Pim J Huis in 't Veld, Marileen Dogterom, and Andrea Musacchio. Multivalency of NDC80 in the outer kinetochore is essential to track shortening microtubules and generate forces. *eLife*, 7:e36764, April 2018. ISSN 2050-084X. doi: 10.7554/eLife.36764. URL https://doi.org/10.7554/eLife.36764.

[123] J. E. Jones and Sydney Chapman. On the determination of molecular fields. —II. From the equation of state of a gas. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 106(738):463–477, October 1924. doi: 10.1098/rspa.1924.0082. URL https://royalsocietypublishing.org/doi/10.1098/rspa.1924.0082.

[124] William B. Redwine, Rogelio Hernández-López, Sirui Zou, Julie Huang, Samara L. Reck-Peterson, and Andres E. Leschziner. Structural Basis for Microtubule Binding and Release by Dynein. *Science*, 337(6101):1532–1536, September 2012. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1224151. URL https://science.sciencemag.org/content/337/6101/1532. Section: Report.

[125] Harinath Doodhi, Taciana Kasciukovic, Marek Gierlinski, Shuyu Li, Lesley Clayton, and Tomoyuki U. Tanaka. In vitro reconstitution of kinetochore–microtubule interface reveals a fundamental error correction mechanism. *BioRxiv*, October 2018. doi: 10.1101/455873.

[126] Edward D. Salmon and Kerry Bloom. Tension sensors reveal how the kinetochore shares its load. *BioEssays*, 39(7):1600216,

2017. ISSN 1521-1878. doi: 10.1002/bies.201600216. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/bies.201600216. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bies.201600216.

[127] Frank Uhlmann. SMC complexes: from DNA to chromosomes. *Nature Reviews Molecular Cell Biology*, 17(7):399–412, July 2016. ISSN 1471-0080. doi: 10.1038/nrm.2016.30. URL https://www.nature.com/articles/nrm.2016.30.

[128] Kathleen M. Trybus. Myosin V from head to tail. *Cellular and molecular life sciences : CMLS*, 65(9):1378–1389, May 2008. ISSN 1420-682X. doi: 10.1007/s00018-008-7507-6. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2613318/.

[129] Walter Gellert, M Hellwich, H Kästner, and H Küstner. *The VNR concise encyclopedia of mathematics*. Springer Science & Business Media, 2 edition, 2012. ISBN 978-94-011-6984-4. URL 10.1007/978-94-011-6982-0.

[130] Cyrille Ka. How can you determine a point is between two other points on a line segment? URL https://stackoverflow.com/questions/328107/how-can-you-determine-a-point-is-between-two-other-points-on-a-line-segment. Library Catalog: stackoverflow.com.

[131] Wm. Randolph Franklin and Mohan S. Kankanhalli. Volumes from overlaying 3-D triangulations in parallel. In David Abel and Beng Chin Ooi, editors, *Advances in Spatial Databases*, Lecture Notes in Computer Science, pages 477–489, Berlin, Heidelberg, 1993. Springer. ISBN 978-3-540-47765-5. doi: 10.1007/3-540-56869-7_27.

[132] Fabio Ganovelli, Federico Ponchio, and Claudio Rocchini. Fast Tetrahedron-Tetrahedron Overlap Algorithm. *Journal of Graphics Tools*, 7(2):17–25, January 2002. ISSN 1086-7651. doi: 10.1080/10867651.2002.10487557. URL https://doi.org/10.1080/10867651.2002.10487557.

[133] Gene M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, AFIPS '67 (Spring), pages 483–485, New York, NY, USA, April 1967. ISBN 978-1-4503-7895-6. doi: 10.1145/1465482.1465560. URL https://doi.org/10.1145/1465482.1465560.

[134] Raymond Greenlaw, H James Hoover, Walter L Ruzzo, and others. *Limits to parallel computation: P-completeness theory.* Oxford University Press on Demand, 1995.

[135] Tim Meyer, Marco D'Abramo, Adam Hospital, Manuel Rueda, Carles Ferrer-Costa, Alberto Pérez, Oliver Carrillo, Jordi Camps, Carles Fenollosa, Dmitry Repchevsky, Josep Lluis Gelpí, and Modesto Orozco. MoDEL (Molecular Dynamics Extended Library): A Database of Atomistic Molecular Dynamics Trajectories. *Structure*, 18(11):1399 – 1409, 2010. ISSN 0969-2126. doi: https://doi.org/10.1016/j.str.2010.07.013. URL http://www.sciencedirect.com/science/article/pii/S0969212610003539.

[136] P.J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992. ISSN 1939-3539. doi: 10.1109/34.121791.

[137] JR Cooper and RB Dooley. Release of the IAPWS formulation 2008 for the viscosity of ordinary water substance. *The International Association for the Properties of Water and Steam, Berlin, Germany*, 2008.