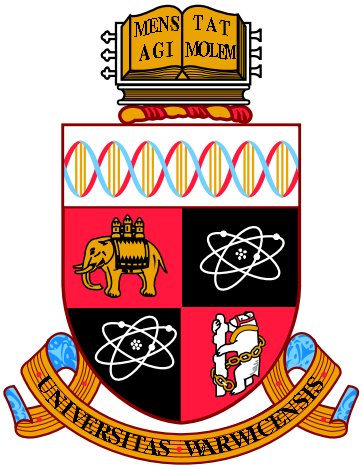

Wavelet Monte Carlo Dynamics

by

Oliver T. Dyer

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Physics

April 2019

Contents

List of Tables **iv**

List of Figures **v**

Acknowledgments **vii**

Declarations **viii**

Abstract **ix**

Abbreviations **x**

Chapter 1 Introduction **1**

Chapter 2 Background **5**

2.1 The Stokes equation and the Oseen tensor 5
2.2 The overdamped Langevin equation 8
2.2.1 Rotational Langevin equations 9
2.3 Diffusion and velocity autocorrelations 11
2.4 Polymer models 12
2.4.1 Rouse and Zimm models 12
2.4.2 Solvent quality and the Flory exponent 13
2.5 Review of Brownian dynamics 15
2.5.1 Unbounded systems 16
2.5.2 Periodic systems 18
2.5.3 Final remarks on Brownian dynamics 19
2.6 Monte Carlo methods 20
2.6.1 Detailed balance and equilibrium 20
2.6.2 Static and dynamic MC 21
2.6.3 Smart MC 22
2.6.4 Accuracy of MC tests 23
2.7 Wavelets 25
2.7.1 1-dimensional wavelets 25

2.7.2 3-dimensional wavelets 28

Chapter 3 Wavelet Monte Carlo dynamics **30**

3.1 Wavelet representation of the Oseen tensor 30
3.2 Wavelet parameter distributions 33
3.2.1 Wavelet orientation and centre 34
3.2.2 Wavelet amplitude 35
3.2.3 Wavelet radii and the effect of finite cut-offs 38
3.3 Fourier representation of the missing mobility 42
3.4 Fourier parameter distributions 44
3.4.1 Fourier amplitude 44
3.4.2 Wavevectors 46
3.5 Time evolution and the probability of making a Fourier move 47
3.5.1 Validation of time in WMCD 50
3.6 Schematic of the WMCD algorithm 53
3.7 Computational cost 55
3.7.1 Computational cost of Fourier moves 55
3.7.2 Computational cost of wavelet moves 56
3.8 Final remarks on the core WMCD algorithm 59

Chapter 4 Further developments for WMCD **61**

4.1 Smart Wavelet Monte Carlo dynamics 61
4.1.1 Smart wavelet moves 61
4.1.2 Smart Fourier moves 62
4.1.3 Verifying hydrodynamics in smart WMCD 63
4.2 Polydisperse WMCD 64
4.2.1 Summary of algorithmic procedure 66
4.2.2 Comments on accuracy and validity 67
4.3 Rotations in WMCD 68
4.3.1 The rotation-translation tensor 69
4.3.2 The rotation-rotation tensor 70
4.3.3 Rotation-only moves 71
4.3.4 Biasing by torques in smart WMCD 74
4.4 Comparing choices of mother wavelet 75
4.4.1 Definitions of wavelets considered 75
4.4.2 Time and computational cost 79
4.4.3 Details of mobility tensors 79
4.4.4 Distribution of Fourier moves 82
4.4.5 Final remarks on wavelet choice 84
4.5 Summary of the current state of WMCD and future paths 85

Chapter 5 Time-dependent polymer diffusivity **87**
5.1 Introduction . . . 87
5.2 Fixman's expression for the change in diffusivity . . . 89
5.3 Data acquisition in MC algorithms . . . 91
5.3.1 Calculating **A** in smart WMCD . . . 92
5.3.2 Calculating **A** in non-smart WMCD . . . 92
5.4 Simulation details and polymer systems . . . 94
5.4.1 Potentials and units . . . 94
5.4.2 Other system perameters . . . 95
5.5 Properties of scaling variables . . . 95
5.5.1 Scaling of the Kirkwood diffusivity . . . 96
5.5.2 Scaling of the Zimm time . . . 96
5.5.3 Scaling of $\langle A^2 \rangle$ . . . 97
5.6 Short-time behaviour of $\langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle$ . . . 99
5.7 Long-time behaviour of $\langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle$ . . . 103
5.8 Long-chain limit of ΔD . . . 107
5.9 Summary . . . 109

Chapter 6 Motion of passive particles in active systems **111**
6.1 Swimmer models . . . 111
6.1.1 Run-and-tumble motion in WMCD . . . 114
6.1.2 The MC acceptance test in active systems . . . 117
6.1.3 System parameters . . . 118
6.2 Low-noise trajectories . . . 118
6.3 Diffusivity of passive particles in active systems . . . 123
6.3.1 D_A with varying a_p and swimmer concentration . . . 124
6.3.2 D_A with varying a_p and k_BT . . . 128
6.3.3 Comments on the effect of including excluded volume interactions . . . 130
6.4 Summary . . . 131

List of Tables

4.1 Important numerical quantities associated with mother wavelets . . 78

6.1 Parameters used in simulations of swimmers 118

List of Figures

1.1 Stokeslet flow field 2

2.1 Example polymer represented as a flexible chain of beads 13
2.2 Metropolis and smart Monte Carlo rejection probabilities 24
2.3 Example 1-dimensional wavelets 26
2.4 1-dimensional signals and their continuous wavelet transforms 28

3.1 Example wavelet in WMCD 33
3.2 Move acceptance rates at different wavelet radii 37
3.3 Components of the translational mobility tensor in WMCD 41
3.4 Move acceptance rates at different wavenumbers 45
3.5 Relaxtation of a polymer's radius of gyration 51
3.6 Centre of mass diffusivity of swollen polymers 52
3.7 Effect of box size on particle diffusivity 52
3.8 Computational cost in homogeneous systems 57
3.9 Computational cost in fractal systems 59

4.1 Plots of integrals appearing in rotational mobility tensors in WMCD 70
4.2 Schematic of move types at different wavelet radii 72
4.3 Derivatives of scalar wavelets 76
4.4 Values of Υ for trigonometric wavelets 78
4.5 Mobility tensors for different mother wavelets, normalised at $r = 0$. 81
4.6 Mobility tensors for different mother wavelets, normalised at $r \to \infty$ 82
4.7 Distributions of Fourier moves for different mother wavelets 83

5.1 $\langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle$ measured in non-smart Glauber WMCD relative to smart WMCD 94
5.2 Kirkwood diffusivity of polymer chains of different length 96
5.3 Zimm time of polymer chains of different length 97
5.4 $\langle A^2 \rangle$ of polymer chains of different length 98
5.5 Polymer velocity autocorrelations 100
5.6 Autocorrelation of forces in dimers 101

5.7 Change in polymer diffusivity over time 104
5.8 Collapse of the long-time tail of $\langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle$ 106
5.9 Extrapolation of the fractional decrease in diffusivity to long chains . 108

6.1 The 2-force swimmer model 113
6.2 Idealised passive particle trajectories 119
6.3 Flow fields relative to swimmers as seen by particles of different sizes 121
6.4 Loop sizes for large passive particles 122
6.5 Active diffusivity at different swimmer concentrations and particle sizes 125
6.6 Example velocity autocorrelations in active systems 126
6.7 Example trajectory of a particle trapped in closed stream lines . . . 127
6.8 Active diffusivity at different temperatures and particle sizes 129

Acknowledgments

First and foremost, thank you to my supervisor Robin Ball; I am struggling to put into words just how grateful I am for his support and guidance over the years. Huge thanks also to the Warwick Scientific Computing RTP for its silent yet ever-present contribution of the computational resources needed throughout the project. I am also grateful to the physics theory and soft-bio. groups for over 3 years of interesting talks and discussions. Likewise, albeit a teeny bit further afield, thanks to Ravi Jagadeeshan for his support during and since my visit to Melbourne.

Solitary and impassive as I often am, I am nonetheless thankful for friends across the physics department and beyond for doing what friends do (for want of a better way of saying it). Particular thanks go to Antonino, Jack and Lewis who chance had me work alongside, and who were always happy to indulge me in conversations about Pokémon, Miracleland, or occasionally something more philosophical.

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. The work presented (including data generated and data analysis) was carried out by the author.

Parts of this thesis have been published by the author:

- O. T. Dyer and R. C. Ball. Wavelet Monte Carlo dynamics: A new algorithm for simulating the hydrodynamics of interacting Brownian particles. *The Journal of Chemical Physics*, 146(12):124111, 2017
- O. T. Dyer and R. C. Ball. Time reversal of the overdamped Langevin equation and Fixman's law. `http://wrap.warwick.ac.uk/116261/`. Uploaded: 15/04/2019

Much of the work in Chapters 4, 5 and 6 is also being prepared for publication.

Abstract

The Wavelet Monte Carlo dynamics (WMCD) algorithm is developed from scratch to simulate hydrodynamically coupled Brownian particles at low Reynolds numbers. The basic premise is to construct a regularised version of the Oseen tensor out of a distribution of 3-dimensional vector wavelets that displace groups of particles so as to evolve the system in time while correlating the motion of all particles according to their separation. In doing so, the Oseen tensor is made implicit in the simulation code and the computational complexity of WMCD scales with system size N as $N \ln N$ (or even linearly in fractal systems), comparing favourably to existing Brownian dynamics algorithms, while the absence of any solvent degrees of freedom also leads to favourable comparisons with explicit-solvent methods. WMCD therefore holds promise to simulate system sizes beyond the reach of the alternatives. Key extensions to the basic algorithm are presented - none of which affect the computational complexity - including additional Fourier moves and smart Monte Carlo biasing to improve the algorithm's dynamical fidelity, as well as schemes to build in polydispersity and hydrodynamic coupling of particle rotations. WMCD is then used in a comprehensive study of the diffusion of isolated polymer chains, using the properties of the centre of mass velocity autocorrelation to identify distinct short and long-time regimes driving the reduction of diffusivity from the Kirkwood value. Using similar methods in a very different context, WMCD is also used to study the enhanced diffusion of passive particles in active suspensions. Here again the velocity autocorrelation proves useful in understanding the underlying physics, with three driving mechanisms identified depending on relative particle sizes. Of particular note is the importance of thermal fluctuations, often neglected in active matter research but central to WMCD.

Abbreviations

BD	Brownian dynamics
CDF	Cumulative distribution function
CWT	Continuous wavelet transform
DMC	Dynamic Monte Carlo
EV	Excluded volume
FDT	Fluctuation-dissipation theorem
FENE	Finitely extensible non-linear elastic
HIs	Hydrodynamic interactions
LB	Lattice Boltzmann
MC	Monte Carlo
MD	Molecular dynamics
PDF	Probability distribution function
RPY	Rotne-Prager-Yamakawa
SMC	Smart Monte Carlo
SPME	Smooth particle mest Ewald
TEA	Truncated expansion ansatz
WCA	Weeks-Chandler-Anderson
WMCD	Wavelet Monte Carlo dynamics

I don't know how it happened
It all took place so quick
But all I can do is hand it to you
And your latest trick

Mark Knopfler, Dire Straits, *Your Latest Trick*
(taken a long way from its original context)

1 Introduction

The term 'soft matter' categorises materials with mesoscopic structure - larger than the microscopic scales of the constituent molecules yet smaller than the macroscopic scales of the system as a whole - and have a high compliance, being easily deformed by thermal fluctuations or external stresses [3, 4, 5]. This encompasses a range of systems far broader than the scope of a single thesis, and this work will narrow down to 3-dimensional fluids predominantly comprised of a simple liquid solvent, but where the behaviour of the solute is the primary interest. These systems are at low enough concentration for flows in the solvent to play an important role in the dynamics of the solute.

Furthermore, only the low Reynolds number regime, typical of systems at scales of micrometres and below, will be considered. Far from the often inertia-dominated macroscopic world where arrows keep flying, wheels keep turning and pendulums keep swinging unless something gets in the way, at low Reynolds numbers viscosity dominates and all three would come to a stop almost immediately. To keep moving a force needs to be continually applied, leading to a constant velocity without acceleration. For a hard sphere of radius a being pushed by a force $\mathbf{F}$ through a fluid of viscosity η this is given by the well known Stokes law [6]

$$\mathbf{v} = \frac{1}{6\pi\eta a}\mathbf{F}. \tag{1.1}$$

The physics behind this is much the same as for terminal velocity when falling in air.

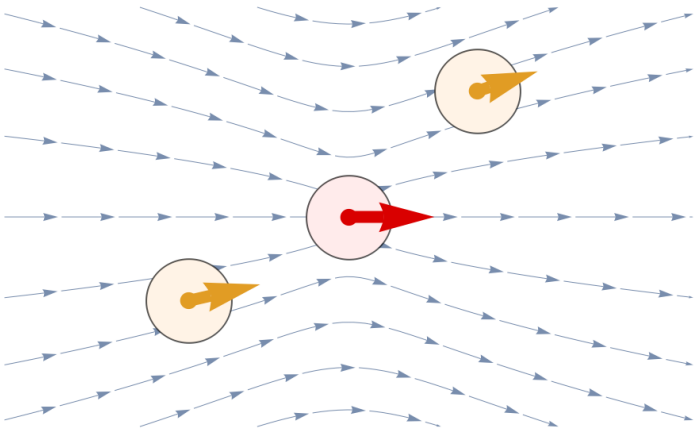

Figure 1.1: Diagram of stream lines in the Stokeslet generated by the force at the origin (red). The velocity of the particle at the origin is given by Eq. (1.1), while the other two particles move with the flow field.

Treacle or similarly viscous everyday liquids are commonly used to help build an intuition of the viscous world. Indeed it is easy to imagine the arrow, wheel or pendulum quickly coming to a stop if fired at or dropped into a vat of treacle, but there is more going on than individual objects stopping: the motions of separated objects are also correlated by fluid flows. The existence of such flows is familiar even in less viscous fluids, with examples including the blast of air that greets you when a fast-moving train passes a station, and the difficulty of pinching a small floating object in water. What is less obvious is the long range and structure of these flows, which for the point force $\mathbf{F}$ in the bulk of the fluid are given by the 'Stokeslet' [7]

$$\mathbf{u}(\mathbf{r}) = \frac{\mathbf{F}}{8\pi\eta r} \cdot (\mathbf{I} + \hat{\mathbf{r}} \otimes \hat{\mathbf{r}}) \,, \tag{1.2}$$

for which stream lines are shown in Fig. 1.1. As opposed to the train scenario where you would expect to stay planted to the spot if you stand a safe distance away, the velocity in the Stokeslet applies to everything in the fluid, *regardless of how heavy it is*. In other words, if one thing moves in a viscous fluid, everything moves. This statement is oversimplified of course, but thanks to the slow decay of the Stokeslet it actually applies pretty well to all viscous fluids in bulk and highlights a major challenge for research in this field.

The feature of mesoscopic soft matter that is most difficult to picture with macroscopic examples is the effect of thermal fluctuations, as any macroscopic object is simply too large to have any appreciable Brownian motion. Putting the object in treacle during an earthquake goes some way to capturing the essential details however, namely that it induces random, unpredictable motion that is correlated at

different positions[1]. Collectively the properties above form the basic building blocks of low Reynolds number fluids, and they apply to a vast array of systems, including generic physical systems such as polymers [8, 9] and liquid crystals [10, 11, 12] along with biological materials at both cellular [13, 14] and sub-cellular scales [15, 16, 17].

As with all of physics, efforts to investigate these systems employ all of experiments, analytic theory and computational simulations, and these are plagued by the usual suspects: the difficulty of making measurements in the dream experimental set-up; the intractability of all but the simplest test systems; and the time it takes to run an accurate simulation. The work in this thesis takes the simulation route, and low Reynolds number physics has been painted in broad strokes because the core of this thesis is not the understanding of a particular physical system, but the development of an efficient numerical algorithm to apply to the field as a whole.

Simulation techniques can be broadly categorised into implicit- and explicit-solvent methods, which, as their names suggest, are distinguished by whether they take direct account of the solvent or not. The approach taken in explicit-solvent methods varies greatly, with Molecular dynamics (MD) being the most obvious as it solves Newton's laws of motion for all the solvent and solute particles [18]. By mimicking what actually happens at microscopic scales MD is almost guaranteed to replicate real phenomena, but it does so at great cost to computational efficiency as simulating so many particles is extremely intensive. This is especially egregious given the behaviour of the solvent molecules is of little interest in its own right.

All explicit-solvent methods pay the price of simulating solvent degrees of freedom, but they can be more efficient about it than MD. Dissipative particle dynamics coarse grains the solvent into a smaller number of aggregate particles with softer interaction potentials [19, 20, 21, 22, 23], while Multiparticle collision dynamics removes potentials between solvent particles altogether and randomises velocities in local centre of mass frames [24, 25, 26, 27]. Meanwhile, by solving the Boltzmann equation Lattice Boltzmann (LB) tracks particle densities instead of solvent particles per se [28, 29, 30]. Care must be taken in each case to correctly equilibrate and incorporate thermal fluctuations, but these methods are not the focus of this thesis so will not be covered in detail. The most relevant detail is that the complexity of all these methods (i.e. how the run time of simulations increases with system size) is linear with the total number of particles[2], both in the solvent and the more interesting solute.

On the other side lie the implicit-solvent methods, which for hydrodynami-

[1]The thermal correlations are governed by the Stokeslet, or more carefully the Oseen tensor from which the Stokeslet inherits its structure. Long ranged as this is, the correlations in an earthquake are more likely to be a constant across the system, revealing one weakness in the analogy.

[2]In LB the number of 'particles' corresponds to the number of lattice sites. The two are proportional given some average fluid density.

cally coupled systems are dominated by Brownian dynamics (BD) [31]. In BD the solvent is replaced with a $3N \times 3N$ mobility tensor coupling the motion of all N solute particles. Without going into the details, this tensor essentially encodes Eq.s (1.1) and (1.2) for the entire system. Removing the solvent degrees of freedom is a huge advantage, but there is a catch: the full mobility tensor can be very expensive to work with, especially for correlating thermal fluctuations.

Much recent work has gone into improving the performance of BD [32, 33, 34, 35, 36], but it remains standard to use an algorithm set out by Fixman back in 1986 whose time complexity rises with system size as $N^{2.25}$ [37]. Comparing this to the linear scaling in the explicit-solvent methods presents a choice: accept the cost of tracking the solvent, or remove the solvent at the price of the computation time rising rapidly with system size. As a rule of thumb, BD and explicit-solvent methods tend to win in more dilute and concentrated systems respectively [34, 38], but both struggle to simulate large systems.

It is here that the work in this thesis wades in and presents (yet) another alternative based on neither BD nor any of the explicit-solvent methods, but on exploiting a wavelet representation of the hydrodynamic correlations to produce a dynamic Monte Carlo algorithm. This approach, called Wavelet Monte Carlo dynamics (WMCD) [1], makes implicit not only the solvent molecules but also the calculation that bogs down BD. The end result being a cost that scales almost linearly.

The thesis proceeds as follows: Chapter 2 runs almost parallel to this introduction, presenting the mathematics of the hydrodynamic equations before providing a more detailed review of BD with a mind to highlight the primary challenges and to emphasise the value of developing a new algorithm from scratch. Introductions to wavelets and Monte Carlo algorithms are then presented in preparation for WMCD.

Chapters 3 and 4 are both devoted to describing the WMCD algorithm in detail. Chapter 3 will focus on the fundamentals, tackling the correlated random motion mentioned above and how to make a dynamic Monte Carlo algorithm out of it. Chapter 4 then presents a series of improvements and extensions, before investigating a question of computational interest: how to choose a wavelet.

Finally, Chapters 5 and 6 will put WMCD to work, first in simulating equilibrium polymer systems and then in mixed active and passive systems emulating micro-swimmers and colloids. The results therein offer insights into the physics of diffusion in those systems, while hinting at the broad range of applicability of WMCD in soft-matter research.

2 Background

2.1 The Stokes equation and the Oseen tensor

This work uses a continuum description for simple fluids, for which two equations will be needed. First mass conservation provides the continuity equation [39]

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2.1}$$

with $\mathbf{u} = \mathbf{u}(\mathbf{r}, t)$ the fluid velocity and ρ the fluid density. Under the incompressible fluid approximation ρ is constant in both space and time and this reduces to

$$\nabla \cdot \mathbf{u} = 0. \tag{2.2}$$

Second is the Navier-Stokes equation [39]

$$\rho \partial_t \mathbf{u} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \eta \nabla^2 \mathbf{u} + \mathbf{f} \tag{2.3}$$

describing momentum transfer in the fluid accounting for the influence of both inertia and viscous drag. In this p is the pressure, η the dynamic viscosity and $\mathbf{f}$ a force per unit volume.

In its full non-linear form Eq. (2.3) is beyond the scope of this thesis, and instead the focus is on the Stokes equation wherein the left hand side is neglected. The formal procedure to reach this offers insights into the regime being studied and is worth going through. To start, a characteristic speed u_0 and length scale ℓ are

introduced to non-dimensionalise Eq. (2.3) via $t^* = u_0 t/\ell$, $\mathbf{u}^* = \mathbf{u}/u_0$, $p^* = (\ell/\eta u_0)p$, $\mathbf{f}^* = \mathbf{f}/\eta u_0 \ell$, and $\nabla^* = \ell\nabla$, giving

$$Re\,(\partial_{t^*}\mathbf{u}^* + \mathbf{u}^* \cdot \nabla^*\mathbf{u}^*) = -\nabla^* p^* + \nabla^{*2}\mathbf{u}^* + \mathbf{f}^* \tag{2.4}$$

with the Reynolds number

$$Re = \frac{\rho u_0 \ell}{\eta}. \tag{2.5}$$

If ℓ is chosen to be the typical length over which $\mathbf{u}$ varies, the right hand side of Eq. (2.4) is of order unity and the non-linear term can be neglected if $Re \ll 1$. This will be the case in the systems studied later where viscosity dominates over inertia, though it is worth noting that in these systems it is the small ℓ and u_0 that make Re small rather than a particularly large value of η. Indeed in macroscopic units ρ/η can be large. E.g. water at room temperature has $\rho/\eta \sim 10^6\,\mathrm{s\,m^{-1}}$ but acts as a viscous fluid at microscopic scales.

Removing the acceleration term is more subtle. In fact the idealised case with zero inertia necessitates infinite acceleration and it is not clear the combination $Re\,\partial_{t^*}\mathbf{u}^*$ can be neglected at all. To progress, the remaining terms can be written as a diffusion equation for the momentum per unit volume $\rho\mathbf{u}$:

$$\partial_t(\rho\mathbf{u}) = \frac{\eta}{\rho}\nabla^2(\rho\mathbf{u}) + (\mathbf{f} - \nabla p). \tag{2.6}$$

From this one readily identifies the diffusivity of momentum $D_{\rho\mathbf{u}} = \eta/\rho$, which has an associated time scale $\tau_{\rho\mathbf{u}} = \ell^2/D_{\rho\mathbf{u}} = \rho\ell^2/\eta$. This can be compared to the time scale for mass diffusion $\tau_m = \ell^2/D_m$, with the ratio being the Schmidt number $\tau_m/\tau_{\rho\mathbf{u}} = Sc = \eta/\rho D_m$. In liquids Sc is large, even being of order 10^3 in water despite it being a fairly low viscosity liquid [39], leading to a separation of time scales for momentum and mass transfer. While this does not help argue $\partial_t(\rho\mathbf{u})$ is small enough to neglect, it does highlight that this term dies off so quickly it contributes almost nothing on the time scales of mass transfer.

The acceleration term can therefore be dropped to reach the Stokes equation

$$-\nabla p + \eta\nabla^2\mathbf{u} = -\mathbf{f}, \tag{2.7}$$

with the proviso that one is working on time scales of mass diffusion. The Stokes equation has two especially useful properties when compared to the full Navier-Stokes equation. First, it has lost all (explicit) reference to time, making it time-reversible. This leads to some neat demonstrations of 'mixing' and then un-mixing ink in viscous fluids [40], and through the scallop theorem presents challenges for swimming at microscopic scales as superbly described by Purcell [41].

Second is its linearity, allowing the flow in systems with generic force distri-

butions to be written as a sum of the Green's function, i.e. the solution when $\mathbf{f}$ is a point force:

$$-\nabla p + \eta \nabla^2 \mathbf{u} = -\mathbf{F}\,\delta(\mathbf{r}). \tag{2.8}$$

The case in an infinite fluid with the constraint $\mathbf{u} \to 0$ at $r = |\mathbf{r}| \to \infty$ is of particular importance both to this thesis and more widely in the field of soft matter. Various methods to find solutions for $p(\mathbf{r})$ and $\mathbf{u}(\mathbf{r})$ have been used [7, 42, 43], finding

$$p(\mathbf{r}) = \frac{\mathbf{F}\cdot\mathbf{r}}{4\pi r^3}, \tag{2.9}$$

$$\mathbf{u}(\mathbf{r}) = \frac{\mathbf{F}}{8\pi\eta r}\cdot\left(\mathbf{I} + \hat{\mathbf{r}}\otimes\hat{\mathbf{r}}\right), \tag{2.10}$$

with $\mathbf{I} = \mathrm{diag}[1,1,1]$ the 3×3 identity matrix. The flow field, often called a Stokeslet, can be broken down into the product of the point force and a mobility tensor $\boldsymbol{\mathcal{G}}(\mathbf{r})$. This is the Oseen tensor

$$\boldsymbol{\mathcal{G}}_{\mathrm{Oseen}}(\mathbf{r}) = \frac{1}{8\pi\eta r}\left(\mathbf{I} + \hat{\mathbf{r}}\otimes\hat{\mathbf{r}}\right), \tag{2.11}$$

around which the algorithm developed in Chapter 3 will be built. From here the linearity of the Stokes equation allows the flow field for any distribution of forces to be constructed using an appropriate sum of Stokeslets, as is standard for Green's functions.

In spite of the simplifications from Navier-Stokes it is not plain sailing from here. Part of the issue is that the r^{-1} decay in the Oseen tensor does not converge if summed over an infinite distance. Hydrodynamic interactions in low Reynolds number fluids are therefore long-ranged and all particles are aware of each other, however far away they might be. This often makes analytics intractable and computation times large.

There is a further caveat here: the asymptote at $r \to 0$ leads to an unphysical infinite response to forces for finite size particles. The simplest way to alleviate this issue is to set the mobility at $r = 0$ to the Stokes mobility $\mathbf{I}/6\pi\eta a$. However, doing this can leave the full $3N \times 3N$ mobility tensor in a system of N hydrodynamically interacting particles at risk of negative eigenvalues [44, 45], which would lead to equally unphysical negative diffusion coefficients. Avoiding both issues requires more care.

Seeking the leading order correction terms to Oseen leads to the Rotne-Prager-Yamakawa tensor [46, 47]

$$\boldsymbol{\mathcal{G}}_{\mathrm{RPY}}(\mathbf{r}) = \boldsymbol{\mathcal{G}}_{\mathrm{Oseen}}(\mathbf{r}) + \frac{1}{12\pi\eta r}\left(\frac{a}{r}\right)^2\left(\mathbf{I} - 3\hat{\mathbf{r}}\otimes\hat{\mathbf{r}}\right), \tag{2.12}$$

which is positive definite for $r \geqslant 2a$. This is often provided with a second branch

extending positive definiteness to $0 \leqslant r \leqslant 2a$ and limiting to the Stokes mobility at $r \to 0$ [46]:

$$\boldsymbol{\mathcal{G}}_{\mathrm{RPY}}(\mathbf{r}) = \frac{1}{6\pi\eta a}\left(\mathbf{I} - \frac{3r}{32a}\left(3\mathbf{I} - \hat{\mathbf{r}} \otimes \hat{\mathbf{r}}\right)\right). \tag{2.13}$$

There are various routes to obtaining Eq. (2.12), with the most insightful posing it as the first in a series of corrections to Oseen when applying Faxén's laws and taking account of successive reflections of the fluid flow between the two particles [43].

The RPY tensor is the standard in the polymer community for Brownian dynamics simulations, but it is not the only tensor used to replace Oseen and the Stokeslet it produces. A commonly used alternative is the 'regularised Stokeslet' [48, 49], which is the solution to Eq. (2.8) when the delta function is replaced by a smooth approximation. The details of the algorithm described in the next chapter will conveniently lead to a regularisation that is entirely separate from both of these but serves the same purpose.

2.2 The overdamped Langevin equation

For the simulation work in this thesis an explicit equation of motion is sought, rather than the statistical descriptions of, say, the Fokker-Planck and Smoluchowski equations [50]. Furthermore, for computational efficiency it is desirable to only follow the Brownian particles of interest and to ignore the solvent molecules.

Langevin equations provide such a description [43, 51]. In these a stochastic noise term $\boldsymbol{\xi}$ describes the fluctuations imparted by the local solvent molecules and is added to a deterministic equation of motion. Often the base equation of motion is Newton's second law, which in the simple case of a single particle experiencing a drag force $-\zeta\mathbf{v}$ and a net conservative force $\mathbf{F}$ leads to the Langevin equation [51]

$$m\frac{d\mathbf{v}}{dt} = -\zeta\mathbf{v} + \mathbf{F} + \boldsymbol{\xi}. \tag{2.14}$$

In the low Reynolds number limit the inertial term plays no part. Further useful changes come by recognising the friction coefficient ζ as the inverse of the mobility and moving to a system of N hydrodynamically coupled particles each labelled $i, j = 1, 2, ..., N$ and separated by $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$. Putting all this together leads to the overdamped Langevin equation [31]

$$\mathbf{v}_i = k_B T \sum_{j=1}^{N} \nabla \cdot \boldsymbol{\mathcal{G}}_{ij}(\mathbf{r}_{ij}) + \sum_{j=1}^{N} \boldsymbol{\mathcal{G}}_{ij}(\mathbf{r}_{ij}) \cdot \mathbf{F}_j + \boldsymbol{\xi}_i, \tag{2.15}$$

where the first sum comes from using the Itô formulation [52], used here due to it

being computationally more convenient than Stratonovich. While important to be aware of this sum, it will be disregarded in this thesis because only divergenceless mobility tensors are considered[1]. The second sum meanwhile follows from the flow field response to point forces[2].

Lastly, the noise is characterised by its first and second moments

$$\langle \boldsymbol{\xi}_i \rangle = 0, \tag{2.16}$$

$$\langle \boldsymbol{\xi}_i(t) \otimes \boldsymbol{\xi}_j(t') \rangle = 2k_B T \boldsymbol{\mathcal{G}}_{ij} \delta(t - t'), \tag{2.17}$$

determined by the fluctuation-dissipation theorem (FDT) [54, 55]. The noise being delta-correlated (or "white") in time comes from working on time scales greater than the time for real solvent molecules to impart a random kick.

Another important property and the motivation for using Itô is $\boldsymbol{\xi}$'s independence of the past *and* present system configuration, leading to

$$\left\langle \boldsymbol{\xi}_i(t) \cdot \left(\boldsymbol{\mathcal{G}}_{ij} \cdot \mathbf{F}_j \right)(t') \right\rangle = 0 \qquad \text{for } t \geqslant t', \ \forall j. \tag{2.18}$$

The same is not true for $t < t'$, which characterises how the system responds to past noise and will be central to Chapter 5.

In the context of numerical simulations where positions are updated by discrete steps $\delta\mathbf{r}$, it is helpful to also give the discrete version of Eq. (2.17) obtained by integrating over both t and t':

$$\langle \delta\mathbf{r}_i \otimes \delta\mathbf{r}_j \rangle = 2k_B T \boldsymbol{\mathcal{G}}_{ij} \delta t. \tag{2.19}$$

This result and the approach to generating the set of displacements to satisfy it will form the central core of this thesis.

2.2.1 Rotational Langevin equations

Particle orientations and rotations have so far been ignored. For systems of passive spheres (where orientation does not matter) or connected macro-particles (where rotation of the molecule as a whole comes from translations of individual sub-particles) this oversight causes little to no harm[3]. With single spherical particles where the orientation is important, such as the model micro-swimmers in Chapter 6, rotations cannot be neglected and the Langevin equation needs extending to account for this.

[1] While both the Oseen and RPY tensors are divergenceless, higher order tensors do not generally share this property [53].

[2] As presented only forces acting at particle centres are included. More generally the sum over j includes all forces on the fluid, and this will be used in Chapter 6.

[3] In the case of flexible polymers, each particle represents a coarse-grained section of chain and bond opening angles can change. If it were taken down to monomeric scales rotations would need treating with care.

As per standard rotational mechanics the particle orientation $\hat{\boldsymbol{\vartheta}}$ evolves with the angular velocity $\boldsymbol{\omega}$ through

$$\frac{d\hat{\boldsymbol{\vartheta}}}{dt} = \boldsymbol{\omega} \times \hat{\boldsymbol{\vartheta}}. \tag{2.20}$$

This shifts attention onto $\boldsymbol{\omega}$, the details for which follow very similarly to the translational velocity $\mathbf{v}$ to the point where they can be grouped into a single 6-dimensional vector with a single Langevin equation [56, 57]. They will however remain separated in this thesis, leading to two coupled Langevin equations for each particle [58]:

$$\mathbf{v}_i = \sum_{j=1}^{N} \boldsymbol{\mathcal{G}}_{ij}^{TT} \cdot \mathbf{F}_j + \sum_{j=1}^{N} \boldsymbol{\mathcal{G}}_{ij}^{TR} \cdot \boldsymbol{\Gamma}_j + \boldsymbol{\xi}_i \tag{2.21}$$

and

$$\boldsymbol{\omega}_i = \sum_{j=1}^{N} \boldsymbol{\mathcal{G}}_{ij}^{RT} \cdot \mathbf{F}_j + \sum_{j=1}^{N} \boldsymbol{\mathcal{G}}_{ij}^{RR} \cdot \boldsymbol{\Gamma}_j + \boldsymbol{\Xi}_i. \tag{2.22}$$

The new quantities being: $\boldsymbol{\Gamma}_j$ the torque on particle j; $\boldsymbol{\Xi}_i$ the rotational noise; and the cross-mobility tensors $\boldsymbol{\mathcal{G}}_{ij}^{\mu\nu}$ with $\mu, \nu = T(\text{ranslation}), R(\text{otation})$ coupling μ-type motion to ν-type forces. Note $\boldsymbol{\mathcal{G}}_{ij} = \boldsymbol{\mathcal{G}}_{ij}^{TT}$ is used for brevity in sections of this thesis where rotations do not appear.

The FDT again finds the correlations between noise terms, with the rotation-dependent results [55]

$$\left\langle \boldsymbol{\Xi}_i(t) \otimes \boldsymbol{\xi}_j(t') \right\rangle = 2k_B T \, \boldsymbol{\mathcal{G}}_{ij}^{RT} \, \delta(t - t'), \tag{2.23}$$

$$\left\langle \boldsymbol{\Xi}_i(t) \otimes \boldsymbol{\Xi}_j(t') \right\rangle = 2k_B T \, \boldsymbol{\mathcal{G}}_{ij}^{RR} \, \delta(t - t'), \tag{2.24}$$

while the translation-rotation equation has been omitted because it can be readily obtained by taking the transpose of Eq. (2.23), which then finds the relation

$$\boldsymbol{\mathcal{G}}_{ij}^{TR} = \left(\boldsymbol{\mathcal{G}}_{ji}^{RT}\right)^{\mathrm{T}}, \tag{2.25}$$

often derived from the Lorentz reciprocal theorem [59].

Lastly, it is worth listing the rotational mobility tensors for spherical particles. Up to the same level of approximation as the RPY tensor they are [57, 58]

$$\boldsymbol{\mathcal{G}}_{ij}^{RT} = \delta_{ij} \, \mathbf{0} - (1 - \delta_{ij}) \frac{[\hat{\mathbf{r}}_{ij}]_\times}{8\pi\eta r_{ij}^2}, \tag{2.26}$$

$$\boldsymbol{\mathcal{G}}_{ij}^{RR} = \frac{\delta_{ij}}{8\pi\eta a_i^3}\mathbf{I} + \frac{1 - \delta_{ij}}{16\pi\eta r_{ij}^3}(3\hat{\mathbf{r}}_{ij} \otimes \hat{\mathbf{r}}_{ij} - \mathbf{I}), \tag{2.27}$$

where $[\hat{\mathbf{r}}]_\times$ is a skew-symmetric tensor with the form $\varepsilon_{abc}\hat{r}_b$ in index notation.

2.3 Diffusion and velocity autocorrelations

Many of the results in the final chapters will revolve around non-trivial diffusion. It is therefore helpful to briefly run through the relevant details here. Starting at the beginning, the best known expression for self-diffusion is that provided by Einstein [60]:

$$D = \lim_{t\to\infty} \frac{\left\langle (\mathbf{r}(t) - \mathbf{r}(0))^2 \right\rangle}{6t}. \tag{2.28}$$

A more powerful expression is obtained by writing the displacement in terms of velocity as

$$\mathbf{r}(t) - \mathbf{r}(0) = \int_0^t dt' \mathbf{v}(t'), \tag{2.29}$$

with which Eq. (2.28) can be re-expressed as

$$\begin{aligned} D &= \lim_{t\to\infty} \frac{1}{6t} \left\langle \int_0^t \int_0^t dt' dt'' \mathbf{v}(t') \cdot \mathbf{v}(t'') \right\rangle \\ &= \lim_{t\to\infty} \frac{1}{3t} \int_0^t d\tau (t-\tau) \left\langle \mathbf{v}(\tau) \cdot \mathbf{v}(0) \right\rangle \qquad (2.30) \\ &= \frac{1}{3} \int_0^\infty d\tau \ \left\langle \mathbf{v}(\tau) \cdot \mathbf{v}(0) \right\rangle, \qquad (2.31) \end{aligned}$$

reaching the Green-Kubo relation for diffusivity [61, 62].

The limit $t \to \infty$ in the above relations should be taken with a pinch of salt, both because no real measurement can truly reach this limit and because any finite system imposes bounds on $\mathbf{r}(t) - \mathbf{r}(0)$ such that the definition in Eq. (2.28) would always drop to 0 [63]. In practice the upper limit is set by the time over which systems relax and $\langle \mathbf{v}(t) \cdot \mathbf{v}(0) \rangle$ decays to zero, beyond which the integral contributes nothing (at least until the finiteness of the system intervenes). Often this means the long-time diffusivity is readily accessible, although a counterexample with very slow relaxation will be explored in Chapter 5.

The long-time limit is not the only one of interest however. In systems described by an overdamped Langevin equation with white noise the limit $t \to 0$ already has a finite diffusivity. Since causal forces take finite time to displace particles this limit isolates the diffusivity provided by the thermal fluctuations. Namely

$$D_{k_BT} = \frac{1}{3} \int_0^\infty dt \, 2k_BT \, \mathrm{Tr}[\boldsymbol{\mathcal{G}}_{ii}] \delta(t) = \frac{1}{3} k_BT \, \mathrm{Tr}[\boldsymbol{\mathcal{G}}_{ii}], \tag{2.32}$$

yielding the Einstein relation [64]. By the same token, if D_{k_BT} can be isolated so can the non-thermal contributions. Chapters 5 and 6 will each present very different systems where such a separation is of great interest.

To round off this section it is helpful to spend some time on the details of the self-mobility $\boldsymbol{\mathcal{G}}_{ii}$ and by extension the self-diffusion. For a sphere this is well known to come from Stokes' law: $\mathbf{F} = 6\pi\eta a\mathbf{v}$ [6]. The drag tensor $6\pi\eta a\mathbf{I}$ is the inverse of the mobility so

$$\boldsymbol{\mathcal{G}}_{ii} = \frac{1}{6\pi\eta a}\mathbf{I}, \qquad \Rightarrow D_{k_BT} = \frac{k_BT}{6\pi\eta a}. \tag{2.33}$$

It will be useful on multiple occasions to be aware of how this changes in periodic systems, commonly used to represent bulk fluid in simulations. Because of the long-range hydrodynamics particles interact with their own periodic images, affecting their self-mobility. In particular, for a cubic box of side length L the self-mobility decreases to [65]

$$\boldsymbol{\mathcal{G}}_{ii} = \frac{1}{6\pi\eta a}\left(1 - 2.837\frac{a}{L}\right)\mathbf{I} \tag{2.34}$$

with a corresponding change to D_{k_BT}.

2.4 Polymer models

While Wavelet Monte Carlo dynamics will be applicable to a wide range of systems, polymer physics has a long and established history to benchmark against [8, 9], and is challenging enough to simulate for new, more efficient algorithms to be of great value in extending our understanding. With this in mind, polymer solutions will be used as test cases as WMCD is developed, for which this section presents the basics of polymer physics and the models used for linear (non-branching) polymers.

2.4.1 Rouse and Zimm models

Rather than simulating polymers at the scales of individual atoms, a coarse grained model is used to represent the polymer as a chain of N_b beads as shown in Fig. 2.1. Each bead contains many monomers but does not adopt all of their detailed properties, a simplification allowed by the universality of polymer behaviour making the large scale properties insensitive to atomic level features such as bond angles [8]. Furthermore, the bonds in the coarse grained chain are considered perfectly flexible, i.e. the angle between two consecutive bonds is uncorrelated before interactions between beads are taken into account, so that the average bond length corresponds to (at least) the Kuhn length [8], itself closely related to the real chain's persistence length.

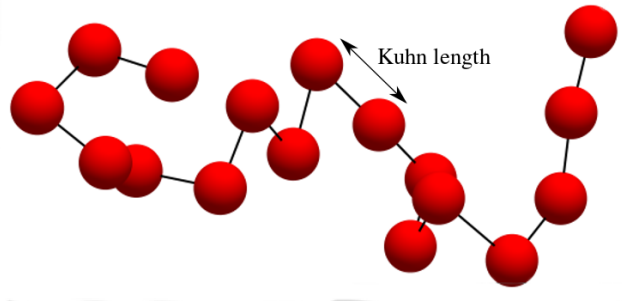

Figure 2.1: Example linear polymer represented by a flexible, coarse grained chain of beads connected by bonds with typical length equal to the Kuhn length. Note this is a 2-dimensional projection of a 3-dimensional chain.

All that is left is to model how the beads interact with each other, both hydrodynamically and through conservative forces. Without hydrodynamics one has the Rouse model [8, 66]. Omitting hydrodynamic interactions (HIs) makes analytic calculations much easier and does not affect the static properties which derive from equilibrium distributions of configurations. However, HIs are important for dynamic properties, as will be seen in the case of polymer diffusion in Chapter 5.

The hydrodynamically coupled alternative is the Zimm model [8, 67], where every pair of beads interacts via the Oseen tensor (or often RPY). This comes at a cost to analytic calculations, which are made intractable in all but the simplest systems. The preaveraged approximation, which takes the statistical average mobility tensor between a pair of particles based on the arc length between them [8, 67, 68], is a useful if imperfect approximation commonly used to make some headway here. Meanwhile it is the HIs that present the greatest challenge computationally, as will be described in Section 2.5.

The full Zimm versus Rouse story is significantly more rich than simply whether HIs are included or not due to hydrodynamic screening, whereby constraints on chain motion can lead to Rouse-like dynamics on large enough space and time scales [69]. As polymer concentration increases from the ultra-dilute regime, where each chain can be considered isolated from each other, screening sets in when the chains begin to overlap and restrict each other's motion [70]. This regime, where chains overlap but the fluid is still predominantly solvent, is called 'semi-dilute'. Entanglement effects are then a further complication [8], but these are not considered here.

2.4.2 Solvent quality and the Flory exponent

With regards to the conservative forces present, one of the most important parts of the polymer model, be it Rouse or Zimm, is the quality of the solvent. Physically this

describes how well the polymer dissolves in the solvent, based on each monomer's energetic preference to be in contact with the solvent or other monomers [4], and influences the chain configurations. Solvent quality is a continuous variable, but it is helpful to focus on the extremal cases and the way they are modelled in simulations.

Poor (or bad) solvent: First is the poor solvent, in which monomers favour contact with each other over the solvent, and hence the polymer collapses into 'globules' (or clusters if multiple chains are nearby) [71, 72]. In a simulation it is simplest to use an attractive inter-bead potential with short-ranged repulsion to account for the excluded volume (EV) due to space already being occupied.

Good solvent: At the other end are good solvents, in which monomers seek to maximise their contact area with the solvent. Rather than collapsing, the polymer swells into a self-avoiding walk configuration, earning the name 'swollen chain'. Because it is only contact that is unfavourable, beads do not need a long ranged repulsion and local EV interactions are sufficient to model chains in good solvent.

Work in this thesis adopts Grest and Kremer's model for chains in good solvent [73], using the Weeks-Chandler-Anderson potential [74]

$$U_{\mathrm{WCA}} = 4\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} + \frac{1}{4} \right) \Theta(2^{1/6}\sigma - r_{ij}) \tag{2.35}$$

for the EV interactions. This is simply the Lennard-Jones potential with the attractive part removed by the Heaviside step function. σ sets the range of repulsion while ϵ sets the energy scale (typically of order $k_B T$).

Meanwhile neighbouring beads are held together by finitely extensible nonlinear elastic (FENE) springs, with potential

$$U_{\mathrm{FENE}} = -\frac{1}{2} k_{\mathrm{FENE}} R_0^2 \ln\left(1 - (r_{ij}/R_0)^2\right) \tag{2.36}$$

confining bond lengths to be shorter than R_0. At $(r_{ij}/R_0)^2 \ll 1$ this is quadratic and looks like a Hookian spring with spring constant k_{FENE}.

θ solvent: Between poor and good solvents there lies a crossover between attractive and repulsive behaviour. A solvent where the two perfectly balance, i.e. when monomers have equal contact energies with both the solvent and other monomers, is called a θ solvent[4] . Under these conditions the chain configurations are ideal

[4]Since experimentally this is achieved by changing the temperature, the θ temperature T_θ is often referred to. T_θ depends on both the polymer and solvent chemistry and does not factor into simulations which are usually set up to be independent of the chemistry.

random walks so can be represented in simulations by turning off inter-bead interactions[5]. They are also referred to as Gaussian chains as the separation of any two beads is given by a Gaussian in arc length between them [8].

In simulations, Gaussian distributed separations are achieved with a quadratic bond potential

$$U_{\text{quad}} = \frac{1}{2} k_{\text{quad}} \left(\mathbf{r}_{i+1} - \mathbf{r}_i \right)^2 . \tag{2.37}$$

Flory exponent: The parameters in the potentials above all affect polymer properties, but perhaps most important is that they are fractal objects with self-similarity on length scales between that of individual beads and the whole chain [75]. The fractal dimension d_f is easily seen in the chain's static structure factor [5]. However, often more convenient than d_f is its reciprocal, called the Flory exponent $\nu = 1/d_f$. ν is defined by how the size of the chain scales with the number of beads, $R \sim N_b^\nu$, and takes the values [4, 76]

$$\nu = \begin{cases} 1/3 & \text{poor} \\ 0.588 \approx 3/5 & \text{good} \\ 1/2 & \theta \end{cases} . \tag{2.38}$$

Many properties of polymer systems, including diffusivity, relaxation times and the polymer contribution to viscosity, exhibit scaling with ν [8, 70]. Several of these will reappear in later chapters.

2.5 Review of Brownian dynamics

While the physics presented so far underpins all that is to come, this work will be as focussed on the development of a computational method as it is on using it to learn about physics. To that end it is worth having some idea of the landscape in which the algorithm is to be placed, especially regarding the more closely related implicit solvent methods.

Hence Brownian dynamics (BD) is the primary point of comparison and is relatively simple to describe after the previous sections. As first set out in a hydrodynamically coupled form by Ermak and McCammon in 1978 [31], BD is a numerical solver of a discrete-time overdamped Langevin equation of the form in Eq. (2.15). Written out in full as per Ermak, with a slightly more modern way to

[5]Real chains must of course still have some EV interactions. However, the beads in the coarse grained models should not be interpreted as solid objects in the same way as monomers, and allowing some bead overlap is not unphysical.

write the noise term [77], this means updating particle positions with

$$\mathbf{r}_i(t+\delta t) - \mathbf{r}_i(t) = k_B T \delta t \sum_j \nabla_{\mathbf{r}_j} \cdot \boldsymbol{\mathcal{G}}_{ij} + \delta t \sum_j \boldsymbol{\mathcal{G}}_{ij} \cdot \mathbf{F}_j + \sqrt{2\delta t} \sum_j \boldsymbol{\mathcal{B}}_{ij} \cdot \mathbf{W}_j. \quad (2.39)$$

As before the first term on the right hand side is zero by considering divergenceless mobility tensors, typically Rotne-Prager-Yamakawa. Of the remaining terms the middle, causal term is readily calculated but the rightmost sum over noise is not. The splitting of the noise $\boldsymbol{\xi}_i = \sqrt{2\delta t}\boldsymbol{\mathcal{B}}_{ij} \cdot \mathbf{W}_j$ into an uncorrelated random Gaussian noise $\mathbf{W}$, satisfying

$$\langle \mathbf{W}_i \rangle = 0, \quad (2.40)$$

$$\langle \mathbf{W}_i \otimes \mathbf{W}_j \rangle = \delta_{ij}\mathbf{I}, \quad (2.41)$$

and a causal tensor $\boldsymbol{\mathcal{B}}$ satisfying

$$\boldsymbol{\mathcal{B}}_{ik} \cdot \boldsymbol{\mathcal{B}}^{\mathrm{T}}_{kj} = k_B T \boldsymbol{\mathcal{G}}_{ij} \quad (2.42)$$

goes some way to revealing where the difficulties lie. Generating the set of $\mathbf{W}_i$ is a quick and efficient process whose complexity scales as N. Meanwhile, $\boldsymbol{\mathcal{B}}$ can be found from the square root of $\boldsymbol{\mathcal{G}}$ with standard algorithms, but since this requires working with the full $3N \times 3N$ mobility tensor it is a stretch to call them efficient. The Cholesky factorisation used by Ermak and and other workers since has computational complexity of order N^3 for example [31, 78]. Even without the overheads of simulating the solvent, a run time increasing this fast will only beat the explicit-solvent alternatives in small systems.

In 1986 Fixman proposed a decomposition scheme using a truncated series of Chebyshev polynomials reducing the complexity to $N^{2.25}$ [37]. A detail of this algorithm that will crop up later is the need to estimate the largest and smallest eigenvalues of $\boldsymbol{\mathcal{G}}$, which contributes the factor of $N^{0.25}$. Even with the improvement to the cost, Fixman's algorithm still scales too quickly for large scale simulations to be practical, prompting the development of the explicit-solvent methods in the 1990s. BD had not been abandoned however, and further developments continued again in the new century. The rest of this section quickly recaps these develoments, referring to the methods above by Cholesky and Chebyshev.

2.5.1 Unbounded systems

The truncated expansion ansatz, 2009

In Ref. [32] Geyer and Winter argued that all forces in BD are approximate to begin with and it is therefore not necessary to implement hydrodynamically coupled noise

forces exactly as per the Rotne-Prager tensor. This led them to the 'truncated expansion ansatz' (TEA), in which random and deterministic forces are treated on equal footing. In doing so they choose to make the algorithm more efficient by assuming the off-diagonal elements of $\boldsymbol{\mathcal{G}}$ are all equal and small compared to the diagonal elements, allowing the truncation of a Taylor series to lowest order in D_{ij}/D_{ii}.

The argument that all BD algorithms are approximations is certainly true, but the added approximation it makes regarding the hydrodynamics is a big one. Given how important the differences are between the Zimm and Rouse models of polymers (with and without HIs respectively) it is not surprising that taking a middle line leads to significant errors in measured dynamical quantities. These errors can reach about 10% in systems where $D_{ij} \approx D_{ii}$ (e.g. with particles in close proximity).

The upshot is an algorithm with small cost prefactor and cost scaling as N^2 for both random and deterministic contributions, and since its introduction in 2009 the TEA algorithm has been a common benchmark for later approaches [79]. That said, while TEA is computationally efficient when compared to standard Cholesky and Chebyshev methods, complexity scaling as N^2 still leaves it expensive for large systems.

Average mobility, 2017

The method proposed by Miao *et al.* takes a qualitatively similar approach to the TEA in simplifying the mobility matrix rather than decomposing it more efficiently [36]. In it they replace the decomposition process with an iterative procedure to find an average mobility matrix to use. Again this leads to complexity scaling as N^2, with about an order of magnitude improvement over TEA on the prefactor.

The method is currently limited to systems with steady state flows, but the bigger limitation is in the use of an average mobility which will limit its fidelity. In this way it is akin to the preaverage approximation for polymers [8]. The neglect of fluctuating HIs is known to change results and restricts its applicability to real systems.

Krylov subspaces, 2012

Ando *et al.* were the first to use Krylov subspaces in the decomposition of $\boldsymbol{\mathcal{G}}$ as an alternative to the Chebyshev approach [33]. Again it is not exact like Cholesky is, but it has the advantage over Chebyshev that it does not require an estimate of the extremal eigenvalues, which reduces the cost scaling from $N^{2.25}$ to N^2 (albeit with a slight dependence on the parameters chosen in the algorithm).

Saadat and Khomami compared the Cholesky, Chebyshev and Krylov approaches [80], finding the prefactor of each to be comparable such that the costs

of the latter two are similar over a wide range of system sizes. Consequently the current state of high fidelity non-periodic BD involves a choice between these three algorithms, with no clear winner.

2.5.2 Periodic systems

So far in this section, and indeed much of this chapter, the difference between periodic and unbounded systems has been implicit if not outright ignored. Largely this is because the form of the Langevin equation does not change: the mobility tensor is just replaced with one appropriate for periodic systems. Essentially this just leads to a sum of periodically placed images of the flow field. Simple enough conceptually but here again the long-range nature of the HIs bites and it takes some finesse to make the sum converge.

To handle this, Beenakker made use of an Ewald sum[6] to sum the contributions from the long-range interactions [82]. This sum is split between position and Fourier (reciprocal) space, with algorithms usually being set up for the complexity of one sum to scale as N and the other as N^2. Furthermore the Ewald sum can also carry the cost of the decomposition of $\boldsymbol{\mathcal{G}}$ [34], so the total cost in periodic BD then scales as N^2 with a slight improvement over Chebyshev.

With the fundamental difference to how the noise term is calculated, developments in periodic BD have taken a separate path to their unbounded counterparts, with the key developments now listed.

Particle mesh Ewald, 2003

Although not technically BD, Banchio and Brady's analogous work on a Stoksian dynamics algorithm (akin to BD but with a second order Langevin equation) can be presented as such because they turned their attention to Brownian systems and targeted the Ewald sum for improvement [83]. To improve the efficiency they used approximate 'particle mesh Ewald techniques' to reduce the complexity to $N^{1.25} \ln N$.

The downside is the very large prefactor, making it only worthwhile for large systems even when compared to a Cholesky (N^3) Stokesian dynamics algorithm. The work did also propose a much faster mean-field method with the same scaling, but that would share the same issues with fidelity as the average mobility approach above.

Optimising the real and Fourier space cost in Ewald, 2012

Following similar work in electrostatics [84], Jain *et al.* introduced an optimisation parameter into the Ewald sum to allow the distribution of computational effort in

[6] These were originally used in electrostatics where potentials also drop off as $1/r$ [81].

the real and Fourier space parts to be equal and scale as $N^{1.5}$ [34]. Otherwise they used an unmodified Chebyshev method in which the cost of eigenvalue estimation raised the complexity to $N^{1.8}$. Furthermore they modified Beenakker's Ewald sum to include the short-range branch of the RPY tensor, thus allowing valid simulation of overlapping Brownian particles.

In spite of the improvements the algorithm was several orders of magnitude more expensive than a hybrid Lattice Boltzmann + Molecular dynamics algorithm for fairly dense systems [34]. Certainly there is some relative gain against explicit solvent methods in more dilute systems, but not enough for serious large-scale semi-dilute simulations.

Sparse arrays, 2015

Saadat and Khomami recently combined many of the techniques above, including the use of Krylov subspaces, Jain *et al.*'s optimisation of the Ewald sum, and a smooth particle mesh Ewald (SPME) sum [35]. Further still, they only included interactions with nearby particles in the real-space part of the Ewald sum, allowing it to be stored as a sparse array and, with SPME handling the reciprocal space part, end up with an entirely matrix-free calculation. Altogether this leads to a complexity of order $N \ln N$.

While this is a significant improvement, the number of approximations needed to reach this should be kept track of, as should the relative cost prefactors. With comparisons to established methods yet to be made there remains a question mark over this. Nevertheless this work may finally make BD competitive in periodic systems.

2.5.3 Final remarks on Brownian dynamics

The state of BD is split by whether the simulated system is periodic or not. Non-periodic systems have not seen major improvements to efficiency since Fixman's use of Chebyshev polynomials, with the complexity reduced only to N^2 with Krylov subspace methods that provide another option but do little for the prospects of simulating large systems. The cost prefactors have been reduced significantly with approaches that approximate the mobility matrix itself, but not without a hit to the accuracy of data. Nevertheless, in infinitely dilute systems the lack of solvent molecules still leaves BD more competitive than explicit-solvent methods [38]. It will be seen that in these systems WMCD will stand as a major improvement on them both.

In periodic systems the Ewald sum has been the main focus and has allowed the complexity to scale better. The most recent developments with sparse arrays may be a game changer for BD here, although that remains an open question due

to the limited comparison to other algorithms currently available. Existing comparisons with explicit-solvent methods are very unfavourable for periodic BD, and again WMCD will be seen to trump BD here.

It is notable that many of the developments with BD have been very recent, with the only one listed not from the last decade pertaining to Stokesian dynamics rather than BD, so future improvements are quite likely. Nor has this account covered the full scope of BD. In confined systems for example, BD has seen $N \ln N$ scaling for some time [85].

2.6 Monte Carlo methods

Another important class of algorithms are the Monte Carlo (MC) algorithms, or more formally Markov chain Monte Carlo where the update algorithm at each time step depends only on the system at that time, with no knowledge of the past [86]. These algorithms take many different forms with differing levels of physical meaning, with the primary distinction being between dynamic and static MC algorithms as will be discussed soon. Dynamical or not, all MC algorithms share the same basic structure:

> Pick a 'move' with parameters taken from a some distribution $\mathcal{P}$.
>
> Calculate the total energy change ΔU in this move.
>
> Accept or reject the move with probability $P_{\text{acc}}(\Delta U/k_B T)$.
> - If accepted, keep the updated configuration.
> - If rejected, revert to the previous configuration.
>
> Repeat steps above for the next move.

There is a lot of freedom within this. What constitutes a move is surprisingly unimportant as far as end results are concerned, especially for static MC, although a wise choice can make a world of difference to computational complexity as will be seen in WMCD. First and foremost though, the origin and details of the acceptance test need describing.

2.6.1 Detailed balance and equilibrium

For MC algorithms to give time reversible equilibrium results, they must satisfy detailed balance [86]. This sets the condition

$$\pi_i \mathcal{T}_{i\to j} = \pi_j \mathcal{T}_{j\to i}, \tag{2.43}$$

where $\pi_i \propto \exp(-\beta U_i)$ is the probability of being in state i and $\mathcal{T}_{i\to j}$ is the transition rate from state i to state j. The transition rate can be expressed as the probability of choosing a move between states i and j multiplied by the acceptance probability of the move. I.e.

$$\mathcal{T}_{i\to j} = \mathcal{P}(i \to j) P_{\text{acc}}(i \to j). \tag{2.44}$$

It is common for the distribution of attempted moves to be symmetric and independent of the configuration, in which case $\mathcal{P}(i \to j) = \mathcal{P}(j \to i)$ and Eq. (2.43) gives

$$\frac{P_{\text{acc}}(i \to j)}{P_{\text{acc}}(j \to i)} = \frac{\pi_j}{\pi_i} = e^{-\beta(U_j - U_i)}, \tag{2.45}$$

which still leaves some freedom on how to choose P_{acc}. The most widely used form of P_{acc} is attributed to Metropolis [87], and is given by

$$P^M_{\text{acc}}(\beta\Delta U) = \min\left(1, e^{-\beta\Delta U}\right) = e^{-\frac{1}{2}\beta\Delta U} e^{-\frac{1}{2}\beta|\Delta U|}. \tag{2.46}$$

The advantage of the Metropolis acceptance probability is that it maximises P_{acc} for a given ΔU, and hence minimises the number of attempted moves (corresponding to computational time spent) between updating the configuration.

However, P^M_{acc} not being smooth at $\beta\Delta U = 0$ can make it difficult to work with in analytical calculations and it is not optimal for accurate dynamical results (this will be discussed in Section 2.6.4). An alternative, more symmetric choice comes from Glauber [88]:

$$P^G_{\text{acc}}(\beta\Delta U) = \frac{1}{2}\left(1 - \tanh\left(\frac{1}{2}\beta\Delta U\right)\right) = \left(1 + e^{\beta\Delta U}\right)^{-1}. \tag{2.47}$$

2.6.2 Static and dynamic MC

While MC simulations ensure the correct equilibrium distribution of states and provide routes between them, they do not generally say anything about the system dynamics. This is evident from the calculations above being completely independent of both time and what the states i and j actually are. These states could be as similar as a slight displacement of a single particle or as distant as a complete reshuffle of the entire system, yet the acceptance test only sees the energies in each state and the likelihood of attempting to move between them, regardless of whether the move has any physical interpretation.

In fact, because the equilibrium distribution of states is governed by energies alone and therefore unaffected by hydrodynamics, static properties can be obtained with any efficient exploration of equilibrium space. This is the premise of static MC, which forsakes expensive hydrodynamic calculations in favour of moves designed to explore equilibrium space as efficiently as possible. Polymer systems alone

have spawned a range of MC move types ranging from displacements of individual monomers [89] to larger scale bond rotations [90, 91, 92], as well as so-called bridging moves used to exchange neighbours in branching polymers [93, 94] and "event-chain" algorithms to choose a sequence of moving particles to avoid particle overlap in systems of hard spheres [95, 96].

Nevertheless it is possible to have a MC algorithm whose dynamics map onto the dynamics of the Langevin equation and in turn match the behaviour of BD [97, 98, 99]. To do this one needs to constrain the simulation to moves that are physically possible over small time increments and to have a handle on what that time increment is. Simulations that meet these requirements are called dynamic MC (DMC) algorithms, or alternatively kinetic MC.

To date, DMC algorithms applied to systems relevant to this thesis have omitted HIs [100, 101, 102], and there is consequently little to gain from a full discussion on how they choose their moves. The designation of time, however, is essentially the same as will be used later, and boils down to using Eq. (2.19) to translate particle displacements into the time step. Subtleties arise through the acceptance test whereby the actual dynamics do not quite match those in BD when the time increment is calculated assuming zero move rejections. The error, whose origin will be sketched out in Sec. 2.6.4, can be reduced by rescaling the time step by the acceptance probability [102], and ideally by maintaining a low rejection rate.

It should be noted that while this approach works when the DMC algorithm is constructed to supply Langevin dynamics, it is not appropriate for all DMC algorithms. For example, in systems where there is a probability that nothing changes over a time interval and events are Poissonian, drawing time increments from a distribution is correct if each move is guaranteed to update the system. This is done in the 'n-fold way' (or Bortz-Kalos-Lebowitz) algorithm [103].

2.6.3 Smart MC

So far only unbiased move selection leading to Eq. (2.45) has been considered. Bringing $\mathcal{P}$ back into the discussion, one can bias $\mathcal{P}$ to impose forces instead of leaving that to the acceptance test. The route followed in this thesis is that of 'smart Monte Carlo' (SMC) [104]. In SMC moves are chosen from a Gaussian distribution centred on the force. In an overdamped Langevin context, conveniently where SMC was first introduced, the mean displacement in the absence of HIs is $\beta D \delta t \, \mathbf{F}$ so the SMC distribution is

$$\mathcal{P}(\delta\mathbf{R}_{i\to j}; \mathbf{F}_i) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{(\delta\mathbf{R}_{i\to j} - \beta D \delta t \, \mathbf{F}_i)^2}{2\sigma^2}\right). \tag{2.48}$$

The move variance can be re-expressed as $\sigma = 2D\delta t$ using Eq. (2.19).

Having modified the move selection in this way, detailed balance requires the acceptance test to be modified too. Expanding the exponent in Eq. (2.48) and including $\mathcal{P}(j \to i)/\mathcal{P}(i \to j)$ in Eq. (2.45) gives

$$\frac{P_{\rm acc}(i \to j)}{P_{\rm acc}(j \to i)} = \exp\left(-\beta\Delta U - \frac{\beta D\delta t}{2\sigma^2}\left[2(\mathbf{F}_j + \mathbf{F}_i)\cdot\delta\mathbf{R}_{i\to j} + \beta D\delta t(\mathbf{F}_j^2 - \mathbf{F}_i^2)\right]\right) \tag{2.49}$$

when using $\delta\mathbf{R}_{j\to i} = -\delta\mathbf{R}_{i\to j}$. After this the Metropolis approach is again taken and $P_{\rm acc}$ is simply the smallest of 1 and the exponential just found.

2.6.4 Accuracy of MC tests

The acceptance test leads to a meaningful choice between MC algorithms and direct implementations of the Langevin equation. The noise term in the latter is set up to supply a chosen variance and in turn particle diffusivity, however without any move rejection this will not quite obey detailed balance. MC methods have the reverse problem, being set up to guarantee equilibrium but at the price of the move rejection affecting the dynamics. This section presents calculations for the estimated errors in particle diffusion when using the 3 acceptance tests introduced above.

Metropolis: To estimate the size of the error in the Metropolis, consider the observed diffusivity when accounting for move rejections. Actual displacements can be written as the product of the attempted displacement $\delta\mathbf{r}_0$ and a Heaviside step function $\Theta(P_{\rm acc} - \texttt{rand})$. The observed diffusivity is then

$$\left\langle \delta\mathbf{r}_0^2\,\Theta\right\rangle - \left\langle \delta\mathbf{r}_0\,\Theta\right\rangle^2 = 6D_{\rm Met}\delta t. \tag{2.50}$$

The mean of the square can separate out the $\delta\mathbf{r}_0$ and Θ factors to reasonable approximation because $\delta\mathbf{r}_0^2$ is the same going both up and down a potential gradient, while $P_{\rm acc}$ certainly is not. This term is then $\langle\delta\mathbf{r}_0^2\rangle\langle\Theta\rangle = \langle\delta\mathbf{r}_0^2\rangle P_{\rm acc}$. The same cannot be done on the second term because it is move rejection that gives biased motion in the first place. However the dominant corrections are expected to be of the same order of magnitude.

If the variance in the attempted moves is chosen to provide diffusivity D, then this implies

$$\begin{aligned} D_{\rm Met} &\approx DP_{\rm acc} \\ &= De^{-\frac{1}{2}\beta\Delta U}e^{-\frac{1}{2}\beta|\Delta U|} \\ &\approx D\left(1 - \frac{1}{2}\beta(\Delta U + |\Delta U|)\right). \end{aligned} \tag{2.51}$$

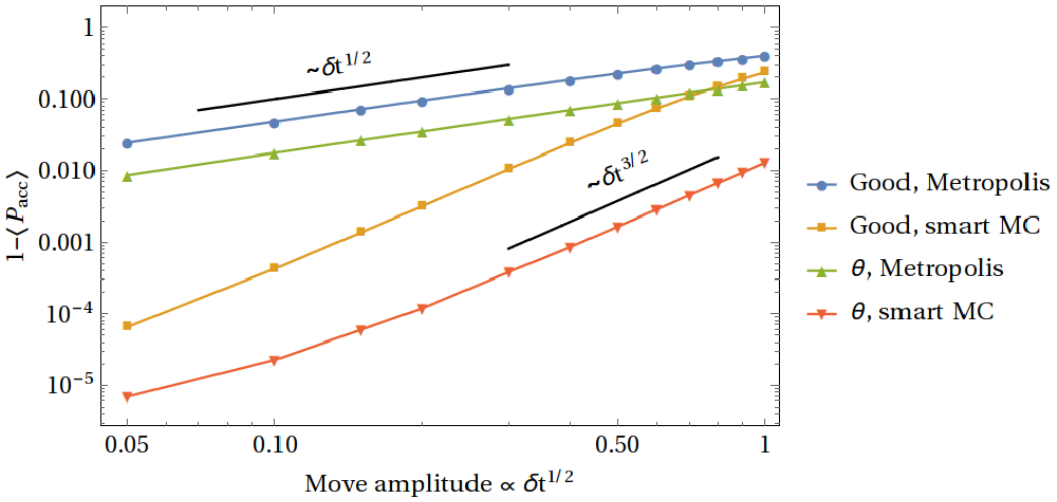

Figure 2.2: Average rejection probabilities for Metropolis and smart Monte Carlo moves with different time steps, in systems of single polymer chains in both θ and good solvent. The moves used follow the WMCD algorithm detailed in Chapters 3 and 4.

Using

$$\Delta U \approx -F\delta r \sim \delta t^{1/2} \tag{2.52}$$

finally gives a mobility error of order $\delta t^{1/2}$ for the Metropolis algorithm.

Smart MC: The argument for SMC is essentially the same as for Metropolis except the expression for P_{acc} in Eq. (2.51), in which $-\beta\Delta U$ is replaced by the exponent in Eq. (2.49). The full analysis finding the lowest power of δt is long and adds nothing so is omitted, but in the end one finds the leading error in SMC to be of order $\delta t^{3/2}$, making smart MC significantly more accurate for a given (small) time step.

Glauber: The Glauber acceptance test is more delicate because P_{acc} contains no even powers of ΔU, so if the same analysis was used as for the Metropolis test one finds any probability lost for $\delta\mathbf{r}$ is exactly compensated for by the reverse move (at least for sufficiently small moves). The mean square displacement is therefore the same as it would be in the absence of forces, leading to

$$\left\langle \delta\mathbf{r}_0^2\,\Theta \right\rangle = 2D\delta t, \tag{2.53}$$

without the mean term. The full variance then has

$$\begin{aligned} 2\delta t D_{\text{Glauber}} &= 2\delta t D - \langle \delta \mathbf{r}_0 \, \Theta \rangle^2 \\ &= 2\delta t D \left(1 - \frac{(\mu \delta t F)^2}{2\delta t D} \right) \end{aligned} \tag{2.54}$$

so that the leading correction term goes as δt, between that of Metropolis and smart MC.

The scaling of the leading errors in the Metropolis and SMC algorithms is confirmed in Fig. 2.2 using data from the WMCD algorithm (Glauber is omitted because its equivalent to P_{acc} in Eq. (2.54) is much more difficult to obtain). In practice algorithmic simplicity and speed also factor into the decision making process, and all three can be the best choice under the right circumstances. For instance the extra layer of complexity in SMC means it will be left out of WMCD until Chapter 4.

2.7 Wavelets

A major feature that separates the work in later chapters from the algorithms described above is the use of wavelets. Since their introduction in 1984[7] [106], wavelets and their associated linear transforms have found widespread use in physics and data analysis, with examples as varied as the analysis of sunspot cycles [107], fingerprint storage and retrieval [108], motion amplification in videos [109], identifying coherent structures in turbulence [110], and simulations of fish swimming in each other's wake [111].

The details of wavelets vary from use to use and the following account focusses solely on the parts needed in later chapters, namely continuous wavelet transforms (CWT) [106, 112]. Notable omissions include discrete wavelet transforms [113, 114] and second generation wavelets [115], although they are central to some of the examples given above.

2.7.1 1-dimensional wavelets

The starting point for any discussion on wavelets is to establish what a wavelet actually is, abstracted from how they are used. Although the maths is largely unchanged in higher dimensions, 1-dimensional wavelets are by far the easiest to build an intuition with.

3 examples of commonly used 1D wavelets are shown in Fig. 2.3, but these

[7] Pinning it down to this date helps keep a clean narrative but under-sells the work on wavelets that followed in the 1980s and the precursors dating as far back as 1910. Luckily Heil and Walnut compiled a full list of the key papers in the development of wavelet theory [105].

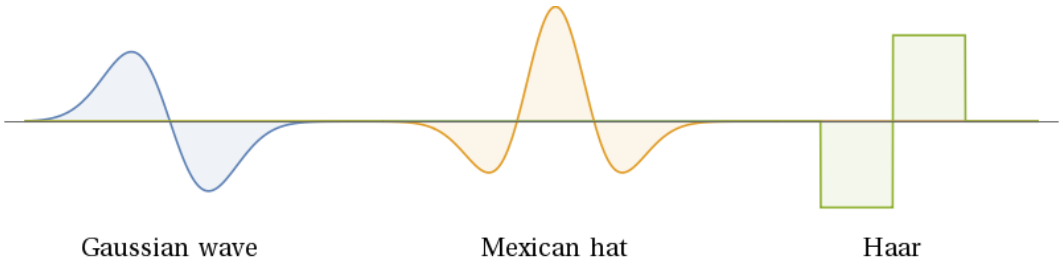

Figure 2.3: Examples of commonly used 1-dimensional scalar wavelets.

are by no means the only options. Rather these 3 provide a clean way to see just how many functions can be classified as a wavelet. The 'Gaussian wave' and 'Mexican hat' wavelets are none other than the first and second derivatives of a Gaussian, and all higher derivatives are valid wavelets too. The Haar wavelet meanwhile highlights that wavelets need not be smooth functions.

There are only 2 conditions real wavelets[8] must satisfy, and they can be satisfied by an infinite set of functions. The first condition is that the wavelet function $\psi(t)$ is square integrable [106, 112]:

$$\int_{-\infty}^{\infty} dt\, |\psi(t)|^2 < \infty. \tag{2.55}$$

The second condition, called the 'admissibility condition', can be expressed in a few ways:

$$\begin{aligned} c_\psi = \int_0^\infty d\omega\, \frac{|\tilde{\psi}(\omega)|^2}{|\omega|} < \infty; \\ \tilde{\psi}(0) = 0; \\ \int_{-\infty}^{\infty} dt\, \psi(t) = 0, \end{aligned} \tag{2.56}$$

with the tilde denoting a Fourier transform. c_ψ will later play a role akin to the factors of 2π in Fourier transforms, but for now the third line in Eq. (2.56) is the clearest: wavelets have zero mean.

Strictly, $\psi(t)$ is called the 'mother wavelet'. Once the mother wavelet func-

[8]There is an additional constraint for complex wavelets but only purely real functions are considered in this thesis.

tion is chosen, the wavelets themselves are translated by b and scaled by λ as[9]

$$\psi(t) \to \lambda^{-\frac{1}{2}} \psi\left(\frac{t-b}{\lambda}\right). \tag{2.57}$$

With this, the forward and reverse wavelet transforms are

$$S(\lambda, b) = \int_{-\infty}^{\infty} dt\, s(t)\, \lambda^{-\frac{1}{2}} \psi^*\left(\frac{t-b}{\lambda}\right), \tag{2.58}$$

$$s(t) = c_\psi^{-1} \int_0^{\infty} \frac{d\lambda}{\lambda} \int_{-\infty}^{\infty} \frac{db}{\lambda} S(\lambda, b)\, \lambda^{-\frac{1}{2}} \psi\left(\frac{t-b}{\lambda}\right). \tag{2.59}$$

A consequence of the (continuous) wavelets not being orthogonal is that the definition for $S(\lambda, b)$ in Eq. (2.58) is not unique, even for a given mother wavelet. This will be ignored in this thesis, and indeed many of the formal details presented in this section will make only a fleeting appearance when setting up the algorithm before being swept under the rug. What will recur however are the wavelet parameters b and λ, the latter playing a particularly prominent role not only in the upcoming mathematics but also in interpreting physical results in later chapters. It is therefore worth spending some time getting accustomed to these quantities.

As has already been said b and λ translate and scale the wavelet, but what does that mean for $S(\lambda, b)$? To help illustrate the answer Fig. 2.4 shows two different signals and their corresponding continuous wavelet transforms, as defined in Eq. (2.58). The left signal presents the basics with a sum of decaying sine waves. The presence of each frequency manifests itself as a band around the corresponding frequency in the CWT. The non-orthogonality of the wavelets accounts for the width of these bands, as opposed to the sharp spectrum of the Fourier transform.

The width of the bands depends on the choice of mother wavelet, but more important is how they decay in b just as the sine waves decay with t. The CWT therefore reveals not only which frequencies are present, but also how their amplitudes vary in time. This is especially useful when looking for transient signals as highlighted by the right example in Fig. 2.4. Here there is some uninteresting noise with a couple of pulses that are well separated from the noise in the CWT. A standard technique is to set a threshold to cut out most of the (high-frequency) noise in the CWT and transform back, giving a cleaned signal [116, 117].

The interpretation of wavelets in the work to come will not be quite so simple as in these 1-dimensional examples, but the key points will nonetheless persist.

[9]There is no standard notation for these parameters and different references often use a completely different set of symbols. The use of λ for scale is intended to invoke its role as a wavelet analogue to wavelength.

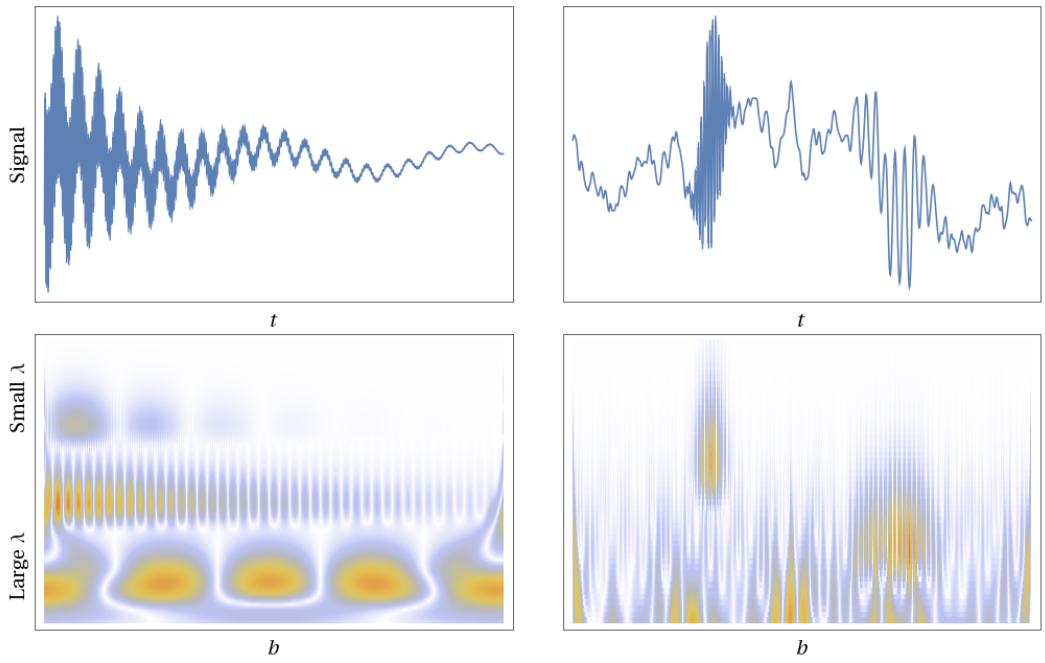

Figure 2.4: Example time-varying signals (top) and intensity maps of their associated CWTs using the Mexican hat wavelet (bottom). The left signal is a sum of exponentially decaying sine waves while the right has transient oscillations superposed with noise.

Namely that the signal (which will be the fluid flow field) can be decomposed into a distribution of 'frequencies' local to each position.

2.7.2 3-dimensional wavelets

Moving up to 3 dimensions makes a few changes to wavelets and the CWT, although the underlying structure is essentially the same. The conditions on (real) mother wavelets just become integrals over 3D space [112]:

$$\int_{-\infty}^{\infty} d^3\mathbf{r}\, \psi(\mathbf{r})^2 \quad < \quad \infty; \tag{2.60}$$

$$c_\psi = \int_{0}^{\infty} d^3\mathbf{k}\, \frac{|\tilde{\psi}(\mathbf{k})|^2}{|\mathbf{k}|^3} \quad < \quad \infty; \tag{2.61}$$

$$\Rightarrow \int_{-\infty}^{\infty} d^3\mathbf{r}\, \psi(\mathbf{r}) \quad = \quad 0. \tag{2.62}$$

The more involved change is to the 3D version of Eq. (2.57), and most significantly the introduction of a third wavelet parameter, $\hat{\mathbf{p}}$, to set its orientation in 3D space. The use of the mother wavelet then requires a rotation from the orientation

$\hat{\mathbf{p}}$ onto the orientation of the mother wavelet, using the rotation matrix $\boldsymbol{\mathcal{R}}(\hat{\mathbf{p}})$. The 3D wavelets are then expressed as

$$\psi(\mathbf{r}) \to \lambda^{-\frac{3}{2}}\psi\left(\lambda^{-1}\boldsymbol{\mathcal{R}}(\hat{\mathbf{p}})\cdot(\mathbf{r}-\mathbf{b})\right) = \lambda^{-\frac{3}{2}}\psi\left(\frac{\mathbf{r}-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right), \tag{2.63}$$

where the second, less explicit form is written for consistency with later chapters. The 3D CWT and its inverse can then be shown to be [105]

$$S(\lambda,\mathbf{b},\hat{\mathbf{p}}) = \int_{-\infty}^{\infty} d^3\mathbf{r}\, s(\mathbf{r})\,\lambda^{-\frac{3}{2}}\psi^*\left(\frac{\mathbf{r}-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right), \tag{2.64}$$

$$s(\mathbf{r}) = c_\psi^{-1}\int_0^{\infty}\frac{d\lambda}{\lambda}\int_{-\infty}^{\infty}\frac{d^3\mathbf{b}}{\lambda^3}\oint d^2\hat{\mathbf{p}}\, S(\lambda,\mathbf{b},\hat{\mathbf{p}})\,\lambda^{-\frac{3}{2}}\psi\left(\frac{\mathbf{r}-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right). \tag{2.65}$$

The multi-dimensional scalar wavelets above are well established [112], but still not quite what will be needed. The function of interest in this work is not the scalar $s(\mathbf{r})$ but a second rank tensor. Continuing to use scalar wavelets so the CWT is also a second rank tensor by integrating each element separately would work, however it will be valuable to introduce vector wavelets and make the tensor with appropriate use of products in the CWT and its inverse.

A simple and useful example to show this is the vector CWT of $\delta(\mathbf{r}_i-\mathbf{r}_j)\mathbf{I}$. Denoting vector wavelets with $\mathbf{w}$, this is

$$\begin{aligned}\mathbf{S}(\lambda,\mathbf{b},\hat{\mathbf{p}}) &= \int_{-\infty}^{\infty} d^3\mathbf{r}_i\,\delta(\mathbf{r}_i-\mathbf{r}_j)\,\mathbf{I}\cdot\left(\lambda^{-\frac{3}{2}}\mathbf{w}\left(\frac{\mathbf{r}_i-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right)\right)\\ &= \lambda^{-\frac{3}{2}}\mathbf{w}\left(\frac{\mathbf{r}_j-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right).\end{aligned} \tag{2.66}$$

The inverse vector CWT then needs to construct a second rank tensor out of two vectors, implying

$$\delta(\mathbf{r}_i-\mathbf{r}_j)\mathbf{I} = c_{\mathbf{w}}^{-1}\int d\lambda\, d^3\mathbf{b}\, d^2\hat{\mathbf{p}}\,\lambda^{-7}\mathbf{w}\left(\frac{\mathbf{r}_j-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right)\otimes\mathbf{w}\left(\frac{\mathbf{r}_i-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right). \tag{2.67}$$

While this specific result will not be used in this thesis, it is functionally almost identical to the result that will underpin WMCD.

3

Wavelet Monte Carlo dynamics

This chapter presents Wavelet Monte Carlo dynamics (WMCD) in its basic form, as published in 2017 [1], while extensions and modifications will be presented in Chapter 4. As the algorithm is being developed from scratch there is no avoiding the gritty details, and these two chapters aim to be as comprehensive as possible for others to work from. Implementation-dependent details will generally be omitted however.

With that in mind it is worth being clear where this is heading from the start. So, what is WMCD?

WMCD *is an implicit-solvent dynamic Monte Carlo algorithm which uses wavelet moves to efficiently evolve systems of Brownian particles hydrodynamically coupled by (a regularised version of) the Oseen tensor.*

Without further ado, let's dive in.

3.1 Wavelet representation of the Oseen tensor

To approach going forward is best viewed as a solver of Eq. (2.19), here repeated for convenience:

$$\langle \delta \mathbf{r}_i \otimes \delta \mathbf{r}_j \rangle = 2k_B T \boldsymbol{\mathcal{G}}_{ij} \delta t. \tag{3.1}$$

Unlike Brownian dynamics the key to WMCD is not to view this as an equation to solve given some $\boldsymbol{\mathcal{G}}_{ij}$, but just to re-express it as an integral:

$$\boldsymbol{\mathcal{G}}_{ij}(\mathbf{r}_{ij}) = \int d\boldsymbol{\Lambda}\mathcal{P}_{\boldsymbol{\Lambda}}\ \delta\mathbf{r}(\mathbf{r}_i, \boldsymbol{\Lambda}) \otimes \delta\mathbf{r}(\mathbf{r}_j, \boldsymbol{\Lambda}), \tag{3.2}$$

suppressing constants for the time being. The integral over the set of parameters $\boldsymbol{\Lambda}$, which collectively have the probability density function (PDF) $\mathcal{P}_{\boldsymbol{\Lambda}}$, is then interpreted as taking the expectation value of the product $\delta\mathbf{r}_i \otimes \delta\mathbf{r}_j$.

From the point of view of Monte Carlo simulations, this integral encodes all the necessary information for an algorithm which correlates particle displacements via the Oseen tensor, namely the form of the Monte Carlo move $\delta\mathbf{r}$, its parameters $\boldsymbol{\Lambda}$, and their distribution $\mathcal{P}_{\boldsymbol{\Lambda}}$. The procedure in each move is then to generate $\boldsymbol{\Lambda}$ and displace particles simultaneously by the vector $\delta\mathbf{r}$ at their position.

One option is to use the known Fourier transform of the Oseen tensor and rearrange the inverse transform into the form in Eq. (3.2). Fourier moves will return in Section 3.3, but they are disregarded here on the grounds that they are too inefficient computationally because the plane-wave displacement fields have infinite extent and hence every particle needs updating in every move. A more efficient approach is to use the localised vector wavelets discussed in Section 2.7, when Eq. (3.2) becomes

$$\boldsymbol{\mathcal{G}}_{ij}(\mathbf{r}_{ij}) = \mathcal{N}_{\boldsymbol{\mathcal{G}}} \int_0^\infty d\lambda \int_{-\infty}^{\infty} d^3\mathbf{b} \oint d^2\hat{\mathbf{p}}\, \lambda^{-5}\mathbf{w}\left(\frac{\mathbf{r}_i - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \otimes \mathbf{w}\left(\frac{\mathbf{r}_j - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right), \tag{3.3}$$

with $\mathcal{N}_{\boldsymbol{\mathcal{G}}}$ a constant derived below.

Eq. (3.3) was constructed to match dimensions of length on either side, but nonetheless needs proving. This is easiest in Fourier space, so first the Fourier transform with respect to $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is taken:

$$\begin{aligned}
\tilde{\boldsymbol{\mathcal{G}}}(\mathbf{k}) &= \int d^3\mathbf{r} e^{-ik\cdot(\mathbf{r}_i - \mathbf{r}_j)} \boldsymbol{\mathcal{G}}_{ij}(\mathbf{r}_{ij}) \\
&= \mathcal{N}_{\boldsymbol{\mathcal{G}}} \int d^2\hat{\mathbf{p}} \frac{d\lambda}{\lambda} \lambda^2 \tilde{\mathbf{w}}\left(\lambda\mathbf{k}, \hat{\mathbf{p}}\right) \otimes \tilde{\mathbf{w}}\left(-\lambda\mathbf{k}, \hat{\mathbf{p}}\right),
\end{aligned} \tag{3.4}$$

where $\tilde{\mathbf{w}}(\mathbf{k}, \hat{\mathbf{p}}) = \int d^3\mathbf{x} e^{-i\mathbf{k}\cdot\mathbf{x}} \mathbf{w}(\mathbf{x}, \hat{\mathbf{p}})$ is the Fourier transform of the mother wavelet at fixed $\hat{\mathbf{p}}$.

Integrating over the wavelet polarisation obtains

$$\tilde{\boldsymbol{\mathcal{G}}}(\mathbf{k}) = 4\pi\mathcal{N}_{\boldsymbol{\mathcal{G}}} \int d\lambda\, \lambda\, \boldsymbol{W}(\lambda\mathbf{k}); \tag{3.5}$$

$$\boldsymbol{W}(\mathbf{k}) = \frac{1}{4\pi} \int d^2\hat{\mathbf{p}}\, \tilde{\mathbf{w}}\left(\mathbf{k}, \hat{\mathbf{p}}\right) \otimes \tilde{\mathbf{w}}\left(-\mathbf{k}, \hat{\mathbf{p}}\right). \tag{3.6}$$

It is at this point it is helpful to impose the incompressibility condition on the wavelets:

$$\nabla \cdot \mathbf{w} = 0, \tag{3.7}$$

which serves as the wavelet equivalent to fluid incompressibility condition in Eq. (2.2). This guarantees the mobility tensor is divergenceless while giving $\mathbf{k} \cdot \boldsymbol{W} = 0$, from which it is possible to infer $\boldsymbol{W}$ must have the tensor structure

$$\boldsymbol{W}(\mathbf{k}) = (\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}) f(k) \tag{3.8}$$

where

$$2f(k) = \mathrm{Tr}\left[\boldsymbol{W}(\mathbf{k})\right] = \frac{1}{4\pi} \int d^2\hat{\mathbf{p}}\, \tilde{\mathbf{w}}\left(\mathbf{k}, \hat{\mathbf{p}}\right) \cdot \tilde{\mathbf{w}}\left(-\mathbf{k}, \hat{\mathbf{p}}\right). \tag{3.9}$$

Reassembling all this and making the substitution $q = \lambda k$ leads to

$$\tilde{\boldsymbol{\mathcal{G}}}(\mathbf{k}) = 4\pi \mathcal{N}_{\boldsymbol{\mathcal{G}}} k^{-2} (\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}) \int_0^\infty dq\, q f(q). \tag{3.10}$$

This tensor structure and k-dependence matches that of the Oseen tensor's Fourier transform $(1/\eta k^2)(\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}})$ [8], leaving the normalisation to correct for the remaining constants. Thus $\mathcal{N}_{\boldsymbol{\mathcal{G}}}$ is required to be

$$\mathcal{N}_{\boldsymbol{\mathcal{G}}}^{-1} = 4\pi\eta \int_0^\infty dq\, q f(q). \tag{3.11}$$

Even with Eq. (3.3) validated, there is still some freedom on how to choose $\delta\mathbf{r}$ and the PDFs for each of λ, $\mathbf{b}$ and $\hat{\mathbf{p}}$. In particular, while it is evident that $\delta\mathbf{r} \propto \mathbf{w}$, the constant of proportionality need not be 1. This constant will be called A_w, and it is to be noted that it may depend on λ, $\mathbf{b}$ and $\hat{\mathbf{p}}$.

Eq. (3.3) can then be written as

$$\boxed{\begin{aligned} \boldsymbol{\mathcal{G}}_{ij}(\mathbf{r}_{ij}) &= \int d\boldsymbol{\Lambda}\, \mathcal{P}_\lambda\, \mathcal{P}_{\mathbf{b}}\, \mathcal{P}_{\hat{\mathbf{p}}}\, A_w \mathbf{w}\left(\frac{\mathbf{r}_i - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \otimes A_w \mathbf{w}\left(\frac{\mathbf{r}_j - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \\ &= \left\langle A_w \mathbf{w}\left(\frac{\mathbf{r}_i - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \otimes A_w \mathbf{w}\left(\frac{\mathbf{r}_j - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \right\rangle_{\boldsymbol{\Lambda}}, \end{aligned}} \tag{3.12}$$

with $\boldsymbol{\Lambda}$ again standing in for the set of move parameters, and the subscript on the angle bracket indicating the parameters averaged over. Most of the effort in turning this into a Monte Carlo algorithm lies in choosing distributions such that

$$\mathcal{P}_\lambda\, \mathcal{P}_{\mathbf{b}}\, \mathcal{P}_{\hat{\mathbf{p}}}\, A_w^2 \propto \lambda^{-5}. \tag{3.13}$$

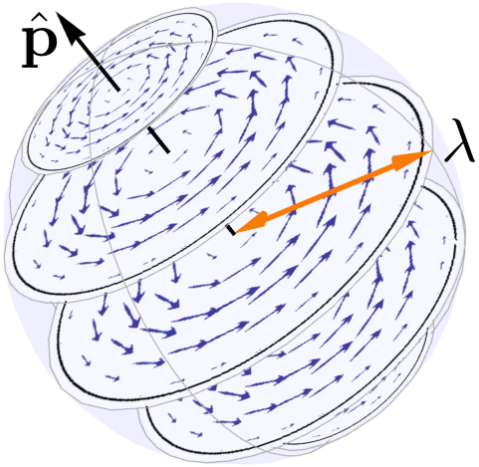

Figure 3.1: Example of a wavelet move. Displacements follow a vector field rotating about the $\hat{\mathbf{p}}$ axis (black arrow) inside a sphere of radius λ. The vector field shown is for the cubic wavelet.

3.2 Wavelet parameter distributions

This section presents the calculations of the individual parameter PDFs, for which it will be helpful to constrain some properties of the mother wavelet. Any vector field satisfying Eq. (3.7) and vectorised versions of Eq.s (2.62) and (2.60) is a valid choice. The endless freedom this provides is a blessing when optimising the minutiae but is decidedly unhelpful when just starting out. In any case, to obtain results that aren't so generic as to be useless this work focusses on wavelets with vector fields rotating about the $\hat{\mathbf{p}}$ axis, as depicted in Fig. 3.1.

This allows the mother wavelet to be written in terms of a spherically symmetric scalar function $\phi(r)$ as[1]

$$\mathbf{w}(\mathbf{r}, \hat{\mathbf{p}}) = \begin{cases} \hat{\mathbf{p}} \times \nabla\phi(r) & \text{for } |\mathbf{r}| \leqslant 1 \\ \mathbf{0} & \text{for } |\mathbf{r}| > 1 \end{cases} . \tag{3.14}$$

Even here there is an infinite set of allowed wavelets, with perhaps the most significant feature being continuity, including at $|\mathbf{r}| = 1$. While not strictly required, how smooth $\phi(r)$ is impacts both the strain energy calculation in Section 3.2.2 and the analysis of the choice of ϕ in Section 4.4, and it is generally best to avoid discontinuous wavelets.

Explicit results will sometimes be needed in the meantime, and these will use

[1] In practice this can be implemented as an exact rotation with rotation angle $A_w|\mathbf{w}|/(r\sin(\cos^{-1}(\hat{\mathbf{r}}\cdot\hat{\mathbf{p}})))$ rather than a linear displacement. This has the advantage of being exactly reversible but makes analytic calculations more challenging. Displacements will be small enough that the two approaches give the same results and the simpler linear version is used in all calculations.

the 'cubic wavelet', which in spherical polar coordinates (r, θ, φ), and polarisation vector along the z-axis ($\theta = 0$), is given by

$$\mathbf{w}(\mathbf{r}, \hat{\mathbf{z}}) = \begin{cases} r \sin\theta (1-r) \hat{\boldsymbol{\varphi}} & \text{for } |\mathbf{r}| \leqslant 1 \\ \mathbf{0} & \text{for } |\mathbf{r}| > 1 \end{cases} . \tag{3.15}$$

The associated ϕ and its Fourier transform are readily found to be

$$\phi(r) = \begin{cases} \frac{1}{2} r^2 - \frac{1}{3} r^3 - \frac{1}{6} & \text{for } r \leqslant 1 \\ 0 & \text{for } r > 1 \end{cases} , \tag{3.16}$$

$$\tilde{\phi}(k) = 4\pi \, k^{-6} \left(5k \sin k - (k^2 - 8) \cos k - 8 \right) . \tag{3.17}$$

Analytical calculations remain as general as possible, and are expressed in terms of ϕ. In many cases the m^{th} moment of the square of $\tilde{\phi}$ appears and will be denoted by

$$M_m \equiv \int_0^\infty dk \, k^m \tilde{\phi}(k)^2 . \tag{3.18}$$

3.2.1 Wavelet orientation and centre

With the form of the mother wavelet chosen, the parameter PDFs can be found. Again these choices are not unique and there are multiple valid routes to satisfying Eq. (3.13). The choices in this thesis are motivated on either computational or physical grounds.

The simplest parameter to address is the orientation $\hat{\mathbf{p}}$. Short of a fundamentally anisotropic system, there is no reason for $\hat{\mathbf{p}}$ to have a non-uniform distribution, and even then it is easiest to let the Monte Carlo acceptance test[2] handle the effects of anisotropy. Therefore the appropriate PDF is simply

$$\boxed{\mathcal{P}_{\hat{\mathbf{p}}} = \frac{1}{4\pi} .} \tag{3.19}$$

The PDF for $\mathbf{b}$ is either equally simple or much more involved depending on how efficient the algorithm needs to be. Since the integrand in Eq. (3.3) has no $\mathbf{b}$-dependence outside of the wavelets, $\mathcal{P}_{\mathbf{b}}$ can be taken as uniform whereupon one has

$$\mathcal{P}_{\mathbf{b}} = \frac{1}{L^3} \tag{3.20}$$

in a cubic simulation box of side length L.

However, such a distribution could generate wavelet moves that contain zero particles and hence do not update the system of interest and in turn waste com-

[2] Or better yet let smart MC bias the moves for you, but that is a story for Section 2.6.3.

putation time. This is especially wasteful in systems at low concentrations. A work-around for this is to choose $\mathbf{b}$ such that each move is guaranteed to displace at least one particle. To do this one can use the PDF

$$\mathcal{P}_{\mathbf{b}|\lambda} = \mathcal{N}_{\mathbf{b}} \sum_{i=1}^{N} \Theta(\lambda - |\mathbf{r}_i - \mathbf{b}|) \tag{3.21}$$

with Θ the Heaviside step function and the change of notation to $\mathcal{P}_{\mathbf{b}|\lambda}$ distinguishing this result from the equally valid uniform case above. With this PDF each particle contributes either 0 or 1 to the sum depending on whether they fall inside a sphere of radius λ centred on $\mathbf{b}$. I.e. within the wavelet. (Note that while the PDF for λ is not yet known, it can still be assumed chosen before $\mathbf{b}$ during a simulation.) Thus the sum simply counts the number of contained particles, or 'movers', which will be denoted by n. Integrating over $\mathbf{b}$ to find the normalisation leads to the equivalent form

$$\boxed{\mathcal{P}_{\mathbf{b}|\lambda} = \frac{3}{4\pi\lambda^3}\frac{n}{N}.} \tag{3.22}$$

$\mathcal{P}_{\mathbf{b}|\lambda}$ will be taken as the default PDF for $\mathbf{b}$ from now on.

3.2.2 Wavelet amplitude

The form of A_w is perhaps the most important as it actually enters into the particle displacements in each move, and hence has a direct influence on the energy changes and in turn the move's acceptance probability P_{acc}. If there is significant variation in P_{acc} across moves, the Oseen tensor would not be faithfully reproduced. For example if P_{acc} decreases sharply with increasing λ, the long-ranged hydrodynamic interactions supplied by moves with large radii would be under-represented and the effective mobility tensor would drop off faster than as $1/r$.

Therefore the approach is to choose A_w such that the typical energy change ΔU is independent of move parameters (of course on a move by move basis ΔU is sensitive to the details of the system, but estimates for a notional 'average system' can nonetheless be made). To estimate the parameter dependence of ΔU, consider the strain energy associated with a move [118]

$$\Delta U_{\text{est}} = \frac{1}{2} G \int d^3\mathbf{r} \left(\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}^{\mathrm{T}} \right), \tag{3.23}$$

with strain tensor $\boldsymbol{\varepsilon}$ and shear modulus G. The colons (:) denote double dot products.

This would estimate the energy change across the whole move, including the implicit fluid that does not enter into P_{acc}. It should therefore be modified to estimate the strain energy of the n moving particles instead. Assuming each of these

feels the average strain in the move the modulus can be estimated by

$$G \approx \frac{3}{4\pi\lambda^3} n U_0, \tag{3.24}$$

with U_0 a system-dependent particle interaction energy expected to be of order $k_B T$. The estimated energy is then

$$\Delta U_{\text{est}} = \frac{3nU_0}{8\pi\lambda^3} \int d^3\mathbf{r} \left(\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}^{\mathrm{T}} \right). \tag{3.25}$$

To perform the integral in Eq. (3.25) the displacements are assumed sufficiently small to use the infinitesimal strain tensor $\boldsymbol{\varepsilon} = A_w \left(\nabla \mathbf{w} + (\nabla \mathbf{w})^{\mathrm{T}} \right)/2$. Also, the assumption that all n particles see the same average strain means the integral is independent of $\mathbf{b}$ and $\hat{\mathbf{p}}$, so it is sufficient to replace $r \to r/\lambda$ in the mother wavelet. The substitution $\mathbf{x} = \mathbf{r}/\lambda$ leads to $d^3\mathbf{r} = \lambda^3 d^3\mathbf{x}$ and $\nabla = \lambda^{-1}\nabla_{\mathbf{x}}$, with $\nabla_{\mathbf{x}}$ the gradient with respect to $\mathbf{x}$, and hence

$$\Delta U_{\text{est}} = \frac{3nU_0}{8\pi\lambda^3} \lambda A_w^2 \int d^3\mathbf{x} \left(\nabla_{\mathbf{x}}\mathbf{w} : \nabla_{\mathbf{x}}\mathbf{w} + \nabla_{\mathbf{x}}\mathbf{w} : (\nabla_{\mathbf{x}}\mathbf{w})^T \right). \tag{3.26}$$

The left term is zero, as seen by integrating it by parts and using the divergencelessness and continuity of $\mathbf{w}$ to kill off the resulting integral and boundary terms respectively.

The integral of the right term can be written in terms of the wavelet's Fourier transform as

$$\frac{1}{(2\pi)^6} \int d^3\mathbf{x}\, d^3\mathbf{k}\, d^3\mathbf{q}\, (i\mathbf{k} \cdot i\mathbf{q}) \left(\tilde{\mathbf{w}}(\mathbf{k}) \cdot \tilde{\mathbf{w}}(\mathbf{q}) \right) e^{i(\mathbf{k}+\mathbf{q})\cdot\mathbf{x}}. \tag{3.27}$$

Integrating over $\mathbf{x}$ yields $(2\pi)^3\delta(\mathbf{k}+\mathbf{q})$, so performing the $\mathbf{q}$ integral gives

$$\frac{1}{(2\pi)^3} \int d^3\mathbf{k}\, k^2 \tilde{\mathbf{w}}(\mathbf{k}) \cdot \tilde{\mathbf{w}}(-\mathbf{k}). \tag{3.28}$$

From here it is necessary to specify the form of $\tilde{\mathbf{w}}$, so the rotational form in Eq. (3.14) with Fourier transform $\tilde{\mathbf{w}}(\mathbf{k}) = i(\hat{\mathbf{p}} \times \mathbf{k})\tilde{\phi}(|\mathbf{k}|)$ needs to be substituted in. Lastly, $(\hat{\mathbf{p}} \times \mathbf{k}) \cdot (\hat{\mathbf{p}} \times \mathbf{k}) = k^2 \sin^2\theta$ can be used to integrate out the angular part to reach the final form of

$$\frac{8\pi}{3(2\pi)^3} \int_0^\infty dk\, k^6 \tilde{\phi}(k)^2, \tag{3.29}$$

and hence

$$\Delta U_{\text{est}} = \frac{U_0 M_6}{(2\pi)^3} \frac{n A_w^2}{\lambda^2}. \tag{3.30}$$

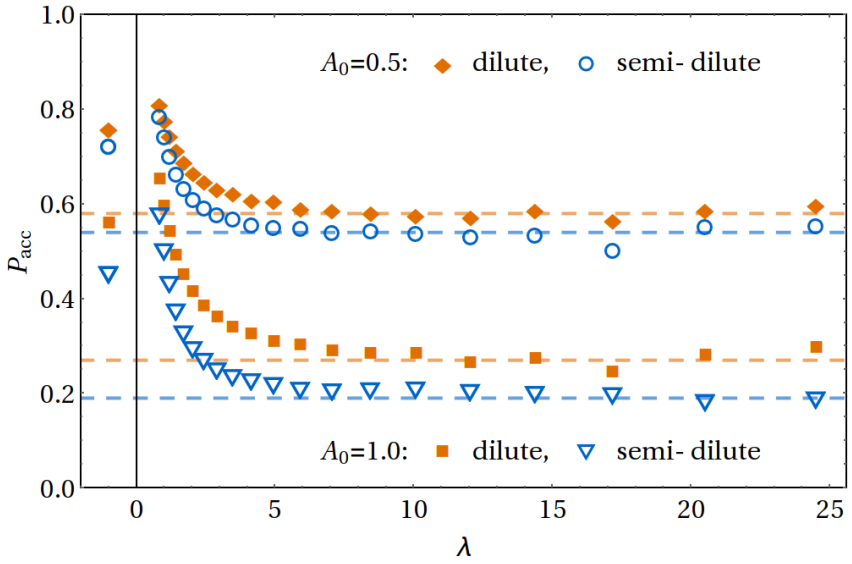

Figure 3.2: Acceptance probabilities over the spectrum of wavelet radii at different move amplitudes. The dilute data used an isolated polymer chain with $N = 2048$, and the semi-dilute data used chains 10 beads long at a bead concentration $N/L^3 = 0.625$ in a periodic box. Other system parameters are as listed in Section 3.5.1. The dashed lines are only to emphasize the asymptotic behaviour, while the markers at $\lambda = -1$ indicate the average P_{acc} over all wavelets.

Requiring this to be independent of move parameters implies A_w must take the form

$$\boxed{A_w(\lambda, n(\mathbf{b}, \lambda)) = A_0\lambda/\sqrt{n},} \tag{3.31}$$

with A_0 a dimensionless constant. In practice A_0 is taken as a free parameter (and indeed Section 3.5 will show it acts as WMCD's primary stand-in for a time step size), so that this form for A_w is enough to proceed.

To show how successful Eq. (3.31) is at fixing the move energy, the λ-spectrum of P_{acc} is shown in Fig. 3.2. This shows that for moves with large radii the strain energy calculation does lead to a move-independent energy change, but for small wavelets it fails. This can be explained by considering a wavelet that contains only a single particle: in this case the relative displacement of this particle to its neighbours is $A_w\mathbf{w}$, rather than $A_w\nabla\mathbf{w}$ as was assumed above. The quantitative difference can also be reasoned without further calculation, because $A_w \to 0$ as $\lambda \to 0$. Consequently $\Delta U \to 0$ and hence $P_{\text{acc}} \to 1$.

Recalling the reason for the strain energy calculation, this small λ behaviour forces a choice on how to remedy it, for which 4 options are presented. The first is to reduce A_0 enough to raise the asymptotic value of P_{acc} and keep its variation within acceptable bounds. Section 3.7 will show this comes at a cost to computation time, which is directly analogous to reducing the time step in other algorithms to

improve fidelity of data.

The second and perhaps ideal solution is to modify the λ-dependence in either A_w or $\mathcal{P}_\lambda$ (addressed in the next section) to compensate for the variation in P_{acc}. This is also the most complicated, and the difference between the dilute and semi-dilute data in Fig. 3.2 suggests this would be a system-dependent fix.

An easier solution to implement is to use a move recycling scheme, whereby the value of λ for any rejected move is reused in the next ($\mathbf{b}$, $\hat{\mathbf{p}}$ and A_w would be generated as normal). In this way the distribution of λ in accepted moves is guaranteed to be the same as in $\mathcal{P}_\lambda$. However, this runs a risk of disrupting the distributions of $\mathbf{b}$ and $\hat{\mathbf{p}}$. To see why, imagine an inhomogeneous system which has a dense cluster of particles in one region and a relatively sparse region elsewhere. It is more likely a move will be rejected in the dense region than the sparse one, so the net effect of the recycling scheme is a shift of some wavelets from the dense to the sparse region, and breaking the uniform distribution of $\mathcal{P}_{\mathbf{b}|\lambda}$. Similar arguments for anisotropic systems lead to breaking the uniformity of $\mathcal{P}_{\widehat{\mathbf{p}}}$, while detailed balance can also be violated if move recycling reduces or negates the effect of the acceptance test (consider the extreme case where *all* parameters are recycled to see this). As such, the recycling scheme must be used with care, if at all.

The final, and in practice preferable option is to change the acceptance test. The data in Fig. 3.2 used the Metropolis test, and, as per Section 2.6.4, improvements can be made to dynamical accuracy by switching to Glauber or, with a little more work as will be shown in Section 4.1, smart Monte Carlo. Data in later chapters uses the SMC version of the code with P_{acc} typically above 99% for all λ, so the concerns raised here have little impact on the main simulation results in this thesis. The results in this chapter continue to use the non-smart Metropolis test however.

3.2.3 Wavelet radii and the effect of finite cut-offs

With A_w^2, $\mathcal{P}_{\widehat{\mathbf{p}}}$ and $\mathcal{P}_{\mathbf{b}|\lambda}$ identified, $\mathcal{P}_\lambda$ is now determined by Eq. (3.13) to be

$$\mathcal{P}_\lambda \sim \lambda^{-4}. \tag{3.32}$$

To construct the Oseen tensor exactly λ would range from 0 to ∞, but both of these values lead to computational difficulties. Allowing infinitely large wavelets causes issues in periodic systems where large wavelets would overlap with their own images and individual particles see multiple parts of the wavelet. Computing such sums of displacements is possible but complicated, and it is much cleaner to set a maximum radius

$$\lambda_{\max} \leqslant L/2. \tag{3.33}$$

Although this concern does not exist in unbounded (non-periodic) systems, a finite value for $\lambda_{\max}$ is assumed here too and the results are easy to extend to $\lambda_{\max} \to \infty$ if desired. The details for filling in the missing $\lambda > \lambda_{\max}$ contribution will be discussed later.

At the other end of the spectrum, the singularity at $\lambda \to 0$ would lead to a non-normalisable $\mathcal{P}_\lambda$. Introducing a minimum radius $\lambda_{\min}$ resolves this, and one might think to set $\lambda_{\min}$ to be much smaller than any important length-scale of the system. In practice $\lambda_{\min}$ should take a well defined (and surprisingly large) value, but this discussion is not yet prepared to explain why and must defer the matter. In any case, with $\lambda_{\min}$ and $\lambda_{\max}$ present the PDF is

$$\boxed{\mathcal{P}_\lambda(\lambda) = \mathcal{N}_\lambda \lambda^{-4}; \qquad \mathcal{N}_\lambda = 3(\lambda_{\min}^{-3} - \lambda_{\max}^{-3})^{-1}.} \tag{3.34}$$

The effect of finite $\lambda_{\min}$ and $\lambda_{\max}$ on the mobility tensor must now be identified, starting from Eq. (3.10). The limits on this integral become $k\lambda_{\min}$ and $k\lambda_{\max}$, while substituting in the rotational form of wavelets leads to $f(q) \to q^2\tilde{\phi}(q)^2/3$ and $\mathcal{N}_{\boldsymbol{\mathcal{G}}} \to A_0^2$ to give

$$\tilde{\boldsymbol{\mathcal{G}}}_w(\mathbf{k}; \lambda_{\min}, \lambda_{\max}) = \frac{4\pi A_0^2}{3k^2} \left(\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}\right) \int_{k\lambda_{\min}}^{k\lambda_{\max}} dq\, q^3 \tilde{\phi}(q)^2. \tag{3.35}$$

Note the subscript on $\tilde{\boldsymbol{\mathcal{G}}}_w$ marking it as the tensor generated by wavelets.

For the purposes of taking the inverse Fourier transform, it will be helpful to use the limits $k\lambda'$ and ∞, then treat Eq. (3.35) as a sum of two such tensors. What is sought is then

$$\boldsymbol{\mathcal{G}}_w(r; \lambda', \infty) = \frac{A_0^2}{6\pi^2} \int_{\text{all } \mathbf{k}} d^3\mathbf{k}\, e^{i\mathbf{k}\cdot\mathbf{r}} k^{-2} \left(\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}\right) \int_{k\lambda'}^{\infty} dq\, q^3 \tilde{\phi}(q)^2. \tag{3.36}$$

The integral over q cannot be performed without knowing the form of $\tilde{\phi}(q)$, so the focus is on the integral over $\mathbf{k}$. The integrand is purely radial, as can be seen by writing $\hat{\mathbf{k}} \otimes \hat{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{r}} = -\nabla \otimes \nabla k^{-2} e^{i\mathbf{k}\cdot\mathbf{r}}$, so the angular integrals over the exponential give the usual result of $4\pi \sin(kr)/kr$.

Commuting the integrals and taking care with their limits yields

$$\boldsymbol{\mathcal{G}}_w(r; \lambda', \infty) = \frac{2A_0^2}{3\pi} \int_0^{\infty} dq\, q^3 \tilde{\phi}(q)^2 \int_0^{q/\lambda'} dk\, \left(\mathbf{I} + \nabla \otimes \nabla k^{-2}\right) \frac{\sin kr}{kr}, \tag{3.37}$$

in which the isotropic term readily integrates to

$$\int_0^{q/\lambda'} dk\, \frac{\sin kr}{kr}\mathbf{I} = \frac{\mathrm{Si}(Q)}{r}\,\mathbf{I}, \tag{3.38}$$

where $Q = qr/\lambda'$ and Si is the sine-integral function.

The other term appears to diverge at the $k \to 0$ limit, but it can be split into

$$\int_0^{q/\lambda'} dk\, \nabla\otimes\nabla\frac{\sin(kr)}{k^3 r} = \nabla\otimes\nabla\frac{1}{r}\left(\int_0^{\infty} dk\,\frac{\sin(kr)}{k^3} - \int_{q/\lambda'}^{\infty} dk\,\frac{\sin(kr)}{k^3}\right). \tag{3.39}$$

Although the first term still appears to diverge, it is exactly what would have been integrated if we had the full Oseen tensor and is therefore known to end up giving $(\pi/4r)(\hat{\mathbf{r}}\otimes\hat{\mathbf{r}} - \mathbf{I})$ [8].

That leaves the second integral with limits over a well behaved domain, allowing direct integration to give

$$\begin{aligned}
\nabla\otimes\nabla\frac{1}{r}\int_{q/\lambda'}^{\infty} dk\,\frac{\sin(kr)}{k^3} &= \frac{\lambda'}{2q}\nabla\otimes\nabla\left[\cos Q + Q^{-1}\sin Q + Q\left(\mathrm{Si}\,(Q) - \frac{\pi}{2}\right)\right] \\
&= \left(\mathrm{Si}\,(Q) - \frac{\pi}{2}\right)\frac{1}{2r}\,(\hat{\mathbf{r}}\otimes\hat{\mathbf{r}} - \mathbf{I}) \\
&\quad + \frac{\sin Q - Q\cos Q}{Q^2}\,\frac{1}{2r}\,(\mathbf{I} - 3\hat{\mathbf{r}}\otimes\hat{\mathbf{r}})\,.
\end{aligned} \tag{3.40}$$

The $\pi/2$ term cancels exactly with the full Oseen term, while the remaining terms plus the contribution from Eq. (3.38) gives the final result

$$\begin{aligned}
\boldsymbol{\mathcal{G}}_w(\mathbf{r};\lambda',\infty) = \frac{A_0^2}{3\pi r}\int_0^{\infty} dq\, q^3\tilde{\phi}(q)^2\Big[\,&(\mathbf{I} + \hat{\mathbf{r}}\otimes\hat{\mathbf{r}})\,\mathrm{Si}\,(Q) \\
&+ (\mathbf{I} - 3\hat{\mathbf{r}}\otimes\hat{\mathbf{r}})\,\frac{\sin Q - Q\cos Q}{Q^2}\Big],
\end{aligned} \tag{3.41}$$

with full tensor given by

$$\boldsymbol{\mathcal{G}}_w(\mathbf{r};\lambda_{\min},\lambda_{\max}) = \boldsymbol{\mathcal{G}}_w(\mathbf{r};\lambda_{\min},\infty) - \boldsymbol{\mathcal{G}}_w(\mathbf{r};\lambda_{\max},\infty). \tag{3.42}$$

The radial and angular elements of this tensor are plotted in Fig. 3.3, in both analytic form and as measured from correlations in particle displacements. For $\lambda_{\min} \lesssim r \lesssim \lambda_{\max}$ this replicates the $1/r$ decay of the Oseen tensor, with deviations outside this range coming from the missing small and large wavelets. The fall-off at large r can be understood intuitively by considering a pair of particles separated

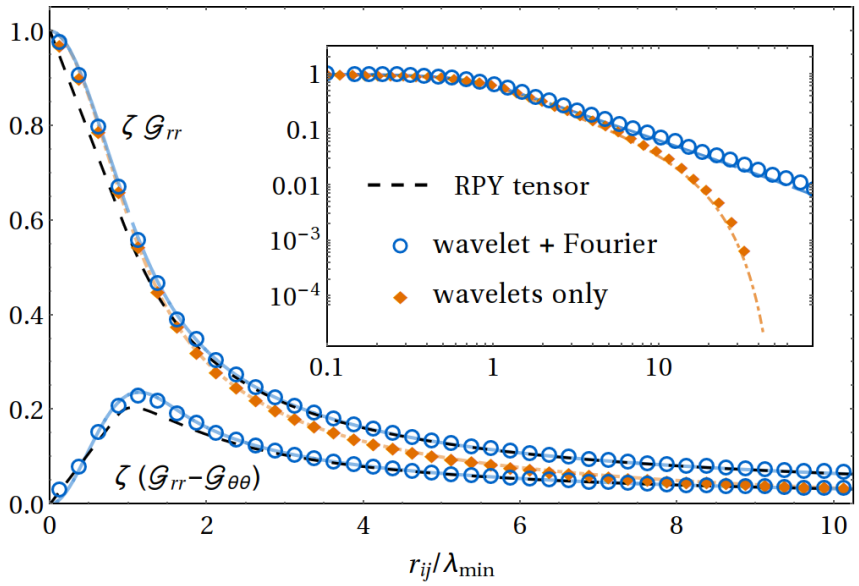

Figure 3.3: Plots of simulated mobility tensor elements normalised by $\zeta = 6\pi\eta a$ with the value of $\lambda_{\max} = 29\lambda_{\min}$. Theoretical curves from Eq. (3.42) lie underneath the data and curves for the full RPY tensor are also shown (dashed) for comparison. Wavelet-only data for $\zeta(\mathcal{G}_{rr} - \mathcal{G}_{\theta\theta})$ are not shown as they are indistinguishable from the wavelet + Fourier data over the domain shown. The inset shows $\zeta\mathcal{G}_{rr}$ using logarithmic scales to highlight the far-field decay. Note the form of the mobility tensor is specific to the cubic wavelet; the form for other wavelets is shown in Section 4.4.

by more than $2\lambda_{\max}$: there is no available wavelet large enough to include both particles, leading to their motion never being correlated. As $\lambda_{\max}$ increases the long-range correlations are more accurately supplied, limiting to the correct $1/r$ behaviour when $\lambda_{\max} \to \infty$ (the data for which was obtained using the Fourier moves described in the next section, but the same could in principle be achieved with large wavelets).

The regularisation of the asymptote at $r \to 0$ is actually beneficial as it remedies the failing of the Oseen tensor at short range, providing a clear link to the single particle mobility while keeping the tensor positive definite. To calculate $\boldsymbol{\mathcal{G}}_w$ in this limit and extract the implied single particle hydrodynamic radius a, $\lambda_{\max}$ is set to ∞ to remove its influence. Taking the Oseen and single particle limits of Eq. (3.42) one finds

$$\boldsymbol{\mathcal{G}}_w(r \to \infty; \lambda_{\min}, \infty) = \frac{A_0^2 M_3}{6r}(\mathbf{I} + \hat{\mathbf{r}} \otimes \hat{\mathbf{r}}) \qquad = \frac{1}{8\pi\eta r}(\mathbf{I} + \hat{\mathbf{r}} \otimes \hat{\mathbf{r}}), \tag{3.43}$$

$$\boldsymbol{\mathcal{G}}_w(r \to 0; \lambda_{\min}, \infty) = \frac{4A_0^2 M_4}{9\pi\lambda_{\min}}\mathbf{I} \qquad = \frac{1}{6\pi\eta a}\mathbf{I}. \tag{3.44}$$

Solving these for a finds an exact proportionality with $\lambda_{\min}$

$$\boxed{a = \frac{\lambda_{\min}}{\lambda_a}; \qquad \lambda_a \equiv \frac{2}{\pi}\frac{M_4}{M_3}} \tag{3.45}$$

so long as the moments M_3 and M_4 exist.

The consequences of Eq. (3.45) are significant. Through it WMCD has an explicit and simple expression for a. Moreover, with $\lambda_a \sim 2$ (no wavelets were ever found with a value smaller than 2), this sets the minimum wavelet volume to be ~ 8 times that of individual particles so there is never a danger of spending time on negligibly small moves.

A second result that will be useful in later comparisons to mobility tensors is

$$A_0^2 = \frac{3}{4\pi\eta M_3}. \tag{3.46}$$

It is worth stressing this result is used specifically for comparing mobility tensors. In actual simulations the additional factors of $2k_BT\delta t$ from Eq. (3.1) enter, though δt is yet to be discussed.

The final point to make about $\boldsymbol{\mathcal{G}}_w$ is that it closely approximates the RPY tensor defined in Eq.s (2.12) and (2.13), as evidenced by the dashed lines in Fig. 3.3. It is possible to modify WMCD to replicate the RPY tensor more closely using Faxén's laws [119], but with how close it is already and the fact that the RPY tensor is itself an approximation (albeit a higher order one than Oseen), there is little motivation to do so.

3.3 Fourier representation of the missing mobility

This section addresses how to include long-ranged interactions when $\lambda_{\max}$ is finite, or more precisely: how to construct $\boldsymbol{\mathcal{G}}_w(\mathbf{r}; \lambda_{\max}, \infty)$ without wavelets. While large wavelets are awkward to work with in periodic systems, the philosophy behind their move structure can be carried over to any arbitrary function, and what is sought are functions that can replace them with an easy periodic sum. The natural choice for this purpose is plane wave, or 'Fourier', moves[3]. The details of these moves are relatively simple when compared to wavelets except the necessity to distinguish between results for periodic and unbounded systems. This is done through the use of the superscripts $\bullet^P$ and $\bullet^\infty$ for each case respectively.

Conveniently, in unbounded systems the Fourier tensor replaces the missing part of the wavelet tensor with no modification, which is most helpfully expressed

[3]This is analogous to the split into real and reciprocal space in the Ewald sum used in periodic Brownian dynamics simulations.

in the Fourier basis in Eq. (3.35):

$$\tilde{\boldsymbol{\mathcal{G}}}_F^{\infty}(\mathbf{k};\lambda_{\max}) = \tilde{\boldsymbol{\mathcal{G}}}_w(\mathbf{k};\lambda_{\max},\infty) = \frac{4\pi A_0^2}{3k^2}\left(\mathbf{I}-\hat{\mathbf{k}}\otimes\hat{\mathbf{k}}\right)\int_{k\lambda_{\max}}^{\infty} dq\, q^3\tilde{\phi}(q)^2. \tag{3.47}$$

The periodic version sees only the wavevectors commensurate with the simulation box, and is essentially the Fourier series equivalent of the Fourier transform above. From a more physical perspective, the flow field from a given periodic particle is the sum of flow fields from each of the particle's images, and is therefore a convolution of $\tilde{\boldsymbol{\mathcal{G}}}_F^{\infty}$ and a Dirac comb with delta functions located at each image. Either way, the periodic tensor is written in Fourier space as

$$\tilde{\boldsymbol{\mathcal{G}}}_F^{P}(\mathbf{k};\lambda_{\max}) = \left(\frac{2\pi}{L}\right)^3 \text{Ш}(\mathbf{k})\;\tilde{\boldsymbol{\mathcal{G}}}_w(\mathbf{k};\lambda_{\max},\infty); \tag{3.48}$$

$$\text{Ш}(\mathbf{k}) \equiv \sum_{\boldsymbol{\ell}} \delta^3(\mathbf{k}-\mathbf{k}_{\boldsymbol{\ell}}), \tag{3.49}$$

with $\mathbf{k}_{\boldsymbol{\ell}} = (2\pi/L)(\ell_x,\ell_y,\ell_z)$ and all $\ell\in\mathbb{Z}$.

For use in simulations these must be re-written in position space, but unlike the previous section where this was done to see the functional form, the interest here is to reach an integral form analogous to the integral over wavelets in Eq. (3.12) and hence a form amenable to application in Monte Carlo simulations. As before this requires an integral of a weighted, symmetric product of displacements.

The tensor structure can be written in a symmetric from by introducing $\hat{\mathbf{e}}\perp\hat{\mathbf{k}}$, whose average over the 2π allowed directions is $\langle\hat{\mathbf{e}}\otimes\hat{\mathbf{e}}\rangle_{\hat{\mathbf{e}}} = (\mathbf{I}-\hat{\mathbf{k}}\otimes\hat{\mathbf{k}})/2$. Similarly the complex exponential in the inverse Fourier transform can be re-expressed by noticing $\tilde{\boldsymbol{\mathcal{G}}}_w(\mathbf{k};\lambda_{\max},\infty)$ is even in $\mathbf{k}$. Then only the real cosine part contributes, and this can be written as the average over phase shift Φ:

$$\cos(\mathbf{k}\cdot\mathbf{r}_{ij}) = 2\left\langle\cos(\mathbf{k}\cdot\mathbf{r}_i+\Phi)\cos(\mathbf{k}\cdot\mathbf{r}_j+\Phi)\right\rangle_{\Phi}. \tag{3.50}$$

Finally, by introducing a move amplitude A_F the Fourier tensor with either boundary condition is expressed as

$$\boxed{\begin{aligned}\boldsymbol{\mathcal{G}}_F(\mathbf{r}_{ij}) &= \int d\boldsymbol{\Lambda}_F\,\mathcal{P}_{\mathbf{k}}\,\mathcal{P}_{\hat{\mathbf{e}}}\,\mathcal{P}_{\Phi}\,A_F\cos(\mathbf{k}\cdot\mathbf{r}_i+\Phi)\hat{\mathbf{e}}\otimes A_F\cos(\mathbf{k}\cdot\mathbf{r}_j+\Phi)\hat{\mathbf{e}}\\ &= \left\langle A_F\cos(\mathbf{k}\cdot\mathbf{r}_i+\Phi)\hat{\mathbf{e}}\otimes A_F\cos(\mathbf{k}\cdot\mathbf{r}_j+\Phi)\hat{\mathbf{e}}\right\rangle_{\boldsymbol{\Lambda}_F},\end{aligned}} \tag{3.51}$$

with $\boldsymbol{\Lambda}_F$ shorthand for all of $\mathbf{k}$, $\hat{\mathbf{e}}$ and Φ.

The Monte Carlo moves identified in Eq. (3.51) are $\delta\mathbf{r} = A_F\cos(\mathbf{k}\cdot\mathbf{r}+\Phi)\hat{\mathbf{e}}$, displacing *all* N particles simultaneously. As with the wavelet moves, the task now

is to determine all the parameter PDFs and A_F.

3.4 Fourier parameter distributions

$\mathcal{P}_{\hat{\mathbf{e}}}$ and $\mathcal{P}_\Phi$ are taken directly from the previous section, namely

$$\boxed{\mathcal{P}_{\hat{\mathbf{e}}} = \mathcal{P}_\Phi = \frac{1}{2\pi}.} \tag{3.52}$$

Though there is some freedom to modify these as long as any changes are compensated for in Eq. (3.51), continuing the assumption that the system has no inherent inhomogeneity or anisotropy makes this a complication with no real gain.

3.4.1 Fourier amplitude

The form of A_F is found in the same way as A_w in Section 3.2.2 by seeking a move energy independent of the move parameters. Happily the calculation is much simpler than it was for wavelets, with the strain tensor

$$\boldsymbol{\varepsilon} = \frac{1}{2} A_F \sin(\mathbf{k}\cdot\mathbf{r} + \Phi)\left[\mathbf{k}\otimes\hat{\mathbf{e}} + \hat{\mathbf{e}}\otimes\mathbf{k}\right] \tag{3.53}$$

leading to

$$\begin{aligned} \Delta U_{\text{est}} &= \frac{1}{2} U_0 \frac{N}{V} A_F^2 k^2 \int_V d^3\mathbf{r} \sin^2(\mathbf{k}\cdot\mathbf{r} + \Phi) \\ &= \frac{1}{4} U_0 N A_F^2 k^2, \end{aligned} \tag{3.54}$$

with V a volume over which the plane wave is periodic (evidently $V = L^3$ would suffice for periodic systems but in general it will not for unbounded systems).

Further to this, the condition of constant ΔU_{est} should be extended to all moves, both plane waves and wavelets alike, rather than just the parameters within each move type. Equating equations (3.30) and (3.54) sets

$$\boxed{A_F = 2\sqrt{\frac{M_6}{(2\pi)^3 N}}\, A_0 k^{-1}.} \tag{3.55}$$

To check whether this does indeed lead to a constant ΔU across all moves (allowing for the similar variation as observed in wavelets), Fig. 3.4 shows the move acceptance spectrum against k, including the asymptotic constant from the wavelet data in Fig. 3.2. This is seen to be successful with long wavelength (low k) modes, where P_{acc} is flat and matches the wavelet values for the same systems. As k increases however, P_{acc} rises to 1 much as it did for small wavelets. The reason for

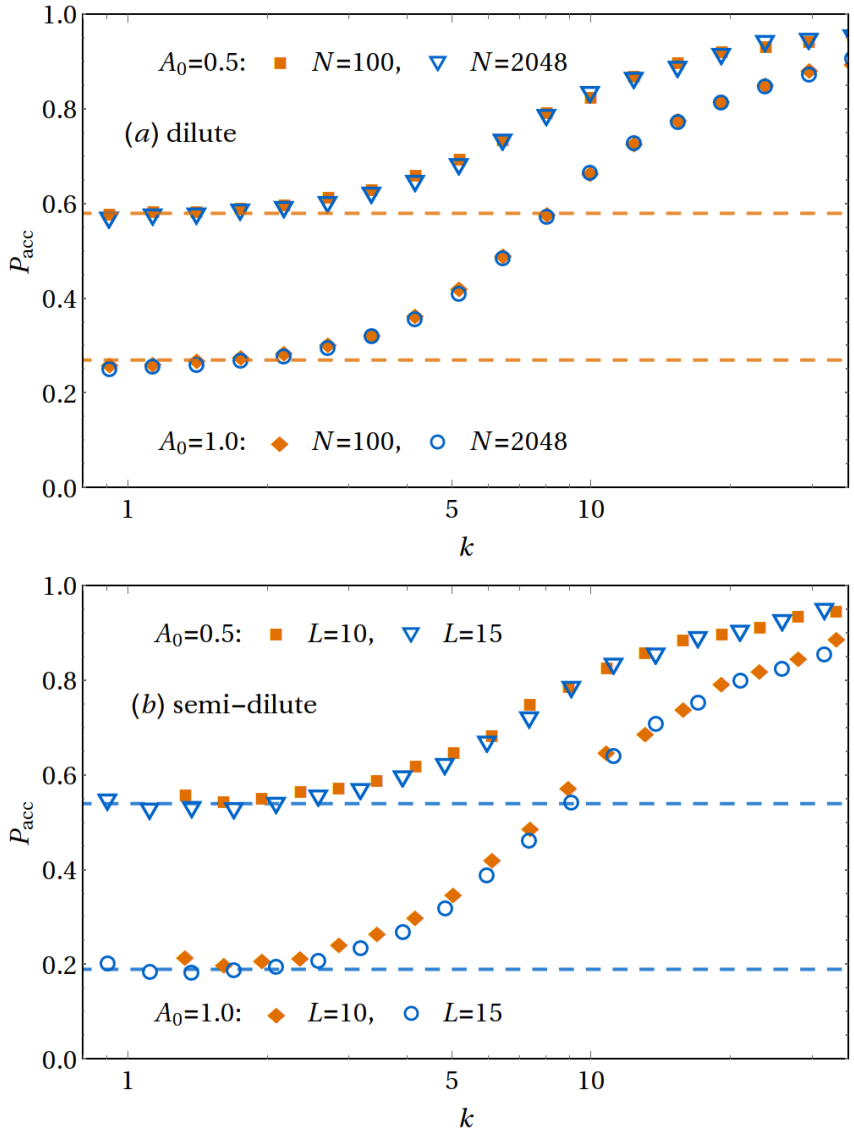

Figure 3.4: Acceptance probabilities over the spectrum of plane waves at different move amplitudes. (a): Single chain data in unbounded systems. (b): Data for semi-dilute periodic systems. All system parameters are identical to those used for Fig. 3.2, as are the dashed lines indicating the asymptotic wavelet values of P_{acc}.

this is essentially the same, with energy changes in short wavelength moves being more sensitive to $\delta\mathbf{r}$ than $\nabla(\delta\mathbf{r})$ and the strain calculation failing as a result. Unlike the wavelets, the dominant Fourier moves (at least by quantity) will be in the regime where the strain estimate is successful so this is less of a concern. That said, the same fixes discussed for wavelets can be applied to Fourier moves.

The divergence of A_F at $k = 0$ also needs addressing, but this is deferred to the next section as $\mathcal{P}_\mathbf{k}$ plays an important role.

3.4.2 Wavevectors

All that remains is to find $\mathcal{P}_\mathbf{k}$, and this is where the boundary conditions matter. In unbounded systems this just requires collecting the factors in Eq. (3.51) not yet accounted for. The isotropy of these systems means $\mathcal{P}^\infty_\mathbf{k}$ should only depend on the radial part of $\mathbf{k}$, leading to

$$\boxed{\mathcal{P}^\infty_\mathbf{k}(k) = \mathcal{N}^\infty_\mathbf{k} k^2 \int_{k\lambda_{\max}}^{\infty} dq\, q^3 \tilde{\phi}(q)^2} \tag{3.56}$$

once the factor of k^2 in $d^3\mathbf{k}$ is included. In this case the normalisation has the simple analytic form

$$\mathcal{N}^\infty_\mathbf{k} = \frac{3\lambda^3_{\max}}{4\pi M_6}. \tag{3.57}$$

In periodic systems the Dirac comb leads to the discrete set of probabilities

$$\boxed{P^P_\mathbf{k}(\mathbf{k}_{\boldsymbol{\ell}}) = \mathcal{N}^P_\mathbf{k} \int_{|\mathbf{k}_{\boldsymbol{\ell}}|\lambda_{\max}}^{\infty} dq\, q^3 \tilde{\phi}(q)^2,} \tag{3.58}$$

where the normalisation, obtained by a sum of this integral over all $\mathbf{k}_{\boldsymbol{\ell}}$, has no simple representation.

The wavelet dependence of both $\mathcal{P}^\infty_\mathbf{k}$ and $P^P_\mathbf{k}$ makes it difficult to make general statements about their functional form. Section 4.4.4 will return to this. Without going into the details, it will be seen that the limiting behaviour of the integral in Eq.s (3.56) and (3.58) is: $\sim (k\lambda_{\max})^0$ as $k \to 0$; and $\sim (k\lambda_{\max})^{-x}$ with $x \geqslant 4$ for large k (at least for the mother wavelets considered). The rapid decay in the large k regime arises from the wavelet moves already supplying most of the high frequency (short-ranged) interactions. The upshot is that the total probability converges quickly with increasing k. This is especially important in periodic simulations as computationally the last thing you want to do is have to pick from a huge set of discrete states. To ensure it decays as fast as possible $\lambda_{\max}$ should be made as large as possible, and hence as close to $L/2$ as the implementation

allows.

The small k limit is important because $A_F \sim k^{-1}$. In unbounded systems $\mathcal{P}^{\infty}_{\mathbf{k}} \sim k^2$ here and there is zero chance of an infinite amplitude $k = 0$ move. However it is worth flagging up that large amplitude, low frequency moves are possible, albeit rare. These are *not* a problem, though they might look concerning when first encountered.

In periodic systems things are reversed with the low frequency moves absent and the $k = 0$ mode present with finite probability. These moves cannot be left in with infinite amplitude, but using finite amplitudes and a fixed direction can set a reference frame. Working in the centre of mass frame, as this thesis always will, amounts to setting $A_F(k = 0) = 0$, and doing so is consistent with their removal from the Ewald sum in Brownian dynamics simulations [82]. Note that setting the amplitude to zero is not equivalent to removing them outright since that would change the probabilities of all the other Fourier moves.

3.5 Time evolution and the probability of making a Fourier move

Now the wavelet and Fourier parts have each been described, they must be tied together by determining the probability of making a Fourier move, P_F, as opposed to a wavelet ($P_w = 1 - P_F$). This task is inextricably linked to the time evolution in WMCD, so they are treated together here.

What allows WMCD to attach a physical time to its results, and hence what makes it a dynamic Monte Carlo method, is the knowledge of how much an isolated particle of hydrodynamic radius a diffuses in a physical time t. That is, it is known that

$$\mathrm{Var}(\delta\mathbf{r}) = \langle \delta r^2 \rangle - \langle \delta \mathbf{r} \rangle^2 = 6D\delta t = \frac{k_B T}{\pi \eta a} \delta t, \tag{3.59}$$

so computing $\mathrm{Var}(\delta\mathbf{r})$ for moves in WMCD gives δt. $\langle \delta \mathbf{r} \rangle$ depends on local configuration and cannot be incorporated into a calculation a priori, but nor does it need to be because WMCD as presented so far is constructed to only supply the Langevin noise. For now $\langle \delta \mathbf{r} \rangle$ only comes about through the acceptance test so the following calculations are free to ignore it altogether.

Anyhow, the mean square displacement of any given particle in a single

wavelet move is found from

$$\begin{aligned}\langle \delta r_i^2 \rangle_w &= \int d\mathbf{\Lambda}\, \mathcal{P}_\lambda\, \mathcal{P}_{\mathbf{b}|\lambda}\, \mathcal{P}_{\hat{\mathbf{p}}}\, A_w^2 \mathbf{w}\left(\frac{\mathbf{r}_i - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \cdot \mathbf{w}\left(\frac{\mathbf{r}_i - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \\ &= \frac{3A_0^2 \mathcal{N}_\lambda}{(4\pi)^2 N} \int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda \lambda^{-5} \oint d^2\hat{\mathbf{p}} \int_{-\infty}^{\infty} d^3\mathbf{x} \lambda^3\, \mathbf{w}(\mathbf{x}; \hat{\mathbf{p}}) \cdot \mathbf{w}(\mathbf{x}; \hat{\mathbf{p}}), \end{aligned} \tag{3.60}$$

with the substitution $\mathbf{x} = (\mathbf{r} - \mathbf{b})/\lambda$ made. The integral over $\mathbf{x}$ follows the same procedure as the wavelet strain energy integral, giving

$$\langle \delta r_i^2 \rangle_w = \frac{A_0^2 M_4}{(2\pi)^4 N} \mathcal{N}_\lambda \int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda \lambda^{-2} \oint d^2\hat{\mathbf{p}}. \tag{3.61}$$

Performing the remaining integrals gives

$$\langle \delta r_i^2 \rangle_w = \frac{2A_0^2 M_4}{(2\pi)^3 N} \mathcal{N}_\lambda \left(\lambda_{\min}^{-1} - \lambda_{\max}^{-1}\right). \tag{3.62}$$

The equivalent calculation for unbounded Fourier moves has

$$\begin{aligned}\langle \delta r_i^2 \rangle_F^\infty &= \int d\mathbf{\Lambda}_F\, \mathcal{P}_{\mathbf{k}}\, \mathcal{P}_\Phi\, \mathcal{P}_{\hat{\mathbf{e}}}\, A_F^2 \cos^2(\mathbf{k} \cdot \mathbf{r} + \Phi)\, \hat{\mathbf{e}} \cdot \hat{\mathbf{e}} \\ &= \frac{4\pi}{2} \mathcal{N}_{\mathbf{k}}^\infty \int_0^\infty dk k^2 A_F(k)^2 \int_{k\lambda_{\max}}^\infty dq\, q^3 \tilde{\phi}(q)^2. \end{aligned} \tag{3.63}$$

Commuting the integrals and substituting in A_F and $\mathcal{N}_{\mathbf{k}}^\infty$ from Eq.s (3.55) and (3.57) leads to

$$\langle \delta r_i^2 \rangle_F^\infty = \frac{2A_0^2 M_4}{(2\pi)^3 N} \frac{3\lambda_{\max}^3}{\lambda_{\max}}, \tag{3.64}$$

which upon inspection is seen to be exactly the same as Eq. (3.62) with $\lambda_{\max} \to \infty$ and $\lambda_{\min} \to \lambda_{\max}$, meaning the Fourier moves diffuse particles by exactly the same amount as the missing wavelets with $\lambda > \lambda_{\max}$.

At first this might not seem surprising; the Fourier moves were added precisely to replace these wavelets after all. But through A_F and the PDF normalisations several constants had been added between Eq. (3.47) and the calculation above, so you might expect to need to correct for a constant between the Fourier and wavelet contributions. It turns out constants were matched between the two moves when their strain energies were matched in Eq. (3.55), so any factors added by the PDF normalisations were reversed in A_F. Mathematically, the strain energy and mean square displacements can be viewed as operators acting on $\langle \delta\mathbf{r} \otimes \delta\mathbf{r} \rangle$, so fixing one of these to be equal also fixes the other.

The bottom line is that in unbounded systems the probability of picking a Fourier move is equal to the probability of picking a wavelet with $\lambda > \lambda_{\max}$, and hence

$$\boxed{P_F^\infty = \frac{\lambda_{\min}^3}{\lambda_{\max}^3}.} \tag{3.65}$$

The story in periodic systems is made complicated by integrals for wavelets, be it for the strain energy or $\langle \delta r_i^2 \rangle$, being somewhere between cumbersome and intractable for $\lambda > L/2$. Certainly the estimated strain energy of wavelet moves is invalid for these wavelets, and therefore the above argument about matching strain energy also matching mean-square displacements does not hold. Fortunately it is already known what periodic tensor (constants and all) these wavelets would construct and how it differs from the periodic Fourier tensor, because of the starting points in Eq.s (3.47) and (3.48) where the unbounded and periodic cases differ only by the convolution. From there the periodic wavelet tensor would pick up the same constants as the unbounded Fourier tensor does, leading to the ratio of periodic wavelet and Fourier tensors being equal to the ratio of constants accumulated by the unbounded and periodic Fourier integrals:

$$\mathcal{R} = \left(\frac{L}{2\pi}\right)^3 \frac{\mathcal{N}_{\mathbf{k}}^P}{\mathcal{N}_{\mathbf{k}}^\infty} = \frac{1}{6\pi^2}\left(\frac{L}{\lambda_{\max}}\right)^3 M_6 \mathcal{N}_{\mathbf{k}}^P. \tag{3.66}$$

To use this to calculate P_F^P imagine that in the pure wavelet case (with $\lambda_{\max} \to \infty$) X moves are needed to diffuse the system some amount, and in the mixed wavelet and Fourier case $X(1+\Delta)$ moves are needed for the same diffusion. Requiring the number of $\lambda < \lambda_{\max}$ moves to be the same in both cases yields the condition

$$(1 - P_F^P)X(1+\Delta) = (1 - (\lambda_{\min}/\lambda_{\max})^3)X, \tag{3.67}$$

while enforcing equal time evolution for large wavelets and plane waves requires

$$\mathcal{R}P_F^P X(1+\Delta) = (\lambda_{\min}/\lambda_{\max})^3 X. \tag{3.68}$$

Clearly X cancels allowing these equations to be solved for P_F^P, resulting in

$$\boxed{P_F^P = \left(1 + \mathcal{R}\left[\frac{\lambda_{\max}^3}{\lambda_{\min}^3} - 1\right]\right)^{-1}.} \tag{3.69}$$

So far this section has held off from writing down the time increments because it is worth handling all of the wavelet, unbounded Fourier and periodic Fourier moves together. Rather conveniently the appropriate time increment is actually the same for all three.

Returning to Eq. (3.59), which describes the diffusion of a *non-periodic* particle, and using the result that the unbounded Fourier moves contribute the same diffusion as missing large wavelets would, we can write the following for the full set of unbounded WMCD moves:

$$\langle \delta r_i^2 \rangle^\infty = \frac{6A_0^2 M_4 \lambda_{\min}^2}{(2\pi)^3 N} = \frac{k_B T}{\pi \eta a} \delta t; \tag{3.70}$$

$$\Rightarrow \boxed{\delta t = \frac{6A_0^2 M_4 \lambda_a^2}{(2\pi)^3 N} \frac{\pi \eta a^3}{k_B T}} = \frac{6A_0^2 M_4 \lambda_a^2}{(2\pi)^3 N} \tau, \tag{3.71}$$

where

$$\tau = \pi \eta a^3 / k_B T \tag{3.72}$$

is the time it takes for an isolated particle to diffuse by its own radius. τ will be used to non-dimensionalise time throughout the later chapters.

Eq. (3.71) also applies in periodic systems. Perhaps the easiest way to see this is to recognise that wavelets do not change between unbounded and periodic systems, so must increment time in the same way. While the Fourier part does change, and ultimately leads to a reduced diffusion if periodic moves are given the same time increment, this is the physical effect of the hydrodynamic interactions between the each particle and their periodic images.

3.5.1 Validation of time in WMCD

Three simple tests are presented here to validate the time increment in Eq. (3.71). These use polymer chains in good solvent, using the WCA and FENE potentials presented in Section 2.4.2. The parameters were set to $\sigma = 2^{-1/6}$, $\epsilon = 1$, $k_{\text{FENE}} = 7 \times 2^{1/3}$ and $R_0 = 2 \times 2^{-1/6}$ to match systems in Ref.s [34] and [38]. Other parameters were matched by setting $k_B T = 1.2$ and $\lambda_{\min} = 0.700$ (using the cubic wavelet, so this corresponds to the bead hydrodynamic radius $a = 0.302$).

Implicit in all of these are the code units for length and energy, set by $2^{1/6}\sigma$ and ϵ respectively. The time unit is defined by Eq. (3.72).

The effect of changing A_0: To test the role of A_0 it is noted that physics should not depend on the size of the time step, at least up to some degree of accuracy. Changing A_0 should therefore lead to identical dynamics, even while scaling the size of particle displacements. Fig. 3.5 shows this in conjunction with the relaxation of the squared radius of gyration

$$\left\langle R_g^2 \right\rangle = \frac{1}{2N^2} \sum_{i,j=1}^{N} \left\langle r_{ij}^2 \right\rangle \tag{3.73}$$

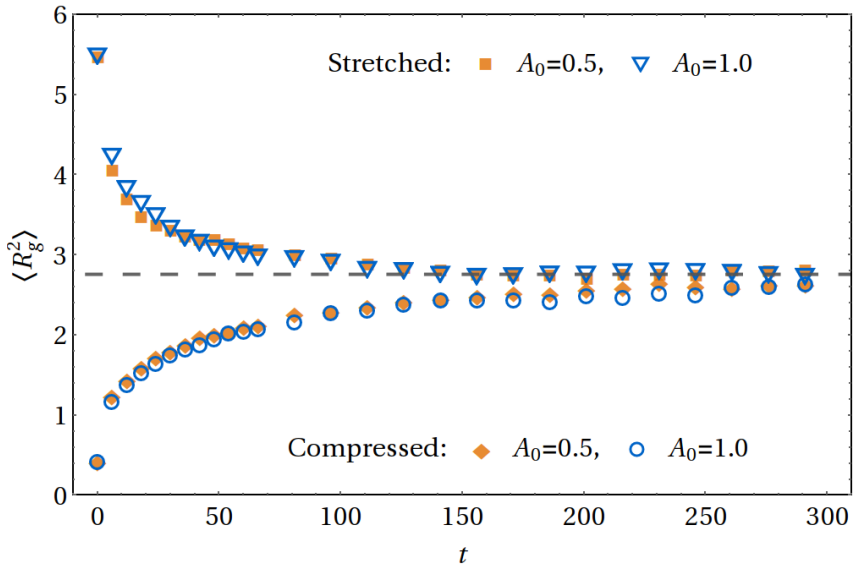

Figure 3.5: Relaxation of an isolated polymer 10 beads long with WCA and FENE potentials, for different A_0 and initial bond lengths. The dashed line shows the value for the same system in Ref. [34].

of short polymer chains. Evidently the value of A_0 has no impact on the relaxation time from either the stretched or compressed initial states, confirming $\delta t \propto A_0^2$ else the time axis would have been rescaled between the $A_0 = 0.5$ and $A_0 = 1.0$ data.

As an added (and reassuring) bonus Fig. 3.5 shows the radius of gyration to always relax to the equilibrium value obtained by Jain *et al.* for the equivalent system [34], confirming WMCD is correctly equilibrating the polymers.

The absolute value of δt: Unfortunately Ref. [34] provides no values of relaxation time to compare to those observed is Fig. 3.5, while equilibrium properties are static and provide no indication of whether δt is correct.

Polymer centre of mass diffusivity provides a convenient test of time because it can be measured both dynamically with $\langle \Delta R^2 \rangle / 6t$ to reasonable accuracy, and through the configurational average of the Kirkwood value [120]. Chapter 5 will show the full story of polymer diffusivity is deep and complex but for now it suffices to say the dynamic and static values should be in close agreement, and since only the dynamic value has knowledge of time this will confirm there are no errant factors in Eq. (3.71). Indeed this is what Fig. 3.6 shows with, once again, an added bonus in the agreement with the datum from Ref. [38].

δt in periodic systems: So far Eq. (3.71) has been validated in unbounded systems, and now the claim it applies without modification to periodic systems is tested. For this, the effect of the periodicity on single particle diffusivity is considered and

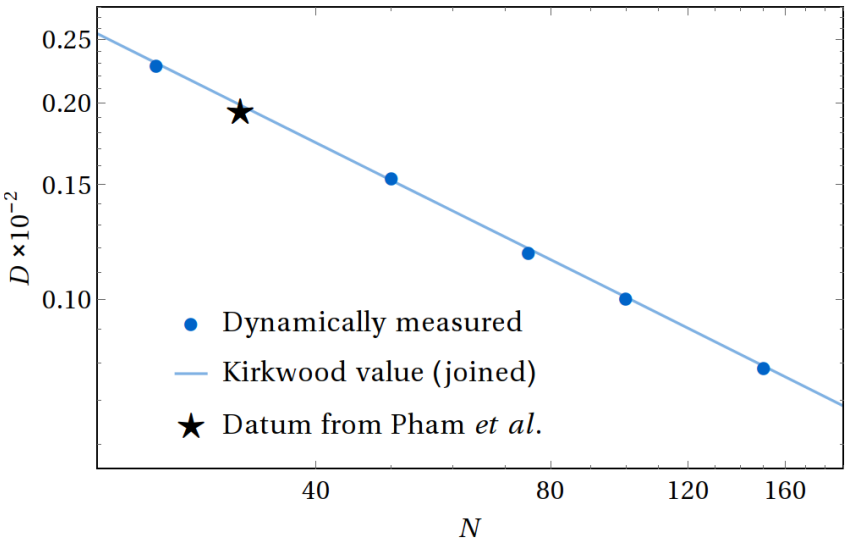

Figure 3.6: Centre of mass diffusivities of isolated polymer chains in unbounded WMCD. The Kirkwood data are joined to most clearly show both the trend and the proximity of the datum from Ref. [38] (star).

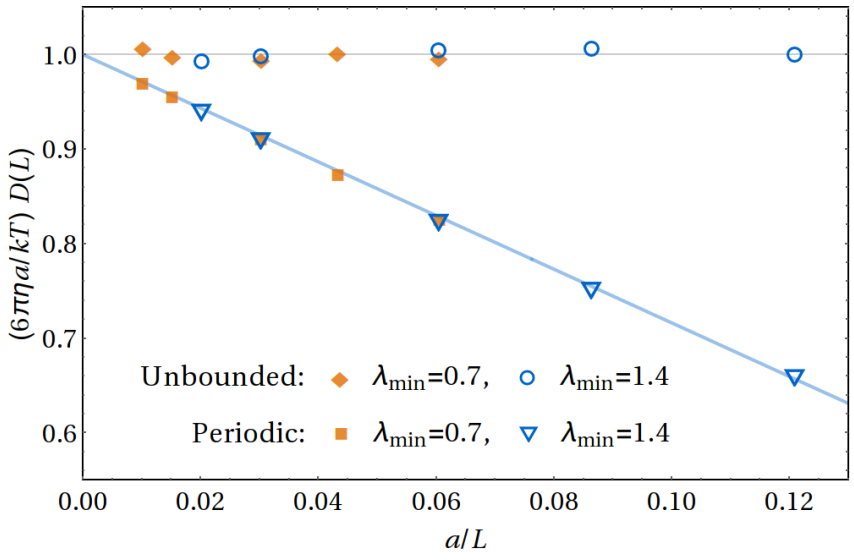

Figure 3.7: Single particle diffusivities measured in both unbounded and periodic WMCD, normalised by the theoretical unbounded value. The solid line plots the periodic value in Eq. (3.74).

compared to Eq. (2.34), so the diffusivity should depend on the box size as

$$D(L)/D(\infty) = \frac{6\pi\eta a}{k_B T} D(L) = \left(1 - 2.837\frac{a}{L}\right). \tag{3.74}$$

This comparison is made for two particle sizes in Fig. 3.7, where both are observed to match expectation, while the diffusivity in the unbounded simulations is correctly seen to be completely independent of L.

3.6 Schematic of the WMCD algorithm

With the distributions of move parameters known from the previous sections, the algorithm structure needs laying out to make use of them. Focussing only on the main loop of the algorithm and omitting implementation-dependent details such as *how* to identify movers, the algorithm is as follows:

Choose between wavelet and Fourier move types.

Wavelet	Fourier
Generate $\mathbf{b}$ and λ	Generate $\mathbf{k}$ and Φ.
Identify movers	Set all particles as movers.

Sum mover energies before the move.

Generate A_w and $\hat{\mathbf{p}}$	Generate A_F and $\hat{\mathbf{e}}$.

Displace all movers with the generated move.

Sum mover energies after the move.

Accept or reject move with a MC test using the total change in energy.
- If accepted, increment time by δt and keep updated mover positions.
- If rejected, return movers to their previous positions.

Repeat steps above for the next move.

With the WMCD algorithm as presented thus far, there is some freedom on when to generate A_w and $\hat{\mathbf{p}}$ for wavelets or A_F and $\hat{\mathbf{e}}$ for plane waves, as none of these require knowledge of the initial energy. The order above anticipates the modifications for smart MC in the next chapter where these parameters will be biased by the forces acting the movers, which are most efficiently calculated alongside

the energies. Note also the time is updated *only after accepted moves*, which over many moves is equivalent to rescaling the time by P_{acc} each move to match the diffusion prescibed by the Langevin equation as discussed in Section 2.6.2.

It is also worth explicitly stating how to generate the parameters. All of $\Phi, \hat{\mathbf{p}}$ and $\hat{\mathbf{e}}$ have uniform distributions so are trivial to obtain (allowing for a bit of care to ensure $\hat{\mathbf{e}}$ is perpendicular to $\mathbf{k}$), while A_w and A_F are simple algebraic functions of other parameters and need no explanation.

Choosing λ: λ can be obtained with the inverse transform method to convert a uniformly distributed random number $\texttt{rand} \in [0,1]$ to $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ via

$$\lambda = \lambda_{\min}\left(1 + \texttt{rand}\left[(\lambda_{\min}/\lambda_{\max})^3 - 1\right]\right)^{-1/3}. \tag{3.75}$$

Choosing b: Generating $\mathbf{b}$ with the distribution in Eq. (3.22) is actually very easy, if a little obscure. $\mathcal{P}_{\mathbf{b}|\lambda}$ will come about automatically in a simulation if using the procedure:

> Choose a particle i with uniform probability.
>
> Place $\mathbf{b}$ with uniform probability in a sphere of radius λ centred on $\mathbf{r}_i$.

The simulation will not actually need to be told the form of $\mathcal{P}_{\mathbf{b}|\lambda}$.

Choosing k in an unbounded system: That leaves the wavevector $\mathbf{k}$, which is awkward even without the split into unbounded and periodic versions because it depends on the choice of mother wavelet. In unbounded systems $\hat{\mathbf{k}}$ is uniformly distributed over all directions and is easy to generate, but for $k = |\mathbf{k}|$ the easiest approach is to externally generate a text file of the cumulative distribution function for $\lambda_{\max}^2 \mathcal{P}_{\mathbf{k}}^{\infty}$ with which to use the inverse transform method numerically. Each mother wavelet needs its own CDF, but multiplying by $\lambda_{\max}^2$ allows $k\lambda_{\max}$ to be grouped into a single variable so a single file can cover all possible values of $\lambda_{\max}$ (it will of course require careful unpacking in the code to get k by itself).

Choosing k in a periodic system: The grouping of $k\lambda_{\max}$ carries over to periodic systems but the similarities stop there because the discrete modes are dependent on the box size L, so a different file would be needed for every value ever used. With those prospects it is better to calculate probabilities directly in the code, for which it is fortunate high frequency moves can be truncated off and a Taylor series for $P_{\mathbf{k}}^P(k\lambda_{\max})$ can replace the integral. Admittedly not a low order Taylor series - the series used in this work had terms upwards of order $(k\lambda_{\max})^{26}$ - but it still beats numerical integration. Then the task is to cycle through discrete wavevectors, each generated uniformly below the truncation, until $\texttt{rand} < P_{\mathbf{k}}^P(k\lambda_{\max})/P_{\mathbf{k}}^P(0)$,

exploiting the fact the $k = 0$ mode has the highest probability[4].

3.7 Computational cost

With the WMCD algorithm set out, attention is now turned to calculating its computational cost C and how it compares to the established algorithms discussed in previous chapters. Note 'cost' and 'complexity' are used more or less interchangeably, with the former preferred for the sake of brevity.

3.7.1 Computational cost of Fourier moves

First consider the Fourier moves, which will prove to be the simpler calculation and serve to illustrate the procedure. A useful starting point is to consider a pure Fourier simulation, and then adapt the calculation to account for the presence of wavelets. In this case the simulation makes a series of moves that displace all N particles, and hence each move carries a cost of N multiplied by an average cost per particle.

Since what is really sought is how long in real time it is going to take to run a simulation of some number of time units, the time evolution per move must also enter the story. Bringing in a factor of $\delta t^{-1} \sim N$ from Eq. (3.71) leads to the cost per time unit of

$$\frac{dC_F^{\text{pure}}}{dt} \propto \frac{N^2}{A_0^2 \langle P_{\text{acc}} \rangle}. \tag{3.76}$$

Here only the factors that you have control over (without a major change to the code) have been kept.

The factor of A_0^{-2} comes directly from δt and emphasises its role in balancing accuracy and computational efficiency, while $\langle P_{\text{acc}} \rangle^{-1}$ originates from updating time only after accepted moves and only introduces a factor of order 1 (it is included mostly because of its dependence on A_0). The take-home message, however, is that the cost of a pure Fourier algorithm scales as N^2.

If wavelets are included, an additional factor of P_F is also present. In periodic systems with P_F^P given by Eq. (3.69) this is a bit messy, but in practice $P_F^P \approx P_F^\infty$. λ_{min} has no link to the system size, so the focus is on the factor of $\lambda_{\text{max}}^{-3}$. In periodic systems with fixed particle density $\lambda_{\text{max}}^{-3} \sim L^{-3} \sim N^{-1}$ and hence

$$\frac{dC_F^P}{dt} \sim N. \tag{3.77}$$

In unbounded systems the global density is zero and this argument needs modifying. By considering the N particles as a cluster and increasing λ_{max} with the cluster size,

[4] If it lands on $k = 0$, remember to set $A_F = 0$. Having done this the move obviously does nothing and it is most efficient to end the move there *after incrementing time*.

e.g. $\lambda_{\max} \sim R_g \sim N^{\nu}$ for a single polymer, the cost of Fourier moves will increase no slower than linearly.

$\lambda_{\max} = L/2$ was assumed above and it is worth considering what changes if a smaller value is chosen. This is clearest with unbounded systems because of the 1-to-1 replacement of wavelets to plane waves, but the conclusions are much the same in periodic systems. When swapping a wavelet for a plane wave, the main question to ask regarding their respective costs is how many particles they need to move; everything other than the sub-dominant cost of parameter generation is the same. There will never be more movers in a wavelet than its corresponding Fourier move, and generally there will be significantly fewer. It is therefore optimal to minimise the number of Fourier moves by choosing $\lambda_{\max} = L/2$.

3.7.2 Computational cost of wavelet moves

Moving on to the computational cost of wavelets, as just stated the main difference is the number of moving particles n. Certainly this depends on the wavelet radius, leading to an integral over λ, and unless the system is perfectly homogeneous it also depends on where the wavelet is located. Incorporating the location-dependence requires prior knowledge of the system configuration, and even then the calculation is only useful for that specific system. More helpful is to ignore the inhomogeneities and instead estimate the number of movers using the fractal dimension d_f of the system. Then

$$n(\lambda) \propto (\lambda/s)^{d_f} \tag{3.78}$$

with s the mean separation of neighbouring particles.

The cost of wavelet moves is then given by

$$\frac{dC_w}{dt} \propto \frac{N}{A_0^2 \langle P_{\text{acc}} \rangle} \int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda \, \mathcal{P}_\lambda(\lambda) \, n(\lambda). \tag{3.79}$$

To perform the integral it is necessary to distinguish between homogeneous ($d_f = 3$) and fractal ($d_f < 3$) systems, with each treated separately below.

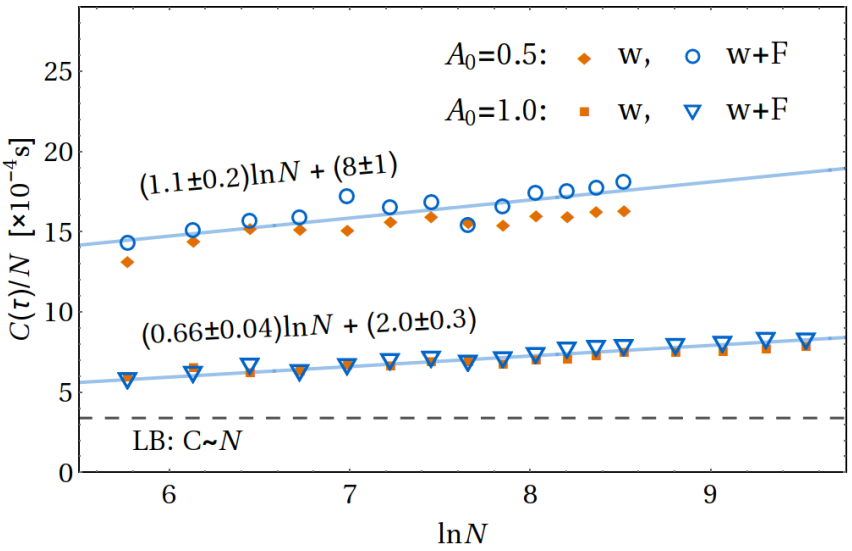

Figure 3.8: CPU cost per particle to evolve semi-dilute (homogeneous) systems by a single time unit. The dashed line indicates LB timings in identical systems from Fig. 8 in Ref. [34], rescaled to our time unit. Timings for both pure wavelet (w) and periodic wavelet plus Fourier (w+F) algorithms are shown for comparison.

Cost in homogeneous systems

When the system is homogeneous[5] the integrand goes as λ^{-1} and integrates to a logarithm. The nearest neighbour distance is $s^3 = L^3/N$, so

$$\begin{aligned} \frac{dC_w^{\text{homo}}}{dt} &\propto \frac{N^2}{A_0^2\langle P_{\text{acc}}\rangle}\left(\frac{\lambda_{\min}}{L}\right)^3 \frac{\ln(\lambda_{\max}/\lambda_{\min})}{1-\lambda_{\min}/\lambda_{\max}} \\ &\propto N\ln N \frac{\lambda_{\min}^3}{A_0^2\langle P_{\text{acc}}\rangle} \end{aligned} \tag{3.80}$$

with the second line using $\lambda_{\max} \sim L \sim N^{1/3}$ and $\lambda_{\max} \gg \lambda_{\min}$.

Fig. 3.8 shows cost timings in periodic semi-dilute polymer systems, confirming the $N \ln N$ scaling in both pure wavelet and full wavelet plus Fourier simulations. Also notable is the relatively small contribution to the cost from the Fourier moves, as might be expected when $\lambda_{\max} \gg \lambda_{\min}$ so $P_F \ll 1$.

The result of cost increasing with decreasing A_0 is also observed in Fig. 3.8. The cleanest verification of the A_0^{-2} factor is found in the constant terms in the fitted lines, which originate in the $\ln \lambda_{\min}$ term omitted in the second line of Eq. (3.80). The factor of $4 =$ (ratio of values of A_0^2) between these constants matches expectation.

Meanwhile, the ratio of the coefficients in front of the log term also needs to account for the change in $\langle P_{\text{acc}}\rangle^{-1}$ (see Fig. 3.2) because this term derives from

[5] An example of a system which is not homogeneous but has $d_f = 3$ is a polymer coil in poor solvent, where the cost calculation carries over.

$\lambda_{\max}$ so one would expect the associated change in cost to enter only in the large moves. Together, the ratio is

$$\frac{A_0^2\langle P_{\rm acc}\rangle|_{1.0}}{A_0^2\langle P_{\rm acc}\rangle|_{0.5}} \approx 1.5 \approx \frac{1.1 \pm 0.2}{0.66 \pm 0.004} \tag{3.81}$$

again agreeing with the factors included in Eq. (3.80).

Finally, Fig. 3.8 includes the cost in LB simulations of identical systems, which is a factor of order 1 smaller than the WMCD cost with $A_0 = 1.0$ for the range of systems included. These relative costs shouldn't be taken too seriously for several reasons, including the unoptimised version of WMCD used for this data having trivial (e.g. using a faster compiler) and involved (e.g. using smart Monte Carlo) options available to improve the cost, and the single system density considered only giving part of the picture. Regarding the latter, these systems were particularly dense for 'semi-dilute' owing to the use of chains only 10 beads long, so the system was inherently favourable to explicit solvent methods like LB. As chain lengths increase and the required density for semi-diluteness decreases, implicit solvent methods compare better (see the next subsection for the extreme case). Nevertheless, the key point is that a rudimentary implementation of WMCD is competitive even in dense systems[6].

Cost in fractal systems

While WMCD performs well in homogeneous systems, it performs even better in fractal ones with $d_f < 3$, where Eq. (3.79) integrates to

$$\begin{aligned} \frac{dC_w^{\rm frac}}{dt} &\propto \frac{N}{A_0^2\langle P_{\rm acc}\rangle}\left(\frac{\lambda_{\min}}{s}\right)^{d_f}\frac{1-(\lambda_{\min}/\lambda_{\max})^{3-d_f}}{1-\lambda_{\min}/\lambda_{\max}} \\ &\propto N\frac{\lambda_{\min}^{d_f}}{A_0^2\langle P_{\rm acc}\rangle}. \end{aligned} \tag{3.82}$$

This time s is taken as the mean bond length between beads and has no dependence on N.

Fig. 3.9 shows the cost in fractal systems and once again verifies the cost calculation and shows negligible contribution from Fourier moves. There is a slight increase in scaling power coming from the details of the implementation, but the real focus is on the comparison with the LB and BD costs. As before there is some freedom to shift these lines by modifying various details of the systems and algorithms, but clearly WMCD presents a good choice for large fractal systems and is able to go beyond the range of systems accessible to the established algorithms.

[6]Perhaps the best way to emphasise this is to point out the BD costs in these systems were omitted from Fig. 3.8 due to being several orders of magnitude larger [34].

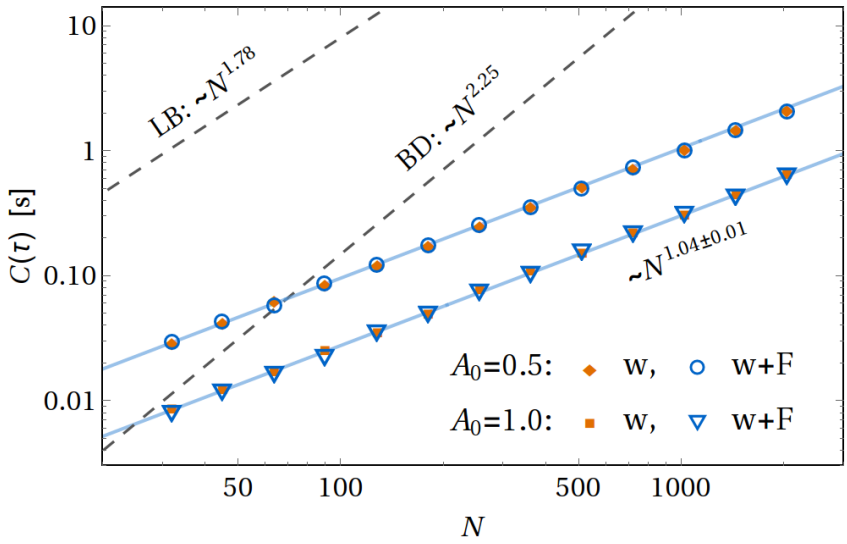

Figure 3.9: CPU cost to evolve systems with an isolated polymer of length N by a single time unit with the pure wavelet (w) and infinite wavelet plus Fourier (w+F) algorithms. The systems considered were identical to those in Fig. 11 in Ref. [38], and the dashed line indicates the BD and LB timings from that plot, rescaled to our time unit.

3.8 Final remarks on the core WMCD algorithm

The account of WMCD presented in this chapter has attempted to be as complete as necessary for someone to be able to write their own simulation code without needing any additional calculations. The actual implementation of WMCD is much simpler than the fine details let on and it is easy to drown out the required parts in the sea of calculations. All one actually *needs* are the 14 boxed equations, which together amount to nothing more than carefully constructed wavelet and Fourier representations of tensors and a set of probabilities, PDFs and a time step. Poetically put: if the calculations are the sea, these are the islands shaped by it and most users of WMCD will not need to step foot off dry land.

Rather than summarising the key details again to conclude the chapter, it is more fruitful to shift the narrative from a computational to a physics viewpoint. Most of the algorithm is pure numerical trickery that can be hidden away inside a black box without hindering one's ability to interpret physical results. The choice of mother wavelet matters physically, and will be addressed once fully equipped to do so in Section 4.4, but even that only matters because of a single parameter: $\lambda_{\min}$.

It is difficult to overstate the importance of $\lambda_{\min}$ to WMCD; its presence reverberated through the calculations and will continue to do so in the next chapter. Eq. (3.45) ties it to the physical size of the simulated particles and deserves promoting to theorem status:

Theorem 1. *The hydrodynamic radius of a particle is proportional to the lower cut-off of wavelet radius, $\lambda_{\min} = \lambda_a a$, with the constant of proportionality dictated by the choice of mother wavelet.*

A much more subtle result that was not highlighted previously is that

Theorem 2. *The distribution of moves (and by extension the correlated motion) seen by a pair of positions separated by $r \geqslant 2\lambda_{\min}$ is identical to the distribution when $\lambda_{\min} = 0$.*

In other words the correlated motion of particles sufficiently separated will be exactly equal to that of the Oseen tensor. This has a particularly useful corollary:

Corollary 1. *The response of a particle to a force (either causal or random) a distance r away is identical for all WMCD simulations with $\lambda_{\min} < r/2$.*

At first these statements seem to show when $\lambda_{\min}$ does not matter, rather than emphasising its importance. To do that, their reverse needs stating: all deviations from the Oseen tensor occur at ranges shorter than $2\lambda_{\min}$. In the next chapter the importance of $\lambda_{\min}$ will grow yet further.

4

Further developments for WMCD

This chapter presents a sequence of extensions and adaptations to the WMCD algorithm. These will address limitations of WMCD as presented so far, namely:

- moves are chosen solely by the noise term in the Langevin equation with the causal terms left to the acceptance test, which can be inaccurate and expensive;
- all particles have the same hydrodynamic radius set by $\lambda_{\min}$;
- and there is no knowledge of orientations and rotational motion.

4.1 Smart Wavelet Monte Carlo dynamics

Addressing the first limitation, this section adapts WMCD to a smart Monte Carlo (SMC) algorithm, allowing WMCD to exploit the greater accuracy for a given time step demonstrated in Section 2.6.4. The details of wavelet and Fourier moves lead to superficial differences with the description of SMC in Section 2.6.3, but the underlying aims and methodology are much the same.

4.1.1 Smart wavelet moves

The goal for smart wavelets is to centre their move parameters on the net generalised force $\boldsymbol{\mathcal{F}}$ inside them. This needs the movers to be known beforehand, which in turn requires knowing λ and $\mathbf{b}$. Therefore these parameters cannot be biased and they keep their non-smart PDFs in Eq.s (3.22) and (3.34).

That leaves $\hat{\mathbf{p}}$ and A_w, which can be grouped into a single vector quantity $\mathbf{Q} = A_w\hat{\mathbf{p}}$ leading to an alternative expression for the wavelet in Eq. (3.14) as

$$A_w\mathbf{w}(\mathbf{r},\hat{\mathbf{p}}) = \begin{cases} \mathbf{Q}\times\nabla\phi(\mathbf{r}) & \text{for } |\mathbf{r}| \leqslant 1 \\ \mathbf{0} & \text{for } |\mathbf{r}| > 1 \end{cases} . \tag{4.1}$$

$\boldsymbol{\mathcal{F}}$ is then obtained as the gradient of

$$\begin{aligned} \Delta U &= -\sum_{i=1}^{n} \mathbf{F}_i \cdot (A_w\mathbf{w}_i) \\ &= -\sum_{i=1}^{n} (\nabla\phi_i \times \mathbf{F}_i)\cdot\mathbf{Q} \end{aligned} \tag{4.2}$$

with respect to $\mathbf{Q}$, giving

$$\boldsymbol{\mathcal{F}} \equiv -\nabla_{\mathbf{Q}}(\Delta U) = \sum_{i=1}^{n} \nabla\phi_i \times \mathbf{F}_i. \tag{4.3}$$

This then biases the distribution of $\mathbf{Q}$ via

$$\mathcal{P}_{\mathbf{Q}}(\mathbf{Q};\boldsymbol{\mathcal{F}}) = \frac{1}{(2\pi\sigma_{\mathbf{Q}}^2)^{3/2}} \exp\left(-\frac{(\mathbf{Q}-\beta D_{\mathbf{Q}}\delta t\,\boldsymbol{\mathcal{F}})^2}{2\sigma_{\mathbf{Q}}^2}\right). \tag{4.4}$$

The variance $\sigma_{\mathbf{Q}}^2$ must match the variance of the product $A_w\hat{\mathbf{p}}$ in non-smart WMCD else the noise will not be correctly supplied. A_w was set with no distribution in Eq. (3.31), so calculating $\sigma_{\mathbf{Q}}^2$ is a simple integral over the unit sphere giving

$$\sigma_{\mathbf{Q}}^2 = 2D_{\mathbf{Q}}\delta t = \frac{A_0^2\lambda^2}{3n}. \tag{4.5}$$

After this, all that needs changing from the non-smart algorithm is the acceptance test. Making the appropriate modifications to Eq. (2.49) leads to the replacement

$$-\beta\Delta U \to -\beta\Delta U - \frac{1}{2}\beta(\boldsymbol{\mathcal{F}}_{\text{fwd}} + \boldsymbol{\mathcal{F}}_{\text{rev}})\cdot\mathbf{Q} - \frac{1}{4}\beta^2 D_{\mathbf{Q}}\delta t(\boldsymbol{\mathcal{F}}_{\text{rev}}^2 - \boldsymbol{\mathcal{F}}_{\text{fwd}}^2) \tag{4.6}$$

in the Metropolis test, where $\mathbf{Q}_{\text{fwd}} = -\mathbf{Q}_{\text{rev}} = \mathbf{Q}$.

4.1.2 Smart Fourier moves

The smart version of Fourier moves follow an analogous procedure to the wavelets, this time with the biased parameter $\mathbf{Q}_F = A_F\hat{\mathbf{e}}$, while $\mathbf{k}$ and Φ are needed to

determine the generalised force. The energy change in the move is

$$\Delta U = -\sum_{i=1}^{N} \mathbf{F}_i \cdot \mathbf{Q}_F \cos(\mathbf{k}\cdot\mathbf{r} + \Phi), \tag{4.7}$$

which upon differentiation gives

$$\boldsymbol{\mathcal{F}}_F \equiv -\nabla_{\mathbf{Q}_F}(\Delta U) = \sum_{i=1}^{N} \mathbf{F}_i \cos(\mathbf{k}\cdot\mathbf{r} + \Phi). \tag{4.8}$$

Because $\hat{\mathbf{e}}$ is confined to the plane perpendicular to $\mathbf{k}$, the Gaussian from which $\mathbf{Q}_F$ should be generated should also be confined to this plane. However it is simpler to use the 3-dimensional Gaussian in Eq. (4.4) with the replacements $\mathbf{Q} \to \mathbf{Q}_F$, $\boldsymbol{\mathcal{F}} \to \boldsymbol{\mathcal{F}}_F$, and

$$\sigma^2_{\mathbf{Q}} \to \sigma^2_{\mathbf{Q}_F} = 2D_{\mathbf{Q}_F}\delta t = \frac{1}{2}\frac{A_0^2 M_6}{2\pi^3 N k^2} \tag{4.9}$$

(note the factor of 1/2 rather than 1/3 because it will end up 2-dimensional) before projecting $\mathbf{Q}_F$ onto the desired plane with

$$\mathbf{Q}_F \to \mathbf{Q}_F \cdot \left(\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}\right). \tag{4.10}$$

Finally, the change to the Metropolis test is just Eq. (4.6) with Fourier subscripts where appropriate.

4.1.3 Verifying hydrodynamics in smart WMCD

It is now shown that smart WMCD leads to the correct hydrodynamic coupling. Only the proof for wavelet moves is given as the process is easy to adapt to Fourier moves and it adds little insight to do so.

Starting from the overdamped Langevin equation in Eq. (2.15), each particle should feel a sum of hydrodynamically propagated forces and a random velocity. Using the linearity of the Stokes equation it is sufficient to verify the action of a single force. The focus therefore lies on the contribution of a single force acting on particle j

$$\boldsymbol{\mathcal{F}} = (\nabla\phi_j) \times \mathbf{F}_j \tag{4.11}$$

to any move containing it.

At particle i there should be a corresponding causal velocity

$$\begin{aligned} \mathbf{u}_i &= \boldsymbol{\mathcal{G}}_{ij} \cdot \mathbf{F}_j \\ &= \frac{1}{2k_B T \delta t}\left(\int d\boldsymbol{\Lambda} \mathcal{P}_\lambda \mathcal{P}_{\mathbf{b}|\lambda} \mathcal{P}_{\widehat{\mathbf{p}}}\, A_w \mathbf{w}_i \otimes A_w \mathbf{w}_j\right) \cdot \mathbf{F}_j \end{aligned} \tag{4.12}$$

when substituting in the wavelet integral for $\langle \delta\mathbf{r}_i \otimes \delta\mathbf{r}_j \rangle = 2k_BT\boldsymbol{\mathcal{G}}\delta t$. The λ and $\mathbf{b}$ integrals will not need performing, while A_w can be substituted for its form in Eq. (3.31) and $\hat{\mathbf{p}}$ can be integrated out using the usual wavelet form in Eq. (3.14) along with

$$\frac{1}{4\pi}\oint d\hat{\mathbf{p}}\,\hat{\mathbf{p}}\otimes\hat{\mathbf{p}} = \frac{1}{3}\mathbf{I}. \tag{4.13}$$

Carefully tracking tensor indices of what remains gives

$$\mathbf{u}_i = \frac{1}{6k_BT\delta t}\int d\lambda d\mathbf{b}\;\mathcal{P}_\lambda\mathcal{P}_{\mathbf{b}|\lambda}\frac{A_0^2\lambda^2}{n}\left((\nabla\phi_j)\times\mathbf{F}_j\right)\times(\nabla\phi_i) \tag{4.14}$$

in which $\boldsymbol{\mathcal{F}}$ is readily identified and can be replaced by

$$\boldsymbol{\mathcal{F}} = \frac{\langle\mathbf{Q}\rangle}{\beta D_{\mathbf{Q}}\delta t} = \frac{6k_BTn}{A_0^2\lambda^2}\int d\mathbf{Q}\,\mathcal{P}_{\mathbf{Q}}\mathbf{Q}. \tag{4.15}$$

Then the integral becomes

$$\begin{aligned}\mathbf{u}_i &= \frac{1}{\delta t}\int d\lambda d\mathbf{b} d\mathbf{Q}\,\mathcal{P}_\lambda\mathcal{P}_{\mathbf{b}|\lambda}\mathcal{P}_{\mathbf{Q}}\,\mathbf{Q}\times\nabla\phi_i \\ &= \frac{\langle A_w\mathbf{w}_i\rangle}{\delta t}.\end{aligned} \tag{4.16}$$

I.e. $\mathbf{u}_i$ is the mean velocity of particle i in a SMC move, exactly as sought.

4.2 Polydisperse WMCD

Everything in WMCD has thus far been appropriate for monodisperse systems, with every particle having the same hydrodynamic radius. A conceptually very simple way to extend to polydisperse systems is to allocate a value of $\lambda_{\min}$ to each particle individually. Many of the key results from the previous chapter then need to be generalised. The cleanest approach is to remove $\lambda_{\min}$ as a lower bound of λ and shift it into the definition of the wavelet vector with a Heaviside step function:

$$\mathbf{w}\left(\frac{\mathbf{r}_i-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right)\to\mathbf{w}\left(\frac{\mathbf{r}_i-\mathbf{b}}{\lambda};\hat{\mathbf{p}}\right)\Theta\left(\lambda-\lambda_i\right). \tag{4.17}$$

The particle i would then only be sensitive to wavelets with $\lambda \geqslant \lambda_i = \lambda_a a_i$ and the global $\lambda_{\min} = \min\{\lambda_i\}$. It will also be useful to define the fraction of particles sensitive to wavelets of a given radius

$$K(\lambda) = \frac{1}{N}\sum_{i=1}^{N}\Theta\left(\lambda-\lambda_i\right). \tag{4.18}$$

The move centring scheme of Section 3.2.1 wants to be made aware of this

to continue ensuring all moves make non-zero displacements of at least one particle. This can be achieved by only choosing particles to centre around from those with $\lambda_i \leqslant \lambda$, leading to $N \to NK(\lambda)$ and Eq. (3.22) becoming:

$$\mathcal{P}_{\mathbf{b}|\lambda} = \frac{3n}{4\pi\lambda^3 NK(\lambda)}. \tag{4.19}$$

By itself this leads to a relative decrease in large λ moves around small particles (and vice versa), so must be compensated for in the PDF for λ by setting

$$\mathcal{P}_\lambda = \mathcal{N}_\lambda \lambda^{-4} K(\lambda). \tag{4.20}$$

The new normalisation is then

$$\mathcal{N}_\lambda = \left(\int_0^\infty d\lambda \lambda^{-4} K(\lambda) \right)^{-1} \tag{4.21}$$

$$= 3N \left(\sum_{i=1}^{N} (\lambda_i^{-3} - \lambda_{\max}^{-3}) \right)^{-1}. \tag{4.22}$$

The time increment per move is calculated in the same way as before, just with the polydisperse PDFs. Because $K(\lambda)$ cancels between Eq.s (4.19) and (4.20) to leave behind the $N^{-1}\,\mathcal{P}_{\mathbf{b}|\lambda}$ originally contributed, only $\mathcal{N}_\lambda$ and the effective lower limit of the λ integral change. Assuming no particle is so large that $\lambda_i > \lambda_{\max}$, Fourier moves can be accounted for by setting $\lambda_{\max} \to \infty$ to give

$$\langle \delta r_i^2 \rangle^\infty = \frac{2M_4 A_0^2}{(2\pi)^3 N \lambda_i} \mathcal{N}_\lambda|_{\lambda_{\max}\to\infty} = \frac{k_B T}{\pi \eta a_i} \delta t \tag{4.23}$$

$$\Rightarrow \delta t = \frac{6\pi M_4}{(2\pi)^3} \frac{\eta A_0^2}{\lambda_a k_B T} \left(\sum_{i=1}^{N} \lambda_i^{-3} \right)^{-1}. \tag{4.24}$$

To factor out the time unit $\tau = k_B T / \pi \eta a^3$ one must first choose a value of a for this purpose, as it can no longer be tied to all particles simultaneously.

Finally, attention turns to the probability of making Fourier moves, which necessarily changes with the change to $\mathcal{N}_\lambda$. Owing to the 1-to-1 correspondence with wavelets, the probability in unbounded systems is

$$P_F^\infty = \mathcal{N}_\lambda|_{\lambda_{\max}\to\infty} \int_{\lambda_{\max}}^{\infty} d\lambda \lambda^{-4} K(\lambda) = N \lambda_{\max}^{-3} \left(\sum_{i=1}^{N} \lambda_i^{-3} \right)^{-1}. \tag{4.25}$$

Meanwhile, the argument for the probability in periodic systems does not change in

polydisperse systems, so the periodic probability is

$$P_F^P = \left(1 + \mathcal{R}\left[(P_F^\infty)^{-1} - 1\right]\right)^{-1}, \tag{4.26}$$

with $\mathcal{R}$ defined as in Eq. (3.66).

4.2.1 Summary of algorithmic procedure

The changes from Section 3.6 needed at an algorithmic level are now listed. The new forms of δt and P_F simply modify the values of constants with no real change as far as the procedure is concerned.

Choosing λ and b: The approach to choosing λ does not need to be made aware of the changes to its normalisation. In fact the polydisperse normalisation is only useful for obtaining δt and P_F analytically. However, the true $\mathcal{P}_\lambda$ is now position-dependent and care must be taken to ensure this is correctly represented in the simulation. Fortunately all that needs adding to the monodisperse algorithm is a while loop testing that the particle selected has $\lambda_i \leqslant \lambda$, leading to the procedure

```
While(λ_i > λ)
 | Choose a particle i.
 ⌊ Choose λ.

Centre b around r_i.
```

Within this the old methodology applies, including the inverse transform in (3.75).

Counting movers: Putting the Heaviside function in the wavelet raises the question of whether particles inside a wavelet with $\lambda_i > \lambda$ count as movers. Certainly getting $\mathcal{P}_{\mathbf{b}|\lambda}$ right requires them to not be counted given the changes to wavelet parameter generation just described, and therefore it is helpful to replace n in $\mathcal{P}_{\mathbf{b}|\lambda}$, $\sigma^2_{\mathbf{Q}}$ and $D_{\mathbf{Q}}$ with

$$nK_n(\lambda) = \sum_{i=1}^{n} \Theta\left(\lambda - \lambda_i\right), \tag{4.27}$$

if n is the total number of particles contained.

The question also extends to the move bias $\boldsymbol{\mathcal{F}}$. Whether this quantity sees all n particles or the subset nK_n affects the flow field local to each force as seen by particles smaller than the particle to which the force is associated (particles of equal or greater size are unaffected). The approach taken for work in this thesis has been to use all n forces, which equates to placing a full Stokeslet at each force which each particle will respond to according to their own size.

If instead only the nK_n forces are included, all smaller particles will respond to that force as though they are the same size as the associated particle, which would be consistent with how the cross-mobility manifests itself with the noise (see below).

4.2.2 Comments on accuracy and validity

This route to polydispersity is set up to give all particles the correct self-mobility according to their size, but it has some quirks when it comes to the cross-mobilities, i.e. $\langle \delta\mathbf{r}_i \otimes \delta\mathbf{r}_j \rangle$ for $i \neq j$. This is only contributed to when *both* particles are displaced so it is governed entirely by the size of the larger particle, whereas intuition says the actual tensor would be aware of the sizes of both particles. Indeed this is the case in the tensors Batchelor found for two hard spheres of differing size [121]. This limits the accuracy at short ranges, although in the absence of lubrication effects [39] or careful implementation of boundary conditions at the particle surfaces this is perhaps no greater a concern than the short-ranged inaccuracies that already exist.

A final point to be aware of is that the calculations in this section assumed $\lambda_{\min} < \lambda_{\max}$ for all particles, and therefore did not modify the Fourier moves. If this condition were not met and the Fourier moves remain unmodified then, at a hydrodynamic level, you unwittingly set $\lambda_i = \lambda_{\max}$ and the large particles may not be as large as you think. These concerns will not be important most of the time, but if you are interested in a mix of particles with very large differences in size then it is imperative the box size is set so $\lambda_{\max} \geqslant \max(\lambda_{\min})$.

Alternatively you could chose to generalise the Fourier distributions to allow for any size of $\lambda_{\min}$. This is certainly possible, though it is messy and best avoided unless increasing the box size is too costly.

4.3 Rotations in WMCD

This section sets out how to include rotations in WMCD based on the Langevin equations and mobility tensors in Section 2.2.1. The Langevin equations themselves will not feature heavily because, as with the purely translational case, the smart WMCD algorithm will enact them implicitly. Instead the attention lies on the rotational mobility tensors $\boldsymbol{\mathcal{G}}^{RT}$ and $\boldsymbol{\mathcal{G}}^{RR}$ and how to generate them in WMCD. For convenience they are repeated here alongside the rotational FDT results:

$$\boldsymbol{\mathcal{G}}^{RT}_{ij} = \delta_{ij}\,\mathbf{0} - (1-\delta_{ij})\frac{[\hat{\mathbf{r}}_{ij}]_\times}{8\pi\eta r_{ij}^2}, \tag{4.28}$$

$$\boldsymbol{\mathcal{G}}^{RR}_{ij} = \frac{\delta_{ij}}{8\pi\eta a_i^3}\mathbf{I} + \frac{1-\delta_{ij}}{16\pi\eta r_{ij}^3}(3\hat{\mathbf{r}}_{ij}\otimes\hat{\mathbf{r}}_{ij} - \mathbf{I}), \tag{4.29}$$

$$\left\langle \boldsymbol{\Xi}_i(t)\otimes\boldsymbol{\xi}_j(t')\right\rangle = 2k_BT\,\boldsymbol{\mathcal{G}}^{RT}_{ij}\,\delta(t-t'), \tag{4.30}$$

$$\left\langle \boldsymbol{\Xi}_i(t)\otimes\boldsymbol{\Xi}_j(t')\right\rangle = 2k_BT\,\boldsymbol{\mathcal{G}}^{RR}_{ij}\,\delta(t-t'). \tag{4.31}$$

A notable feature of both Eq.s (4.28) and (4.29) is that, much like the Oseen tensor, they are poorly behaved in the limit $r_{ij}\to 0$. As with $\boldsymbol{\mathcal{G}}^{TT}$, the lower wavelet cut-off in WMCD will be seen to regularise these tensors. In the meantime it is worth reiterating that these tensors are for perfect spheres and the equivalent tensors for non-spherical particles, whether by design or imperfection, can have significant differences. For example, where the self term $\boldsymbol{\mathcal{G}}^{RT}_{ii}$ is zero for a sphere, it can be non-zero for an anisotropic particle [56, 122].

Ignoring the self term in each case, it can be shown that

$$\boldsymbol{\mathcal{G}}^{RT} = \frac{1}{2}\nabla\times\boldsymbol{\mathcal{G}}^{TT}_{\text{Oseen}} \tag{4.32}$$

$$\boldsymbol{\mathcal{G}}^{RR} = \frac{1}{2}\nabla\times\boldsymbol{\mathcal{G}}^{RT} \tag{4.33}$$

which finds simple interpretation when considering the vorticity $\nabla\times\mathbf{v}$. As this is twice the local angular velocity in the fluid, the noise terms are related by $\boldsymbol{\Xi} = (1/2)\nabla\times\boldsymbol{\xi}$, and substitution into Eq.s (4.30) and (4.31) recovers these results.

In the context of WMCD, where $\boldsymbol{\mathcal{G}}^{TT}$ is also constructed from a dyadic product of vectors, the same procedure finds the rotational moves

$$\boldsymbol{\omega}_w = \frac{A_w}{2\delta t}\nabla\times\mathbf{w} = \frac{A_w}{2\delta t\lambda}\left(\hat{\mathbf{p}}\,\nabla^2_{\mathbf{x}}\phi(\mathbf{x}) - (\hat{\mathbf{p}}\cdot\nabla_{\mathbf{x}})\nabla_{\mathbf{x}}\phi(\mathbf{x})\right) \tag{4.34}$$

$$\boldsymbol{\omega}_F = \frac{A_F}{2\delta t}\nabla\times\cos(\mathbf{k}\cdot\mathbf{r}+\Phi)\hat{\mathbf{e}} = -\frac{A_F}{2\delta t}\sin(\mathbf{k}\cdot\mathbf{r}+\Phi)\,\mathbf{k}\times\hat{\mathbf{e}}. \tag{4.35}$$

While these expressions are vital for applying rotations in a WMCD simulation, in analytic calculations it is far easier to take the curl of Eq. (3.41) repeated here for

easy reference:

$$\boldsymbol{\mathcal{G}}_{ij}^{TT}(\mathbf{r}_{ij}) = \frac{1}{4\pi^2 M_3 \eta r_{ij}} \int_0^\infty dq\, q^3 \tilde{\phi}(q)^2 \Big[\mathrm{Si}\,(Q)\,(\mathbf{I} + \hat{\mathbf{r}}_{ij} \otimes \hat{\mathbf{r}}_{ij}) + \frac{\sin Q - Q\cos Q}{Q^2}\,(\mathbf{I} - 3\hat{\mathbf{r}}_{ij} \otimes \hat{\mathbf{r}}_{ij}) \Big], \quad (4.36)$$

having substituted in $A_0^2 = 3/(4\pi\eta M_3)$ from Eq. (3.43) and again with $Q = qr_{ij}/\lambda_{\min}$.

4.3.1 The rotation-translation tensor

The WMCD $\boldsymbol{\mathcal{G}}_{ij}^{TT}$ tensor has many terms with zero curl, which is easiest to see in the Fourier basis where $\mathbf{k} \times (\hat{\mathbf{k}} \otimes \hat{\mathbf{k}}) = 0$. The only surviving term is *twice* the one in $\mathrm{Si}(Q)\mathbf{I}$, leading to

$$\begin{aligned} \boldsymbol{\mathcal{G}}_{ij}^{RT}(\mathbf{r}_{ij}) &= \frac{1}{2}\nabla \times \left[\frac{2}{4\pi^2 M_3 \eta} \int_0^\infty dq q^3 \tilde{\phi}(q)^2 \frac{\mathrm{Si}(Q)}{r_{ij}} \right] \mathbf{I} \\ &= \left[\frac{1}{4\pi^2 M_3 \eta} \int_0^\infty dq q^3 \tilde{\phi}(q)^2 \left(\sin Q - \mathrm{Si}\,(Q)\right) \right] \frac{[\hat{\mathbf{r}}_{ij}]_\times}{r_{ij}^2}. \end{aligned} \quad (4.37)$$

When checking this against the tensor in Eq. (4.28) it is important to focus on $r_{ij} \geqslant 2\lambda_{\min}$ as it is only at these distances that WMCD has wavelets to supply the full dynamics. Fig. 4.1 shows that for these distances the integral in Eq. (4.37) is both negative and constant, and hence all but the numerical factor is seen to agree.

Regarding this numerical factor, the constant value of -1.02 seen in Fig. 4.1 is determined by the choice of mother wavelet. The long-range limiting value of the integral can actually be deduced analytically because as $r_{ij}/\lambda_{\min} \to \infty$ the sine term oscillates much faster than any change in $q^3\tilde{\phi}^2$, leading this term to integrate to zero. Meanwhile $\mathrm{Si}(Q) \to \pi/2$, leaving the integral to equal $\pi M_3/2$. Consequently the long-range form of the tensor is

$$\boldsymbol{\mathcal{G}}_{ij}^{RT}(r_{ij} \geqslant 2\lambda_{\min}) = -\frac{[\hat{\mathbf{r}}_{ij}]_\times}{8\pi\eta r_{ij}^2}, \quad (4.38)$$

in complete agreement with Eq. (4.28).

The other known limit to compare to is $r_{ij} \to 0$ (or more correctly $r_{ij} = 0$ since the standard tensor does not behave smoothly in this limit) where the tensor should approach 0. In this limit the Taylor series of the integrand can be taken, finding to lowest order

$$\sin Q - \mathrm{Si}(Q) \approx -\frac{Q^3}{9} \quad (4.39)$$

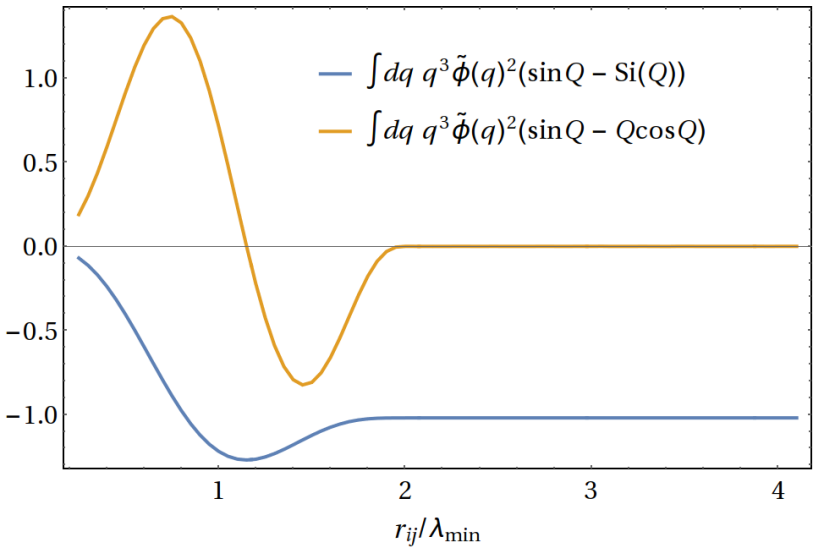

Figure 4.1: Integrals appearing in Eq.s (4.37) and (4.42) evaluated for the cubic wavelet. Their constant value for $r_{ij}/\lambda_{\min} \geqslant 2$ is generic for all choices of mother wavelet.

and in turn

$$\boldsymbol{\mathcal{G}}^{RT}_{ij}(r_{ij} \to 0) = -\frac{M_6\ r_{ij}[\hat{\mathbf{r}}_{ij}]_\times}{36\pi^2 M_3 \eta \lambda^3_{\min}}. \tag{4.40}$$

This correctly limits to 0, while the best test of the functional form is how its curl (i.e. $\boldsymbol{\mathcal{G}}^{RR}$) behaves.

4.3.2 The rotation-rotation tensor

Now the process is repeated to obtain

$$\begin{aligned} \boldsymbol{\mathcal{G}}^{RR}_{ij} &= \frac{1}{2}\nabla \times \boldsymbol{\mathcal{G}}^{RT}_{ij} && (4.41)\\ &= \frac{1}{8\pi^2 M_3 \eta r^3_{ij}} \int_0^\infty dq q^3 \tilde{\phi}(q)^2 \left[(\sin Q - \mathrm{Si}(Q))\,(\mathbf{I} - 3\hat{\mathbf{r}}_{ij}\otimes\hat{\mathbf{r}}_{ij})\right. \\ &\qquad \left. + (\sin Q - Q\cos Q)(\mathbf{I} - \hat{\mathbf{r}}_{ij}\otimes\hat{\mathbf{r}}_{ij})\right]. && (4.42) \end{aligned}$$

The first task is again to verify the long ranged behaviour, which reuses the above analysis for $\sin Q - \mathrm{Si}(Q)$ and uses Fig. 4.1 to disregard the remaining terms that integrate to zero. Hence the long range form is seen to be

$$\boldsymbol{\mathcal{G}}^{RR}_{ij}(r_{ij} \geqslant 2\lambda_{\min}) = \frac{3\hat{\mathbf{r}}_{ij}\otimes\hat{\mathbf{r}}_{ij} - \mathbf{I}}{16\pi\eta r^3_{ij}} \tag{4.43}$$

matching the long range part of Eq. (4.29) as expected.

What will prove more onerous is the $r_{ij} \to 0$ limit, or equivalently $i = j$.

Here the $\hat{\mathbf{r}} \otimes \hat{\mathbf{r}}$ terms in Eq. (4.42) cancel exactly leading to the isotropic tensor

$$\boldsymbol{\mathcal{G}}_{ii}^{RR} = \frac{M_6}{36\pi^2 M_3 \eta \lambda_{\min}^3} \mathbf{I}, \tag{4.44}$$

in which the moments of $\tilde{\phi}^2$ have not cancelled. Matching this to the self term in Eq. (4.29) therefore requires

$$\left(\frac{\lambda_{\min}}{a}\right)^3 = \frac{2M_6}{9\pi M_3}. \tag{4.45}$$

This result requires great care as is seen when using the usual $\lambda_a = 2M_4/(\pi M_3)$, leading to a condition on the moments:

$$\Upsilon \equiv \frac{\pi^2}{36} \frac{M_6}{M_3} \left(\frac{M_3}{M_4}\right)^3 = 1. \tag{4.46}$$

This condition is *not* met for the cubic wavelets, but other wavelets can be found to meet the condition. A detailed discussion on the choice of mother wavelet is left for Section 4.4 so as to not sidestep too far from the present discussion. For now it is assumed $\Upsilon \neq 1$ and work proceeds by aligning rotational and translational diffusion in a way that does not require Υ to be unity[1].

4.3.3 Rotation-only moves

Although correlating translational and rotational motion with Eq. (4.30) requires both types of moves to be performed inside the same moves, this does not preclude the use of moves of pure rotation or translation alongside these mixed moves. The approach taken parallels the use of Fourier moves alongside wavelets, this time adding wavelets that are identical to the wavelets used thus far but with zero translational displacement.

Without modifying the PDFs the only parameter with the freedom to allow this is $\lambda_{\min}$, or λ_a since this ought to be developed with polydispersity in mind. Thus λ_a is split into translational and rotational values

$$\lambda_{a,T} = \frac{2}{\pi} \frac{M_4}{M_3}, \tag{4.47}$$

$$\lambda_{a,R} = \sqrt[3]{\frac{2}{9\pi} \frac{M_6}{M_3}}, \tag{4.48}$$

from Eq.s (3.45) and (4.45) respectively.

[1] As real particles are not perfect spheres you may not be too worried about satisfying the Υ condition exactly. Nevertheless, it is desirable to be able to tune the relative rotational and translational motion and the upcoming methods motivated by this condition can be used more generally for this purpose.

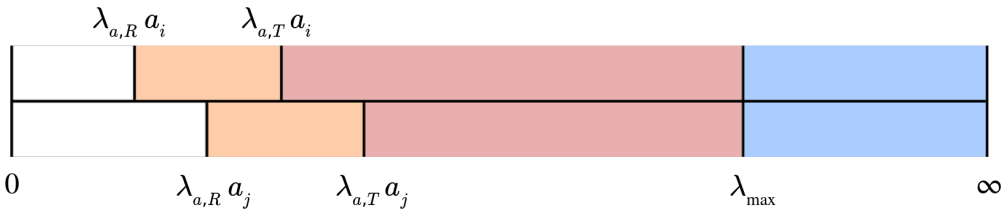

Figure 4.2: Schematic of ranges of distinct types of moves for two particles i and j of different hydrodynamic radii, as viewed on the wavelet radius axis. From left to right, the move types are: no move (white); rotation-only wavelet (orange); rotation and translation wavelet (red); and Fourier (blue).

This work assumes $\lambda_{a,R} < \lambda_{a,T}$, and hence that there are wavelets with $\lambda_{a,R}a \leqslant \lambda < \lambda_{a,T}a$ that only serve to rotate particles[2]. Algorithmically this is not a major change from the procedure established in previous sections: one simply needs to choose rotation-only moves with the appropriate probability then follow the usual wavelet generation procedure without performing translational displacements in the update step. The three key details that need addressing are the probability of making a rotation-only move, the change to the time increment this entails and the use of translational move biasing in rotation-only moves.

In monodisperse systems the probability is easy to calculate because the rotation-only moves are nothing but an extension to the domain of λ. The distributions of all other parameters are identical, so the probability of a rotation-only move is simply the integral of the normalised PDF:

$$\begin{aligned} P_r &= \frac{\int_{\lambda_{a,R}a}^{\lambda_{a,T}a} d\lambda \lambda^{-4}}{\int_{\lambda_{a,R}a}^{\infty} d\lambda \lambda^{-4}} \\ &= 1 - \left(\frac{\lambda_{a,R}}{\lambda_{a,T}}\right)^3 = 1 - \Upsilon. \end{aligned} \tag{4.49}$$

Writing the version in a polydisperse system is a more delicate task because of the overlap of rotation-only and mixed moves as shown in Fig. 4.2. The Heaviside functions in the translational wavelet and its rotational counterpart

$$\boldsymbol{\omega}_i = \frac{A_w}{2\delta t}(\nabla \times \mathbf{w}_i)\, \Theta(\lambda - \lambda_{a,R}a_i) \tag{4.50}$$

can handle many of the complications in the code but they still need feeding into the analytics too.

Allowing any $\lambda_{\min,T} \leqslant \min\{\lambda_{a,T}a_i\}$ and setting $\lambda_i = \lambda_{a,R}a_i$, P_r is found to

[2]The reverse could be true, at which point obvious changes to upcoming results need to be made, but Section 4.4 will show that the chosen way around is the more fruitful.

be

$$\begin{aligned} P_r &= \frac{\int_0^{\lambda_{\min,T}} d\lambda \lambda^{-4} K(\lambda)}{\int_0^{\infty} d\lambda \lambda^{-4} K(\lambda)} \\ &= \left(\sum_i^N \lambda_{a,R}^{-3} a_i^{-3} \right)^{-1} \sum_j^N \Theta(\lambda_{\min,T} - \lambda_{a,R} a_j) \left(\lambda_{a,R}^{-3} a_j^{-3} - \lambda_{\min,T}^{-3} \right). \end{aligned} \quad (4.51)$$

This form is complicated but only needs calculating once each simulation and is sufficiently general to cover all valid choices of $\lambda_{\min,T}$.

There are similar intricacies for the probability of making a Fourier move, that depend on exactly when you choose between move types. If the procedure is

```
If(rand1 < P_r) { Rotation-only wavelet move }
Else
{
        If(rand2 < P_F) { Mixed Fourier move }
        Else { Mixed wavelet move }
}
```

with $\texttt{rand1}, \texttt{rand2} \in [0, 1]$, then the correct P_F is

$$P_F^{\infty} = N \lambda_{\max}^{-3} \left(\sum_{i=1}^N \max(\lambda_{\min,T}, \lambda_{a,R} a_i)^{-3} \right)^{-1} \quad (4.52)$$

in unbounded systems, while the periodic version simply feeds this into Eq. (4.26). If the procedure is instead

```
If(rand1 < P_r) { Rotation-only wavelet move }
Else if(rand1 < P_r + P_F) { Mixed Fourier move }
Else { Mixed wavelet move }
```

then Eq. (4.52) picks up a factor of $(1 - P_r)$.

Fortunately, taking care with these probabilities ensures the actual distribution of moves made is identical for all choices of $\lambda_{\min,T} \leqslant \min\{\lambda_{a,T} a_i\}$ and the time increment does not need to worry about the details. For the time increment the argument is that the (average) time update across all rotation moves (from $\lambda_{a,R} a$ to ∞) must be the same as across translation moves (from $\lambda_{a,T} a$ to ∞) in a simulation where rotations are not considered, because the translational motion is identical in the two cases. This leads to comparing the number of moves required to sample

each of these ranges, which amounts to

$$\begin{aligned}\frac{\delta t_R}{\delta t_T} &= \frac{\int_{\lambda_{a,R}a}^{\infty} d\lambda \lambda^{-4}}{\int_{\lambda_{a,T}a}^{\infty} d\lambda \lambda^{-4}} \\ &= \left(\frac{\lambda_{a,R}}{\lambda_{a,T}}\right)^3 = \Upsilon. \end{aligned} \tag{4.53}$$

This result applies to both mono- and polydisperse systems, since the latter simply has sums of integrals that do not change the final ratio. Eq. (4.24) then gives

$$\Rightarrow \delta t = \frac{6\pi M_4}{(2\pi)^3} \frac{\eta A_0^2}{\lambda_{a,T} k_B T} \left(\sum_{i=1}^{N} \lambda_{a,R}^{-3} a_i^{-3} \right)^{-1} \tag{4.54}$$

The final note on rotation-only moves follows from the discussion of move biasing in polydisperse systems. For consistency, if each force is treated as a perfect point force with an associated Stokeslet, its full rotation field should be generated too. Therefore, forces are included in the move bias for rotation-only moves, even though they do not confer any translational motion.

4.3.4 Biasing by torques in smart WMCD

Now that the mobility tensors are known attention turns to biasing smart WMCD moves with torques as well as linear forces.

To begin, the bias on $\mathbf{Q}$ and its Fourier analogue are split into translational and rotational terms,

$$\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{F}}_t + \boldsymbol{\mathcal{F}}_r, \tag{4.55}$$

$$\boldsymbol{\mathcal{F}}_F = \boldsymbol{\mathcal{F}}_{F,t} + \boldsymbol{\mathcal{F}}_{F,r}, \tag{4.56}$$

with $\boldsymbol{\mathcal{F}}_t$ and $\boldsymbol{\mathcal{F}}_{F,t}$ found as per Section 4.1. The task then is to find $\boldsymbol{\mathcal{F}}_r$ and $\boldsymbol{\mathcal{F}}_{F,r}$ for a given set of torques $\boldsymbol{\Gamma}_i$, which are really point rotations at their respective positions $\mathbf{r}_i$. In each case

$$\boldsymbol{\mathcal{F}}_r = -\nabla_{\mathbf{Q}}(\Delta U) = -\nabla_{\mathbf{Q}} \left(-\sum_i \boldsymbol{\Gamma}_i \cdot \boldsymbol{\omega}_i \delta t \right) \tag{4.57}$$

is sought using $\boldsymbol{\omega}$ in Eq.s (4.34) and (4.35). This finds

$$\boldsymbol{\mathcal{F}}_{w,r} = \frac{1}{2\lambda} \sum_i \left(\boldsymbol{\Gamma}_i \nabla_{\mathbf{x}}^2 \phi(\mathbf{x}_i) - (\boldsymbol{\Gamma}_i \cdot \nabla_{\mathbf{x}}) \nabla_{\mathbf{x}} \phi(\mathbf{x}_i) \right) \tag{4.58}$$

$$\boldsymbol{\mathcal{F}}_{F,r} = \frac{1}{2} \sum_i \sin(\mathbf{k} \cdot \mathbf{r}_i + \Phi)\, \mathbf{k} \times \boldsymbol{\Gamma}_i. \tag{4.59}$$

Both be checked with similar calculations to Section 4.1.3 to give $\langle \boldsymbol{\omega} \rangle \propto \boldsymbol{\Gamma}$.

4.4 Comparing choices of mother wavelet

This section investigates properties of different mother wavelets. Apart from a brief mention of the cubic wavelet back in Section 3.2 the topic of choosing a wavelet has been left untouched, and even then it was presented mainly to shore up understanding of otherwise fairly abstract calculations, rather than towards a serious discussion.

So far then, this thesis has overlooked one of the advantages of wavelet analysis: that there is great freedom to tailor wavelets to a given task. This matter has been left until now because it is only after presenting the full scope of WMCD that one knows what is desired of this choice. In particular, every single result found thus far that contains any moment M_m, $\lambda_{a,T}, \lambda_{a,R}$, or direct mention of the wavelet or its stream function ϕ is dependent on this choice and helps determine which choice of mother wavelet is optimal.

Needless to say there have been too many occurrences of these quantities over the last two chapters to realistically take all of them into account. Likewise, with an infinite set of mother wavelets to choose from it is not possible to truly optimise the choice, even when constrained to the form in Eq. (3.14). What is presented in this section is therefore not the full story of how to optimise the mother wavelet in WMCD, but an identification of many of the key properties to consider when making this choice.

4.4.1 Definitions of wavelets considered

The properties of three mother wavelets will be explored. The first is the cubic wavelet introduced previously, which will now be given the subscript 3 to distinguish it from the alternatives:

$$\phi_3(r) = \frac{1}{2}r^2 - \frac{1}{3}r^3 - \frac{1}{6}, \tag{4.60}$$

$$\tilde{\phi}_3(k) = 4\pi\, k^{-6} \left(5k \sin k - (k^2 - 8)\cos k - 8\right), \tag{4.61}$$

$$\mathbf{w}_3(\mathbf{r}, \hat{\mathbf{p}}) = r(1-r)\hat{\mathbf{p}} \times \hat{\mathbf{r}}, \tag{4.62}$$

$$\mathbf{w}_3^R(\mathbf{r}, \lambda, \hat{\mathbf{p}}) = \frac{1}{2\lambda}\left[(2-3r)\,\hat{\mathbf{p}} + r(\hat{\mathbf{p}} \cdot \hat{\mathbf{r}})\hat{\mathbf{r}}\right] \tag{4.63}$$

with the rotational mother wavelet $\mathbf{w}^R \equiv (1/2)\nabla \times \mathbf{w}$ and the understanding that all position-space equations have $\mathbf{r} = (\mathbf{r}_i - \mathbf{b})/\lambda$ and $0 \leqslant r \leqslant 1$.

ϕ_3 and its derivatives are plotted in Fig. 4.3(a), with the first and second derivatives roughly corresponding to translational and rotational displacements respectively. A notable feature is that $\partial_r^2 \phi_3$ is non-zero at $r = 1$, leading to a finite

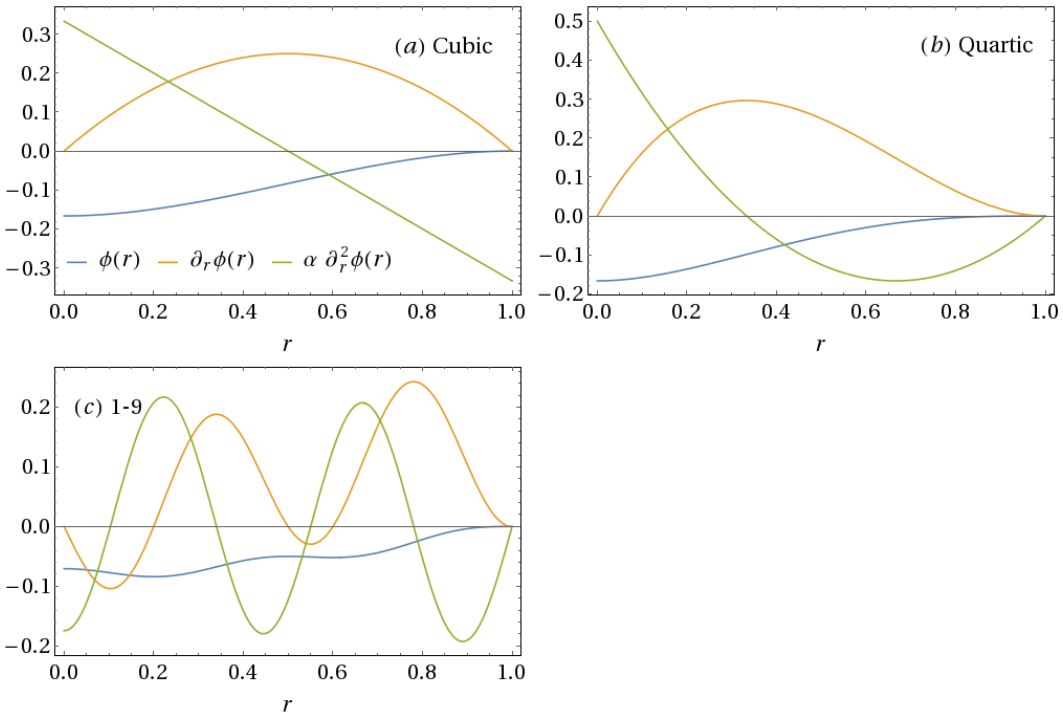

Figure 4.3: Plots of the zeroth, first and second derivatives of $\phi(r)$ for (a): cubic, (b): quartic and (c): 1-9 wavelets. The value of α is 1/3, 1/4 and 1/9 respectively.

jump in rotations across the wavelet's boundary. In simulations that do not account for rotational motion this is of little concern, but since the cubic wavelet was constructed as a simple function that provides zero shear across the boundary, it makes sense to extend this to rotations too.

The simplest approach to achieve this is to add another term into ϕ whose coefficient can be tailored to give a vanishing second derivative at $r = 1$. This leads to the quartic wavelet:

$$\phi_4(r) = r^2 - \frac{4}{3}r^3 + \frac{1}{2}r^4 - \frac{1}{6}, \tag{4.64}$$

$$\tilde{\phi}_4(k) = 16\pi\, k^{-7}\left((15 - k^2)\sin k - 7k\cos k - 8k\right), \tag{4.65}$$

$$\mathbf{w}_4(\mathbf{r}, \hat{\mathbf{p}}) = 2r(1-r)^2\hat{\mathbf{p}} \times \hat{\mathbf{r}}, \tag{4.66}$$

$$\mathbf{w}_4^R(\mathbf{r}, \lambda, \hat{\mathbf{p}}) = \frac{2}{\lambda}\left[(1-2r)\,\hat{\mathbf{p}} + r(\hat{\mathbf{p}}\cdot\hat{\mathbf{r}})\hat{\mathbf{r}}\right]. \tag{4.67}$$

Common features of the cubic and quartic wavelets include vanishing first derivatives at both $r = 0$ and $r = 1$. As has already been stated, the latter leads to finite shear at the boundary, which is useful for the strain energy calculations used for setting A_w. Meanwhile, the zero gradient at $r = 0$ ensures a finite strain around the wavelet's axis (a finite gradient would imply rotation angles $\sim (r\sin\theta)^{-1}$ diverge).

They also share the un-ideal feature that $\Upsilon \neq 1$, requiring some separation of translational and rotational moves or an acceptance that the particle represented will not behave as an ideal solid sphere. A first guess at a wavelet satisfying $\Upsilon = 1$ might be a quintic wavelet which would add an extra coefficient to tune Υ with. However, while coefficients can be chosen to maximise Υ, it will always be less than unity in this case.

Rather than go through the arduous task of tuning multiple coefficients in yet higher order polynomial wavelets, it is illustrative to introduce a class of trigonometric wavelets:

$$\phi_{\ell,m}(r) = \frac{1}{4\pi}\left(\frac{s_m}{m} - \frac{s_\ell}{\ell}\right), \tag{4.68}$$

$$\tilde{\phi}_{\ell,m}(k) = \frac{\pi}{k}\left(\frac{k\cos(k) + \frac{1}{2}\left(k^2 - \frac{\pi^2\ell^2}{4}\right)\sin(k) - k\cos\left(\frac{\pi\ell}{2}\right)}{\left(k^2 - \frac{\pi^2\ell^2}{4}\right)^2} - \text{ same with } \ell \leftrightarrow m\right), \tag{4.69}$$

$$\mathbf{w}_{\ell,m}(\mathbf{r}, \hat{\mathbf{p}}) = \frac{1}{8}(c_\ell - c_m)\hat{\mathbf{p}} \times \hat{\mathbf{r}}, \tag{4.70}$$

$$\mathbf{w}^R_{\ell,m}(\mathbf{r}, \lambda, \hat{\mathbf{p}}) = \frac{\pi}{32\lambda}\Bigg[\left(\ell s_\ell - m s_m\right)\hat{\mathbf{p}} + \left(\frac{2}{\pi r}(c_\ell - c_m) - \left(\ell s_\ell - m s_m\right)\right)(\hat{\mathbf{p}} \cdot \hat{\mathbf{r}})\hat{\mathbf{r}}\Bigg], \tag{4.71}$$

where $\ell, m \in \mathbb{Z}$, $s_m = \sin(\pi m(1-r)/2)$ and $c_m = \cos(\pi m(1-r)/2)$.

Wavelets of this kind will satisfy the same conditions as the cubic and quartic wavelets when both ℓ and m are odd, or when they are even with $(\ell - m)\text{mod}4 = 0$. Analytic expressions of the moments M_m are not always forthcoming, but $\tilde{\phi}_{\ell,m}$ is well behaved and decays fast enough for each moment to be easy to calculate numerically. Thus the value of Υ can be obtained for allowed values of ℓ and m, and these have been plotted in Fig. 4.4.

The values of Υ seen in this figure increase with m (or more generally whichever is largest of $|\ell|$ and $|m|$). This makes sense as increasing m increases the second derivative of $\phi_{\ell,m}$ more than the first, so yields a relative increase in rotational motion per wavelet[3]. Thus the tuning of Υ can be understood as a task in finding wavelets with the right balance of first and second derivatives.

Fortunately, small values of ℓ and m are sufficient to find a case where $\Upsilon \approx 1$, with $\ell = 1$ and $m = 9$ having $\Upsilon = 1.02$. Υ being slightly greater than unity implies motion for perfect spheres would want a small number of pure translational moves,

[3]Looking at fixed $m > \ell$ reveals Υ is actually minimal when $\ell \approx m$ and increases as ℓ is reduced to 0. The reasons for this are unclear, but also unimportant and do not warrant a full discussion.

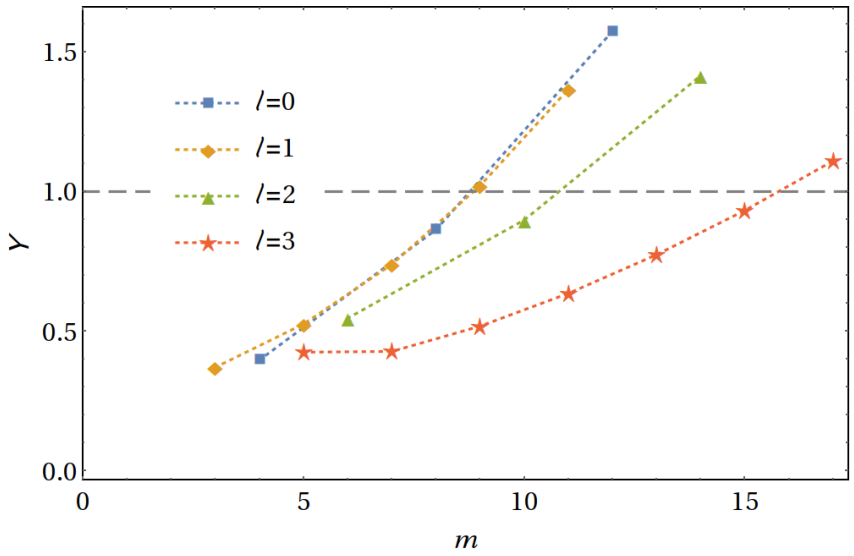

Figure 4.4: Value of Υ for various choices of ℓ and m in the wavelet defined in Eq.s (4.68)-(4.71).

Table 4.1: Important numerical quantities associated with mother wavelets.

	Cubic	Quartic	1-9
M_3	$\frac{2\pi^2}{105}(9-8\ln 2) \approx 0.6945$	$\frac{128\pi^2}{2835}(5-6\ln 2) \approx 0.3748$	0.3377
M_4	$\frac{(2\pi)^3}{105} \approx 2.362$	$\frac{2(2\pi)^3}{315} \approx 1.575$	$\frac{13079\pi}{129600} + \frac{\pi^3}{24} \approx 1.609$
M_6	$\frac{(2\pi)^3}{5} \approx 49.61$	$\frac{16(2\pi)^3}{105} \approx 37.80$	$\frac{1119\pi^3}{6400} + \frac{41\pi^5}{96} \approx 136.1$
$\lambda_{a,T}$	2.316	2.675	3.033
$\lambda_{a,R}$	1.755	1.925	3.055
Υ	0.4352	0.3727	1.022

reversing the scheme discussed in Section 4.3.3, but in practice it is close enough to ignore the difference and set all moves to supply both translational and rotational motion. Henceforth all analysis will focus on these values of ℓ and m, and accordingly the wavelet will be referred to as the '1-9' wavelet. The derivatives of $\phi_{1,9}$ are shown in Fig. 4.3(c).

With all three wavelets now defined, Table 4.1 collates the numerical quantities pertinent to their comparison. Be aware that the moments themselves can be scaled with global numerical factors in the definitions of the stream functions. That said, Eq. (3.43) tells us that for moves with different mother wavelets to have a roughly equal contribution to mobility, the amplitude should follow

$$A_0^2 \propto M_3^{-1}. \tag{4.72}$$

For a fair comparison this is assumed below.

4.4.2 Time and computational cost

One inevitable question surrounding which wavelet is optimal is how the computational cost depends on the choice. The answer would lie in any wavelet dependence in the results in Section 3.7, but since these results were obtained prior to the many changes of this chapter it is first helpful to focus on the time increment δt.

By now this has been presented in several forms differing by mono- vs polydispersity as well as whether purely rotational moves are included. To begin with, the focus is on the polydisperse translational δt in Eq. (4.24), in which $\lambda_i \propto \lambda_{a,T}$ leads to

$$\delta t_T \propto \frac{A_0^2 M_4}{\lambda_{a,T}} \lambda_{a,T}^3 \propto \lambda_{a,T}^3 \tag{4.73}$$

when ignoring all factors independent of the mother wavelet. The persisting mother wavelet dependence stems entirely from $\mathcal{N}_\lambda$, and hence is a measure of the number of wavelets required to supply δt. With rotations included Eq. (4.53) gives

$$\delta t_R = \Upsilon \delta t_T \propto \lambda_{a,R}^3. \tag{4.74}$$

Passing this into the computational cost results in Section 3.7 gives the relevant factors

$$C \propto \frac{\lambda_a^{d_f}}{\delta t} \propto \lambda_a^{d_f - 3} \tag{4.75}$$

for either translations or rotations as appropriate. Note that while $\langle P_{\text{acc}} \rangle$ (and ΔU contained therein) most definitely does carry some dependence on the mother wavelet, in the context of smart WMCD this is a very minor consideration owing to the very small rejection probability.

In fractal systems ($d_f < 3$) the factors of λ_a do not cancel and the cost favours a larger λ_a, while all dependence on the mother wavelet cancels out in homogeneous systems. That is not to say minor differences do not exist, especially since this cancellation only truly applies to moves large enough to 'see' the fractal dimension, but in comparisons between cubic and quartic wavelets the cost only had a slight preference for the latter.

4.4.3 Details of mobility tensors

At first sight the question of how the choice of mother wavelet affects the mobility tensor seems an odd one. After all, right from the start WMCD was set up to reconstruct the Oseen tensor for *any* mother wavelet, and the same was true of the rotation tensors earlier in this chapter. Are the mobility tensors not going to be the same regardless of this choice?

The answer is both yes and no, and hinges in the existence of $\lambda_{\min}$ and the

locality of wavelet moves. Certainly any particles separated by a distance greater than $2\lambda_{\min}$ would be unable to tell the difference (at least when averaged over a sufficient number of wavelets) because of Theorem 1 at the end of the previous chapter. However, the story is very different for pairs of particles with separations smaller than this. For these, many of the moves involved in generating correlated motion are missing, and hence they no longer simply follow Oseen or the rotational equivalents. Furthermore, exactly how it changes is going to depend on the details of the mother wavelet, as is most clear by considering two extreme wavelets: one with all its motion near its centre and another with all its motion near its edge.

In the first example moves with $\lambda \approx r_{ij}/2$ do not contribute much because the only such moves that contain both particles would have them towards the outside where there is little motion. Cutting out these moves would have little effect and the correlations would still resemble Oseen.

In the example with all motion near the edge, the correlated motion of a pair has a major contribution from moves with $\lambda \approx r_{ij}/2$, so cutting these out would utterly destroy the Oseen power law.

To see the effect of the more moderate cubic, quartic and 1-9 wavelets, Fig. 4.5 plots the non-zero components of all 3 mobility tensors, while Fig. 4.6 plots the same curves divided by their long-ranged forms (note the different scaling on the horizontal axes, aimed at revealing the physical tensor and its agreement beyond $2\lambda_{\min}$ respectively). Parts (a) and (b) in both figures show that near-field translational mobilities are qualitatively similar for all three wavelets, with the main difference being how steep the rise to the self-mobility is.

It is only when looking at rotations that major differences are observed, especially with the 1-9 wavelet. In Fig. 4.5(c) it is seen to decay much faster than the other two wavelets when $r/\lambda_{\min} < 0.5$, while in part (d) this is such that the tensor is practically zero for all but the closest (overlapping) pairs of particles. This is in contrast to the cubic and quartic wavelets which have a much smoother and more realistic bridge between the positive self- and negative far-field correlations. Indeed Fig. 4.6(d) shows most clearly how $\boldsymbol{\mathcal{G}}^{RR}_{\theta\theta}$ changes sign 3 times, and has significant undulations also seen in $\boldsymbol{\mathcal{G}}^{RR}_{rr}$ and $\boldsymbol{\mathcal{G}}^{RT}_{\theta\varphi}$ even when it doesn't change sign. Evidently there is a price to pay in the near-field for using such an oscillatory wavelet to match $\lambda_{a,T}$ and $\lambda_{a,R}$.

Differences between the cubic and quartic wavelets are relatively minor, even to the point of having near-identical $\boldsymbol{\mathcal{G}}^{TT}$ tensors. The most noticeable difference is in the cubic wavelet's linear approach to $r \to 0$ in $\boldsymbol{\mathcal{G}}^{RR}$, which might be something one wants to avoid if a short-ranged rigid rotation is sought. In fact the RPY tensor shares a similar feature in $\boldsymbol{\mathcal{G}}^{TT}$, which would affect the rigidity of the translational flow. These sorts of concerns will resurface in Chapter 6, and present a physical

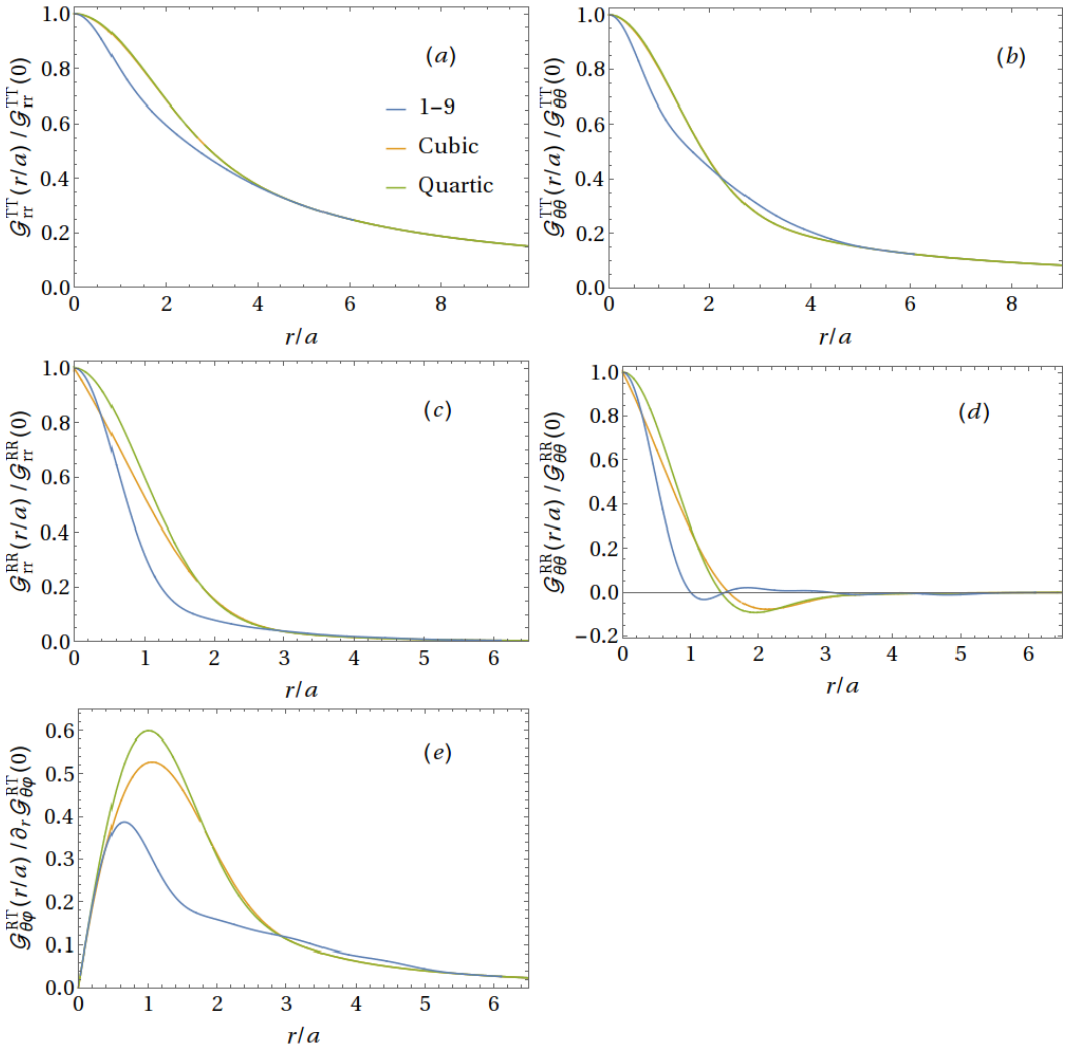

Figure 4.5: Plots of elements of mobility tensors for each of the cubic, quartic and 1-9 wavelets. Parts (a)-(d) are normalised to 1 at $r = 0$ while part (e) is normalised to have gradient 1 at $r = 0$.

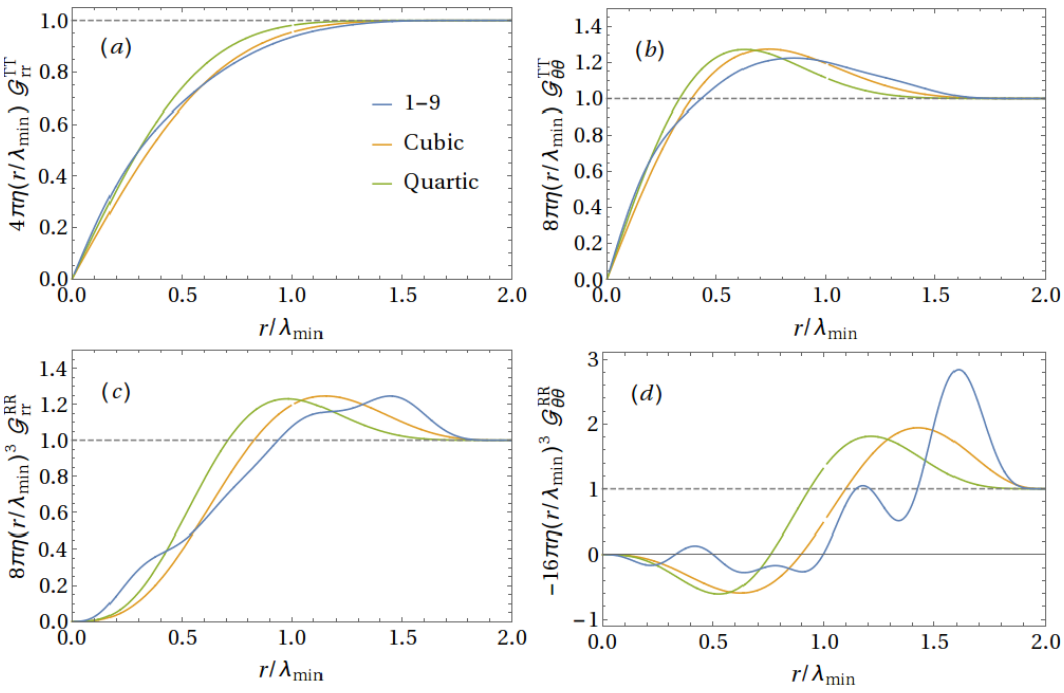

Figure 4.6: Plots of elements of mobility tensors normalised to 1 in the far-field. The plot for $\boldsymbol{\mathcal{G}}^{RT}$ is identical to (c).

motivation for choosing one wavelet over another.

4.4.4 Distribution of Fourier moves

The last and most subtle point of comparison is not in the wavelets themselves but in the distribution of Fourier moves. The focus here is on the distributions for unbounded systems because they are much easier to visualise than their discrete counterparts, while the conclusions are similar for both cases.

Fig. 4.7 shows both the PDFs and their associated CDFs

$$\mathcal{C}^{\infty}_{k}(k) = \int_0^k dk' \oint d^2\hat{\mathbf{k}}'\, \mathcal{P}^{\infty}_{\mathbf{k}}(\mathbf{k}') = 1 - \frac{1}{M_6}\int_{k\lambda_{\max}}^{\infty} dq\, q^3\tilde{\phi}(q)^2\, \left(q^3 - (k\lambda_{\max})^3\right) \tag{4.76}$$

for wavenumbers in Fourier moves. For all choices of mother wavelet $\mathcal{P}^{\infty}_{k}$ follows the generic form of initially increasing as k^2 then decaying as some power law with small oscillations or 'steps'. The steps are neither surprising nor particularly significant: they are indicative of a functional form akin to $k^{-\alpha} \times$ (sum of trig. terms), which given all $\tilde{\phi}$ have such a form it would be expected that their integrals would too.

The first important detail is the value of $k\lambda_{\max}$ at which the regimes in Fig. 4.7(a) switch. Because the PDF is normalised, a large value means the normalisation factor must be small, which is seen by the $\sim k^2$ part of the 1-9 curve in

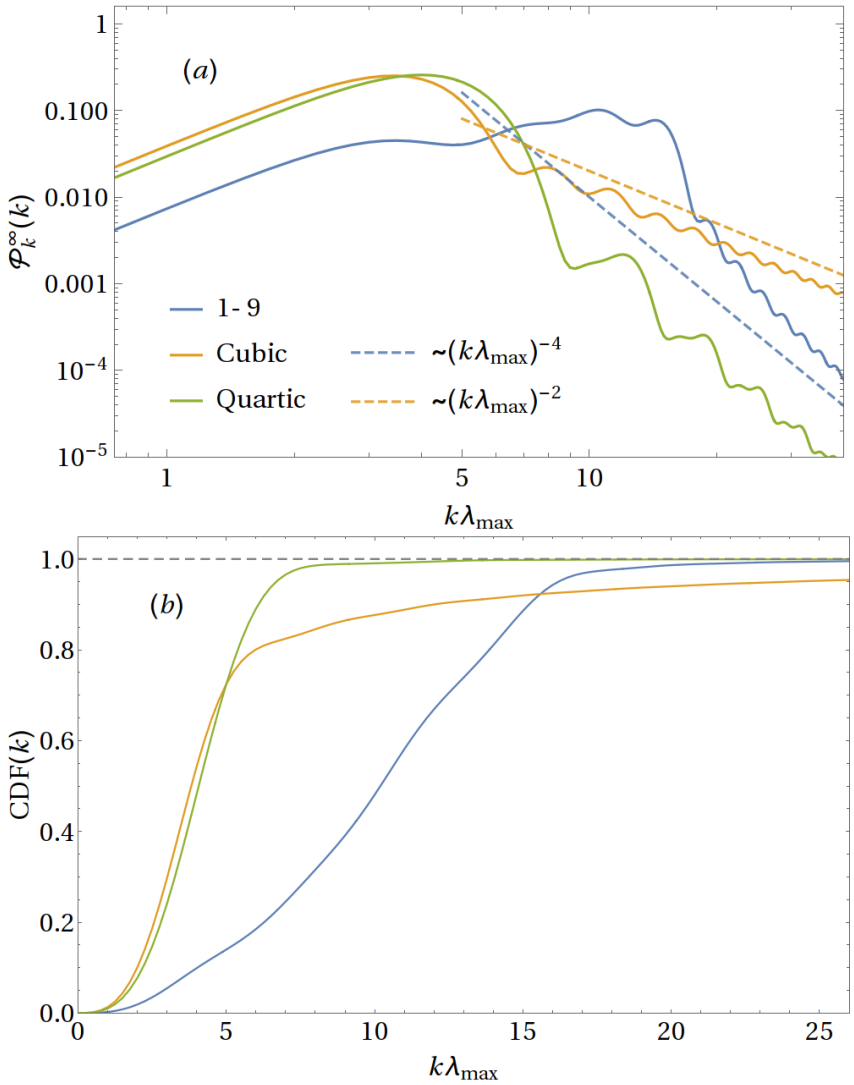

Figure 4.7: (a): PDFs of wavenumbers in unbounded systems for the three choices of wavelet. Dotted lines indicate the power law decays at high frequencies. (b): Plots of the corresponding cumulative distribution functions.

Fig. 4.7(a) being over a factor of 5 times smaller than for the other wavelets. This in turn means the CDF in Fig. 4.7(b) rises over 5 times slower, and the 1-9 wavelet leads to a relatively large number of mid-frequency moves.

Given the actual probability of making a Fourier move is no different this might not seem very important, but the implication of needing many mid-to-high frequency moves is that wavelets alone do not fully supply (or perhaps oversupply) correlations at short-to-mid ranges. Expressed more carefully, a slowly rising Fourier CDF means moves with $\lambda \gg r_{ij}$ (be them supplied as wavelets or Fourier moves) are still important for interactions at a range r_{ij}. Because of the rarity of large moves, this in turn means it requires more moves to faithfully supply the correct hydrodynamic interactions.

The power of the decay is equally important for this, and again the quartic wavelet performs well on this front with a k^{-4} decay and consequently a rapid k^{-3} convergence in the CDF, leading to a negligible number of high frequency moves. This time it is the cubic wavelet that performs poorly, with $\mathcal{P}_k^\infty \sim k^{-2}$ and a slow k^{-1} approach to unity in the CDF.

The asymptotic decay is something one has control over when choosing a mother wavelet. Due to Paley-Wiener theorems this is connected to the analyticity of ϕ at the wavelet boundary [123]. Of the wavelets being discussed, the quartic and 1-9 wavelets have a continuous second derivative at their boundary while only the first derivative of the cubic wavelet is continuous. So the rule of thumb is to use a mother wavelet with a continuous second derivative to ensure the Fourier distribution can be truncated at moderate frequencies.

4.4.5 Final remarks on wavelet choice

Having gone through the various properties of mother wavelets it is worth summarising them with a view to their intended use. On the one hand, if the aim is to just have a working WMCD code with a sensible bridge between near- and far-field mobility tensors it is really not worth worrying too much about which wavelet is used. Some choices are better than others, but as the computational cost is more or less independent of the choice and simple polynomial wavelets can already construct reasonable tensors you can go a long way without trying. The only point that is really worth considering is the continuity at the wavelet boundary, though even then it is only to make life easier by truncating Fourier moves off at lower frequencies and you can get by just fine with an imperfect choice[4]. All in all the quartic wavelet is a good option.

If, on the other hand, the fine details are important then a determined hunt

[4]The data in the next chapter was gathered before any of this analysis and used the cubic wavelet.

for the perfect wavelet may be worthwhile. Hopefully the 1-9 wavelet serves as a good example of both what to do and what to avoid. Highly oscillatory wavelets will generally give wobbly tensors and are probably best avoided as a result. However, they have illustrated that finding wavelets with $\Upsilon \approx 1$ is a task suited to a family of wavelets connected by a small number of parameters which can be varied to obtain just the right amount of second derivative to satisfy the condition.

4.5 Summary of the current state of WMCD and future paths

This chapter has presented several advances on the core WMCD algorithm. Chief among them from an algorithmic perspective is the use of smart Monte Carlo. Smart WMCD is optional and if writing software from scratch is a complication that can be avoided until the basics have been tested, but in the end it is almost strictly superior to the non-smart version, simultaneously providing improved computational cost, accuracy and move acceptance rates. The price of calculating forces is even beneficial in many circumstances and will be exploited in the next chapter, while the direct control over the relative size of causal and random motion will be useful in setting the Péclet number in Chapter 6.

The other changes expanded the scope of WMCD, which can now simulate systems with a mix of particle sizes and rotate them with the vorticity of the local flow field. The details of these advances are not perfect - the near-field cross-mobility between differently sized particles is only sensitive to the size of the larger particle for instance - but they hold their own against existing methods, especially in simulations of many particles where balancing accuracy and computational efficiency is always a concern. In fact the guarantee of divergence-free, positive definite mobility tensors that match both the required long-range structure and the self-mobility is a success in itself, even before the efficient $N \ln N$ complexity of the algorithm enters the picture.

The primary agent in these extensions has been $\lambda_{\min}$, which admittedly is now not a single system-wide parameter but one that each particle carries two distinct copies of. Nevertheless, the idea of $\lambda_{\min}$ as a one-parameter manipulator of the mobility tensors has been behind it all. Certainly this has gone a surprisingly long way and it has not yet run its course either - it can be used to manipulate self-mobilities to differ from that of a hard sphere if desired - but it is not the only trick WMCD has at its disposal.

The discussion of mother wavelets in this section also barely scratches the surface. While it has highlighted key features to look out for in computational considerations, including continuity and the associated distribution of Fourier moves,

the most interesting avenue for physical purposes would be in tuning the near-field structure of the various mobility tensors. Chapter 6 will present a system where such freedom could be extremely valuable, and with an infinite set of wavelets to choose from a determined effort must surely be fruitful there.

Two other likely routes for future developments would be parallelisation and the inclusion of different boundary conditions. Throughout the PhD many ideas have been thrown around for both, although none have been taken far enough for inclusion in this thesis. Nevertheless, the potential value of these advances is so great that the leading thoughts on how to approach them are worth a quick summary.

Parallelisation: The locality of wavelet moves allows multiple non-overlapping moves to be performed simultaneously without needing to know about each other. Variation in wavelet size will make this a non-trivial scheme to work out, especially for wavelets too large for the simulation box to fit multiples of. Ideas to avoid it becoming temporarily serial include using each processor to handle a subset of movers or even to continue to place smaller wavelets while the large one runs. There is lots to refine here, but WMCD is quite able to be parallelised with a bit of effort.

Walls and interfaces: Many real systems include liquid-liquid (e.g. oil in water) or solid-liquid interfaces (i.e. hard walls) which impose boundary conditions. These conditions affect fluid flows near the interface such that the free-space Oseen tensor and regularisations of it no longer apply, thereby leaving WMCD as presented unable to simulate these systems accurately. The effect on the mobility tensor is dependent on the geometry of the interface and is not generally known beforehand. In the instances where it is, such as the Blake tensor with an infinte planar no-slip wall [124], it is possible Blake's image scheme could be mimicked using image wavelets placed on the far side of the wall. In this case first guesses at an image scheme have so far failed so it might need either a creative image system or a clever choice of mother wavelet.

An alternative, more expensive approach with much wider applicability is to adapt the particle freezing method used in dissipative particle dynamics [125], which essentially builds the wall explicitly out of particles that enforce zero velocity at a no-slip boundary by being fixed in place. Some progress in WMCD was made in this direction, placing particles at the boundary with motion restricted to match the boundary condition. E.g. placing particles in a tight 3D potential well for no-slip conditions or confining them to a plane in liquid-liquid interfaces to suppress transverse flows while allowing parallel flows.

5

Time-dependent polymer diffusivity

5.1 Introduction

This chapter uses WMCD to investigate the diffusivity of polymers in (ultra-)dilute solution. This is something that seems fairly simple at first glance; just simulate a single polymer chain of N beads in equilibrium and look at the statistics of how it moves. Indeed within the Rouse model [8, 66], where hydrodynamic interactions are absent, with monodisperse beads the story *is* very simple. In the centre of mass Langevin equation

$$\mathbf{V} = \frac{1}{N}\sum_{i,j=1}^{N} \boldsymbol{\mathcal{G}}_{ij}\cdot\mathbf{F}_j + \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\xi}_i \tag{5.1}$$

the causal terms with $i \neq j$ vanish. The remaining causal terms sum to zero as only internal pair forces are present and all the self-mobilities are equal.

Using the Green-Kubo relation the chain diffusivity in the Rouse model is easily calculated to be

$$\begin{aligned} D_{\text{Rouse}} &= \frac{1}{3}\int_0^\infty dt\langle \mathbf{V}(t)\cdot\mathbf{V}(0)\rangle \\ &= \frac{1}{3N^2}\sum_{i,j=1}^{N}\int_0^\infty dt\langle \boldsymbol{\xi}_i(t)\cdot\boldsymbol{\xi}_j(0)\rangle \\ &= \frac{k_BT}{3N}\text{Tr}[\boldsymbol{\mathcal{G}}_{ii}] = \frac{D_{\text{bead}}}{N}. \end{aligned} \tag{5.2}$$

This result is pleasingly simple, but also misleading. When hydrodynamic interactions are considered the story becomes much more complex because the fluid flow leads to a non-zero causal velocity, despite still having zero net force. The only exception is a symmetric dimer, i.e. an $N = 2$ polymer with beads of equal size. Calculations on anything from $N = 3$ upwards require averages over configuration space.

Exactly what effect the non-zero velocity has - be it from causal or random forces - depends on the time scale considered. On time scales short enough for inertia to be important the persistence of the fluid flow also leads to a persistence in the polymer motion so that the measured diffusivity *increases* over time [126, 127, 128, 129]. In 3 dimensions the autocorrelation of the centre of mass velocity decays over time as $t^{-3/2}$, matching the decay of the underlying fluid flow after an impulse force [130].

While inertial time scales have received considerable attention, there has been a paucity of work at diffusive time scales. The go-to calculation of diffusivity in this regime follows the procedure used above to find D_{Rouse}, including ignoring the causal terms in $\mathbf{V}$. The end result, called the Kirkwood diffusivity [120], is

$$D_K = D_{\text{Rouse}} + \frac{k_B T}{3N^2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \langle \text{Tr}[\boldsymbol{\mathcal{G}}_{ij}] \rangle \tag{5.3}$$

where the continued presence of $D_{\text{Rouse}} \sim N^{-1}$ will later play an important role as the leading correction to scaling in the other terms, which are often written in terms of the chain hydrodynamic radius $R_H \sim N^{\nu}$ as $k_B T/(6\pi\eta R_H)$. In the meantime, the major advantage of D_K that it is calculable from a configurational average over $\boldsymbol{\mathcal{G}}_{ij}$ without any dynamical information.

The basis for ignoring the causal part of $\mathbf{V}$ lies in the powers of δt in discrete version of the Langevin equation shown back in Eq. (2.39); the $\sqrt{\delta t}$ of the noise leads this to dominate over the causal terms for $\delta t \to 0$. D_K is therefore the diffusivity one expects to find when measuring over fluid relaxation time scales. These time scales only capture a fraction of polymer behaviour however, and when viewed on longer time scales the diffusivity can change. Exact analytic calculations are only possible by making the preaveraged approximation [68, 67, 8]. This calculation was made by Fixman for Gaussian chains [131] and recently generalised to chains with other Flory exponents by Selby [132], and predicts the polymer diffusivity decreases by less than 2% in the limit of infinitely long chains.

To fully explore polymer dynamics one has to look out to the chain's longest relaxation time, often called the Zimm time τ_z, across which the chain's configuration completely decorrelates with itself. In doing so Fixman [131, 133, 134] and later Liu

and Dünweg [78] both found the chain diffusivity to *decrease* from D_K, albeit only by a few per cent.

The challenges in gathering this data are 3-fold. First is the usual difficulty of rising computational cost with system size. With a single chain Brownian dynamics algorithms win out over explicit solvent techniques [38], but even then that entails a cost rising as $N^{2.25}$ without the recent sophisticated methods described in Section 2.5. On this front WMCD's linear complexity for fractal single-chain systems is a distinct advantage. The second difficulty is unavoidable. Not only does the computational cost rise, but so does $\tau_z \sim N^{3\nu}$. Together that means the computation time effectively increases as

$$C \sim N^{1+3\nu}; \qquad 1+3\nu = \begin{cases} 2.5 & \text{Gaussian chain} \\ 2.764 & \text{Swollen chain} \end{cases} . \tag{5.4}$$

Finally, identifying the change in diffusivity requires picking out a signal of a few per cent in primarily noise-driven dynamics, and doing so with small errors in a decaying velocity autocorrelation all the way out to τ_z. This requires simulations over a large number of uncorrelated chains.

With all of these hurdles it is understandable that previous work was limited to fairly modest chain lengths (capping out at 200), still without gleaning a full understanding, and despite "the overall effort... in terms of single-processor time [being] estimated as roughly 6.7 years" (quoting Liu) [78][1]. This chapter aims to help fill in the gaps in understanding with the most comprehensive study to date, for the first time looking at chains in both θ and good solvent alongside one another and putting a greater focus on the velocity autocorrelation.

5.2 Fixman's expression for the change in diffusivity

For consistency with previous work the notation in this section follows that in Ref. [78], in which the causal and stochastic sums in the Langevin equation are called **A** and **B** respectively. Eq. (5.1) is then written rather succinctly as

$$\mathbf{V}(t) = \frac{1}{N}\mathbf{A}(t) + \frac{1}{N}\mathbf{B}(t). \tag{5.5}$$

Having brushed most of the notational baggage under the rug the centre of mass velocity autocorrelation splits into four clean correlation functions:

$$N^2\langle \mathbf{V}(t)\cdot\mathbf{V}(0)\rangle = \langle \mathbf{A}(t)\cdot\mathbf{A}(0)\rangle + \langle \mathbf{A}(t)\cdot\mathbf{B}(0)\rangle + \langle \mathbf{B}(t)\cdot\mathbf{A}(0)\rangle + \langle \mathbf{B}(t)\cdot\mathbf{B}(0)\rangle. \tag{5.6}$$

[1] For comparison, the single-processor time for the data in this chapter was about 2.25 years.

The last of these will ultimately lead to D_K and does not warrant further discussion, while $\langle \mathbf{B}(t) \cdot \mathbf{A}(0) \rangle = 0$ as $t \geqslant 0$ and the noise is completely uncorrelated with the past and present[2].

The other cross-term has previously been shown to obey $\langle \mathbf{A}(t) \cdot \mathbf{B}(0) \rangle = -2\langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle$ numerically in unbounded systems [78] and systems confined to a channel [135]. Showing this analytically requires a detour that is best served by briefly returning to the single particle Langevin equation with its usual notation. First it is necessary to argue the Langevin process is time reversible in equilibrium, i.e that it is equally likely to move a system from state x_1 to state x_2 as it is to do the reverse. Accounting for the probabilities of being in these states to begin with this amounts to seeking

$$P(x_1 \to x_2 \,|\, x_1)P(x_1) = P(x_2 \to x_1 \,|\, x_2)P(x_2). \tag{5.7}$$

This is none other than the detailed balance condition, so inasmuch as the Langevin equation obeys detailed balance, which it certainly will when implemented through a Monte Carlo simulation like WMCD, it describes a time reversible process. Using this the time-reversed Langevin equation can be written

$$\mathbf{v}_i^R(t^R) = \frac{d\mathbf{r}_i^R}{dt^R} = \sum_j \boldsymbol{\mathcal{G}}_{ij}(\mathbf{r}_{ij}^R) \cdot \mathbf{F}_j(\mathbf{r}_j^R) + \boldsymbol{\xi}_i^R(t^R), \tag{5.8}$$

with $t^R = -t$ and $\mathbf{r}^R(t^R) = \mathbf{r}(t)$, and equate $\mathbf{v}_i^R(t^R) = -\mathbf{v}_i(t)$. Abbreviating the causal sum to $\boldsymbol{\Sigma}_i^R(t^R) = \boldsymbol{\Sigma}_i(t)$, this implies

$$\boldsymbol{\xi}_i^R(t^R) = -\boldsymbol{\xi}_i(t) - 2\boldsymbol{\Sigma}_i(t). \tag{5.9}$$

Noise including a causal contribution is surprising and is easily misinterpreted. Neither noise would actually be generated with knowledge of $\boldsymbol{\Sigma}_i$, but because the causal term is symmetric not anti-symmetric under time-reversal the forwards and backwards noises must differ so as to compensate for this. More insightfully Eq. (5.9) leads to $\boldsymbol{\xi}_i^R$ being independent of its own past (i.e. the future in forwards time) for which the contribution from $\boldsymbol{\Sigma}_i$ removes the correlations in $\langle \boldsymbol{\xi}_i(t) \cdot \boldsymbol{\Sigma}_i(t' > t) \rangle$, which is non-zero going forwards because the causal forces have to work to undo thermal fluctuations. Substituting Eq. (5.9) into the time-reversed correlation leads to the main result

$$\langle \boldsymbol{\xi}_i^R(t^R) \cdot \boldsymbol{\Sigma}_i^R(t'^R) \rangle = 0 = -\langle (\boldsymbol{\xi}_i(t) + 2\boldsymbol{\Sigma}_i(t)) \cdot \boldsymbol{\Sigma}_i(t') \rangle \qquad t^R > t'^R \tag{5.10}$$

[2]The Itô term in the Langevin equation remains absent because it is zero, not because it is forgotten.

$$\Rightarrow \qquad \langle \boldsymbol{\xi}_i(t) \cdot \boldsymbol{\Sigma}_i(t') \rangle = -2 \langle \boldsymbol{\Sigma}_i(t) \cdot \boldsymbol{\Sigma}_i(t') \rangle \qquad t < t'. \tag{5.11}$$

Analogous results can also be found for Langevin equations with inertial terms and with a mobility tensor with memory [2]. These cases will play no role in this thesis, so now attention returns to the polymers, where Eq. (5.11) can be summed to give the sought result for the centre of mass displacements:

$$\langle \mathbf{A}(t) \cdot \mathbf{B}(0) \rangle = -2 \langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle. \tag{5.12}$$

Finally the velocity autocorrelation can be expressed as

$$N^2 \langle \mathbf{V}(t) \cdot \mathbf{V}(0) \rangle = \langle \mathbf{B}(t) \cdot \mathbf{B}(0) \rangle - \langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle. \tag{5.13}$$

Integrating this up to the diffusivity then leads to Fixman's expression [131]

$$D = D_K - \Delta D; \tag{5.14}$$

$$\Delta D = \frac{1}{3N^2} \int_0^\infty dt \langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle. \tag{5.15}$$

Even before doing anything with this two important properties want emphasising. Firstly, by separating out ΔD from D_K it can be accessed directly rather than as a change in the total diffusivity. The data will still contain noise from the fluctuations in configurations averaged over, but by no longer needing to compete with D_K the signal is much stronger.

Second is the clear result that D can only decrease from D_K since the integral over an autocorrelation function must be positive[3]. A more intuitive understanding can be found by comparing to the case where the polymer beads initially adopt the same configuration but their bonds are severed. D_K does not see the causal forces and hence does not distinguish between these two cases, but over finite time the two cases clearly differ: the beads in the bond-less case simply diffuse away while the forces in the connected case work to undo this diffusion and hold the chain together. It is then clear that $\langle \mathbf{A}(t) \cdot \mathbf{B}(0) \rangle$ must be negative, and hence so must ΔD.

5.3 Data acquisition in MC algorithms

In some sense, WMCD is an unnatural choice of algorithm for measuring $\mathbf{A}$ because the mobility tensor appears in WMCD only implicitly via the move choice. Compare this to BD, where $\boldsymbol{\mathcal{G}}$ is calculated as part of the update algorithm and consequently

[3]This comes from the Wiener-Khinchine theorem which shows the Fourier transform of an autocorrelation function is a power spectral density [136], and hence that it must be non-negative everywhere, including at $k = 0$.

A is obtained without additional calculation.

While it is not a complicated task to simply write a function in the WMCD code to calculate **A** directly, doing so raises the cost scaling from $\sim N$ to $\sim N^2$ and is preferably avoided. Fortunately the details of WMCD can be exploited to obtain **A** without raising the cost scaling, and this section describes how to do so with both smart and non-smart versions of the algorithm.

5.3.1 Calculating A in smart WMCD

With a bit of care **A** can be obtained in smart WMCD with nothing more than what is already calculated as normal. As was shown in Section 4.1.3 the mean displacement of any particle in a smart WMCD move is equal to $\sum_j \boldsymbol{\mathcal{G}}_{ij} \cdot \mathbf{F}_j \, \delta t$, and consequently the mean centre of mass displacement is $\mathbf{A}\delta t/N$.

The hitch is that the actual displacement includes move variance from $\boldsymbol{\xi}$ and for a valid simulation this cannot be removed. To extract the causal component of the displacement ghost moves are added between generating move parameters and updating the system. The modified procedure is as follows:

> Generate move parameters as usual except **Q**.
> Set $\mathbf{Q} = \mu\delta t\boldsymbol{\mathcal{F}}$ and make the corresponding move.
> Calculate centre of mass displacement from this move and hence find **A**.
> Find **Q** again with variance included and update the system as normal.

This procedure adds to the cost of the simulation by adding an extra move step, but does not destroy the linear scaling with system size and is in practice a small fraction of the total cost.

5.3.2 Calculating A in non-smart WMCD

Without smart MC biasing **Q** around causal forces, WMCD (and hydrodynamically coupled MC methods in general, for all of which the below applies) moves provide only the variance and do not have direct access to the mean. Nor are forces calculated to begin with, but they do calculate energy changes for the acceptance test which is closely related to forces via the work done

$$\Delta U \approx -\sum_j \delta\mathbf{r}_j \cdot \mathbf{F}_j. \tag{5.16}$$

The rest of this subsection sets out how to estimate **A** using this.

First the centre of mass displacement in a move is written in terms of the attempted move's displacement $\delta\mathbf{R}_0$ as

$$\delta\mathbf{R} = \delta\mathbf{R}_0 \, \Theta(P_{\text{acc}} - \texttt{rand}); \qquad \texttt{rand} \in (0, 1], \tag{5.17}$$

where Θ accounts for the move acceptance test. A useful property is $\langle\Theta\rangle_{\texttt{rand}} = P_{\text{acc}}$. A subsequent Taylor expansion of P_{acc} then brings Eq. (5.16) into Eq. (5.17). For this purpose the Metropolis test cannot be used because it is not smooth at $\Delta U = 0$, and instead the better behaved Glauber test should be used, giving

$$\begin{aligned} P_{\text{acc}} &= \frac{1}{2}\left(1 - \tanh\left(\frac{1}{2}\beta\Delta U\right)\right) && (5.18)\\ &\approx \frac{1}{2}\left(1 + \frac{1}{2}\beta\sum_j \delta\mathbf{r}_j \cdot \mathbf{F}_j\right). && (5.19) \end{aligned}$$

Next $\delta\mathbf{R}$ is split into terms from P_{acc} interpreted as being the causal and noise contributions:

$$\begin{aligned} \delta\mathbf{R} &= \delta\mathbf{R}_A + \delta\mathbf{R}_B; && (5.20)\\ \delta\mathbf{R}_A &= \delta\mathbf{R}_0 \frac{\Theta}{P_{\text{acc}}}\left(P_{\text{acc}} - \frac{1}{2}\right); && (5.21)\\ \delta\mathbf{R}_B &= \delta\mathbf{R}_0 \frac{\Theta}{2P_{\text{acc}}}. && (5.22) \end{aligned}$$

The sole focus is now on $\delta\mathbf{R}_A$, which should be an estimate of the mean displacement measured in the smart WMCD procedure in the previous subsection. To confirm this, $\delta\mathbf{R}_0$ is written as $N^{-1}\sum_i \delta\mathbf{r}_i$, leading to

$$\begin{aligned} \langle\delta\mathbf{R}_A\rangle &\approx \frac{\beta}{4N}\frac{\langle\Theta\rangle_{\texttt{rand}}}{P_{\text{acc}}}\sum_{ij}\langle\delta\mathbf{r}_i(\delta\mathbf{r}_j\cdot\mathbf{F}_j)\rangle \\ &= \frac{\beta}{4N}\sum_{ij}\langle\delta\mathbf{r}_i\otimes\delta\mathbf{r}_j\rangle\cdot\mathbf{F}_j \\ &= \frac{\delta t}{2N}\sum_{ij}\boldsymbol{\mathcal{G}}_{ij}\cdot\mathbf{F}_j = \frac{\mathbf{A}\delta t}{2N}. && (5.23) \end{aligned}$$

This is a factor of $1/2$ smaller than the smart WMCD result, which comes from the above calculation's assumption that the same δt is added after every attempted move, whereas in reality time in WMCD is defined by (unhindered) motion and factors in P_{acc}, which averages to $1/2$ with the Glauber test. Hence this $\delta t/2$ is equivalent to the δt from before, and the two methods match.

The accuracy of the two approaches is limited by the accuracy of any given WMCD simulation, but owing to having a higher order of algorithm (as discussed back in Section 2.6.4) and not needing to truncate a Taylor series the smart WMCD approach is taken to be the more accurate of the two and will be the only one used for data collection in the rest of the chapter. To support this, the ratio of

$$C_{AA}(t) = \langle\mathbf{A}(t)\cdot\mathbf{A}(0)\rangle \qquad (5.24)$$

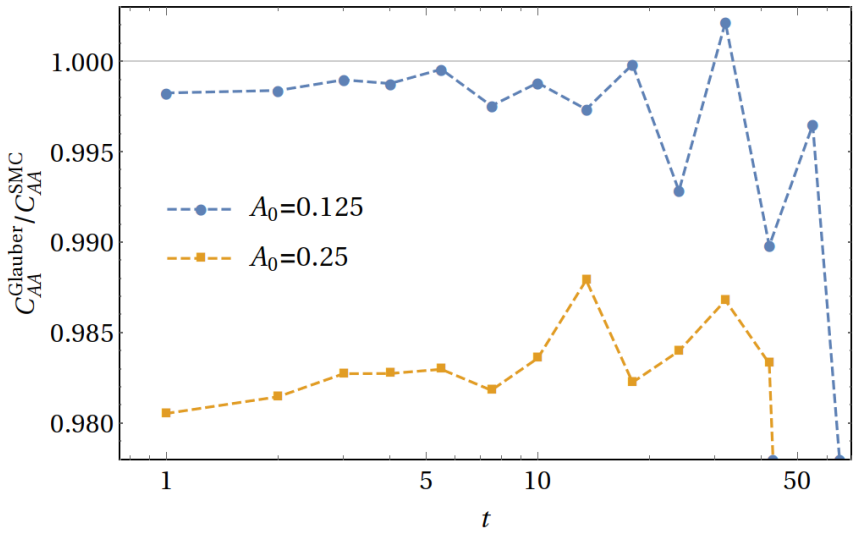

Figure 5.1: Autocorrelation of **A** measured in non-smart Glauber WMCD relative to smart WMCD for a Gaussian chain with $N = 20$. Two values of the move amplitude A_0 were used to reveal the dependence of accuracy on time step.

in Glauber and smart MC simulations are shown in Fig. 5.1.

The first thing to note in this figure is that, although the two differ, the fractional error is not large, being less than 2% in the larger timestep, and that the lines are fairly flat until noise in the data becomes significant at larger t. This all implies the Glauber approach captures the same physics as smart MC, just with slight underestimates of C_{AA}. Moreover, an estimate of the error can be made based on the next lowest order term in P_{acc} which scales as $\delta r^3 \sim A_0^3$, suggesting a factor of approximately 8 between the errors in Fig. 5.1, which is close to what is observed.

5.4 Simulation details and polymer systems

5.4.1 Potentials and units

This chapter simulates polymers in both good and θ solvents, referred to as swollen and Gaussian chains respectively. The details for the former are identical to those in Section 3.5.1 except for a slight reduction of k_BT to 1. Repeating the relevant quantities for convenience, the bead hydrodynamic radius is $a = 0.302$ and the WCA and FENE potentials use the parameters: $\sigma = 2^{-1/6}$; $\epsilon = 1$; $R_0 = 2\sigma \approx 1.782$; and $k_{\text{FENE}} = 7\epsilon/\sigma^2 \approx 8.819$ (all in implicit code units).

The Gaussian chain data use the same values for a and k_BT, and hence use the same length and energy units. As per Section 2.4.2 the WCA and FENE

potentials are replaced by the quadratic bond potential

$$U_{\text{quad}} = \frac{1}{2} k_{\text{quad}} \left(\mathbf{r}_{i+1} - \mathbf{r}_i \right)^2 , \tag{5.25}$$

while the excluded volume interactions are removed altogether. The spring constant is set to $k_{\text{quad}} = 2.736$ to give the root-mean-square bond length $\sqrt{3} \times 2a \approx 1.046$, similar in value to the r.m.s. bond length in the swollen chains of 0.965. While not a perfect match, it does make comparisons between the two chain types easier. As always, the time unit for both types of system is set by bead diffusion: $\tau = \pi \eta a^3 / k_B T$.

5.4.2 Other system perameters

One of the main aims in the following sections is to look for universal behaviour of polymers of different length. As mentioned earlier, $N = 3$ is the smallest chain for which $\Delta D \neq 0$ so chains of this length upwards are considered. Ideally this would extend to asymptotically large N were the rising computational cost ($\sim N^{3\nu+1}$) not so limiting. The longest chains for which data is presented equal those of Liu and Dünweg at $N = 200$. Collecting enough data for the $N = 200$ swollen chains (approx. 4000 configurations) alone took 1.12 years of processor time, almost exactly half of the total 2.25 years of computing time required to collect the data in this chapter.

Aside from chain lengths, other relevant details include:

- The systems were unbounded with only a single polymer in each.
- Smart WMCD was used with $A_0 = 0.25$ for most data, with move rejection rates of order 10^{-3} and 10^{-5} for swollen and Gaussian chains respectively. These are sufficiently small to give high fidelity results.
- The systems were monodisperse, all beads with the same $a = 0.302$.
- The cubic wavelet was used.
- The version of WMCD used did not include rotations.

All other parameters in WMCD can be derived from those listed above (e.g. δt) or have no influence on the results (e.g. L because the unbounded Fourier moves are used).

5.5 Properties of scaling variables

The analysis of C_{AA} will seek universal behaviour by scaling the data. In preparation for that, this section presents and discusses the quantities used.

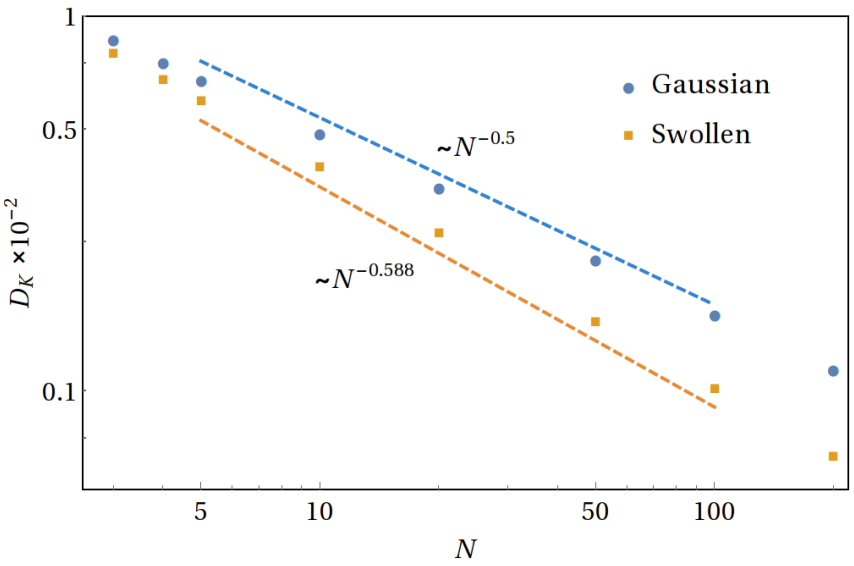

Figure 5.2: Kirkwood diffusivity against chain length. Dashed lines indicate the theoretical scaling in the long-chain limit.

5.5.1 Scaling of the Kirkwood diffusivity

The Kirkwood diffusivity is well known to scale with chain length as $N^{-\nu}$ [8]. Fig. 5.2 shows this is seen in the systems simulated. The most noteworthy part of this is not that D_K scales, but that it does so for such short chains, especially in swollen chains where the scaling seems to hold all the way down to $N = 3$.

The reason for this stems from results of Ref. [137] which expands the off-diagonal sum in Eq. (5.3) to give

$$D_K = \frac{k_B T}{6\pi\eta a}\left[N^{-1} + a\left(c_\nu N^{-\nu} - c_1 N^{-1} + \cdots\right)\right], \tag{5.26}$$

where the first term is simply D_{Rouse} and the terms in the parentheses collectively give R_H^{-1}. While the slowest decaying $N^{-\nu}$ term dominates in long chains, the N^{-1} terms are expected to be a major correction in short to moderate length chains. Due to their differing signs however, it is possible to set the system up to cancel the N^{-1} terms and thereby observe asymptotic scaling in short chains. The exact values for the coefficients c_ν and c_1 vary with chain parameters, but using the values found in Ref. [137] as a rough estimate finds both $ac_\nu \approx ac_{-1} \approx 1$ for our particle sizes. It is therefore very possible the $N^{-\nu}$ term dominates.

5.5.2 Scaling of the Zimm time

The Zimm time and its scaling with chain length has already made an appearance in explaining the computational challenges of gathering data for long chains. Here

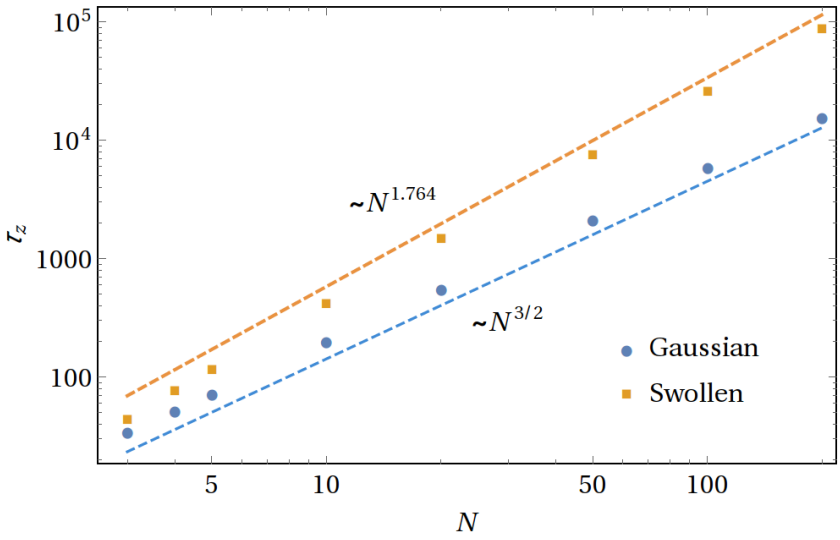

Figure 5.3: Zimm time against chain length. Dashed lines indicate the theoretical scaling in the long-chain limit.

it is presented more formally by invoking Rouse modes[4] [8]

$$\mathbf{X}_p = \frac{1}{N}\sum_{i=1}^{N} \mathbf{r}_i \cos\left(\frac{p\pi}{N}\left(i - \frac{1}{2}\right)\right). \tag{5.27}$$

Of these the $p = 1$ mode, which describes the largest scale features in the chain, is slowest to relax and therefore provides a definition of the Zimm time through

$$\langle \mathbf{X}_1(t) \cdot \mathbf{X}_1(0) \rangle \propto e^{-t/\tau_z}. \tag{5.28}$$

Fig. 5.3 shows τ_z obtained from exponential fits of $\langle \mathbf{X}_1(t) \cdot \mathbf{X}_1(0) \rangle$. As with the D_K data this is very quick to reach the known asymptotic scaling of $\tau_z \sim N^{3\nu}$, although this time the reasons are less buried: τ_z is dominated by the lowest mode(s) by definition, and hence is influenced little by the local effects that drive the corrections in D_K.

5.5.3 Scaling of $\langle A^2 \rangle$

The zero-time value $C_{AA}(0) = \langle A^2 \rangle$ is much less well studied than either D_K or τ_z, with only Liu and Dünweg having studied it previously [78]. Alongside data reproduced in Fig. 5.4(b) they provided a qualitative argument based on the chain being in thermal equilibrium and therefore each bead sees a statistically uncorrelated

[4]While Rouse modes are not strictly the normal modes in the Zimm model, nor even in the Rouse model with excluded volume interactions, it is common practice to use them in analytical calculations because they are expected to be fairly close to the true normal modes in the system.

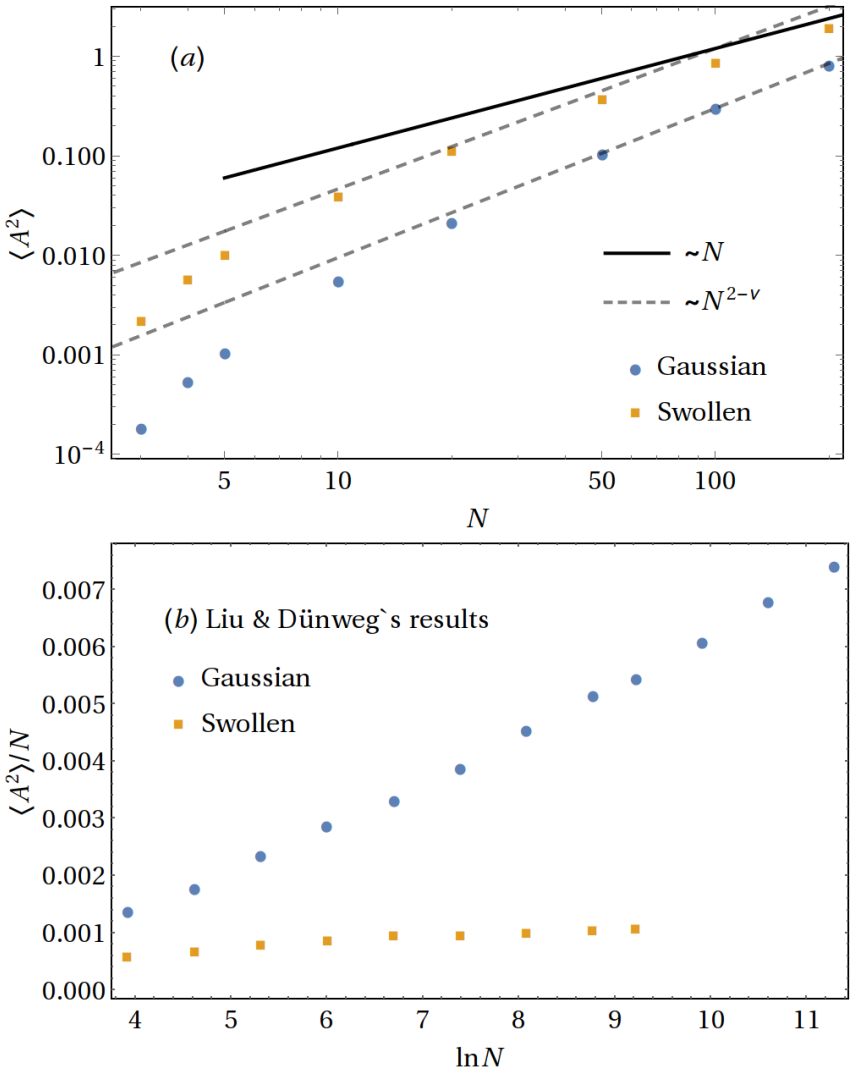

Figure 5.4: Plots showing the dependence of $\langle A^2 \rangle$ on chain length. (a): Data taken in this work with WMCD. The dashed lines are purely to guide the eye and are not fits. (b): Data recreated from Fig. 4 in Ref. [78], after redefining units to match those used here and plotting to highlight the presence (or not) of a logarithmic factor. Note the systems are still not identical so this only brings values to the same order of magnitude.

flow "of order of a typical thermal velocity". $\langle A^2 \rangle$ would therefore be a sum of roughly equal and independent contributions along the chain, leading to $\langle A^2 \rangle \sim N$.

The data gathered in WMCD in Fig. 5.4(a) does not extend to such long chains (the chains used for Fig. 5.4(b) had been generated in hydrodynamically uncoupled simulations making it a much faster process), though where the two domains do overlap the values of $\langle A^2 \rangle$ are within a factor of order unity of each other. This is as close as one could expect given the system parameters, including potentials used and the ratio of bond length to bead radius, do not match.

The deviation from predicted scaling in small N regime motivates a more careful explanation than the one presented by Liu and Dünweg, which offers no clues about the origin of the deviation. While the local flow velocity can be expected to be typical of thermally-driven flows, one must remember $\mathbf{A}$ comes entirely from causal forces, allowing a more precise understanding. In the chain, all forces are part of an equal and opposite pair on average separated by one mean bond length. At large distances this looks like a dipole force, whose flow field decays as r^{-2} and can be considered local. However, this still has some range over which velocities have significant correlations, which could loosely be cast as a blob radius. Each blob can then be considered statistically uncorrelated and Liu and Dünweg's argument applies for blobs rather than beads.

Chains shorter than a few blobs are expected to have significant corrections to scaling. Furthermore, as beads are added to short chains $\langle A^2 \rangle$ is expected to rise faster than linearly because these beads add to the not yet converged contributions from the beads already there, as well as contributing their own forces. Hence the steep initial increase in Fig. 5.4(a) has a qualitative understanding.

The story still needs more care in long chains too. The flow field decaying as r^{-2} does not guarantee its sum along the arc s of the chain converges unless $r_{ij}^{-2} \sim s^{-2\nu}$ decays faster than s^{-1}. This is true for swollen chains where $\nu \approx 0.588$ but Gaussian chains are the edge case with $\nu = 1/2$. Consequently an additional logarithmic factor is expected in $\langle A^2 \rangle$ for Gaussian chains. The data in Fig. 5.4(a) is not sufficient to confirm this, but it is clearly seen in Fig. 5.4(b) by $\langle A^2 \rangle / N$ being linear with $\ln N$. The same figure also confirms there is no logarithmic factor in the swollen chain data.

5.6 Short-time behaviour of $\langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle$

Attention now turns to the behaviour of C_{AA}. Starting with a simple normalisation of data, Fig. 5.5 plots $C_{AA}(t)/\langle A^2 \rangle$ against time, which is guaranteed to collapse data at $t = 0$. The continued collapse of data at larger t will be discussed in detail shortly, but even at a first glance it helps identify two distinct regimes: First is a

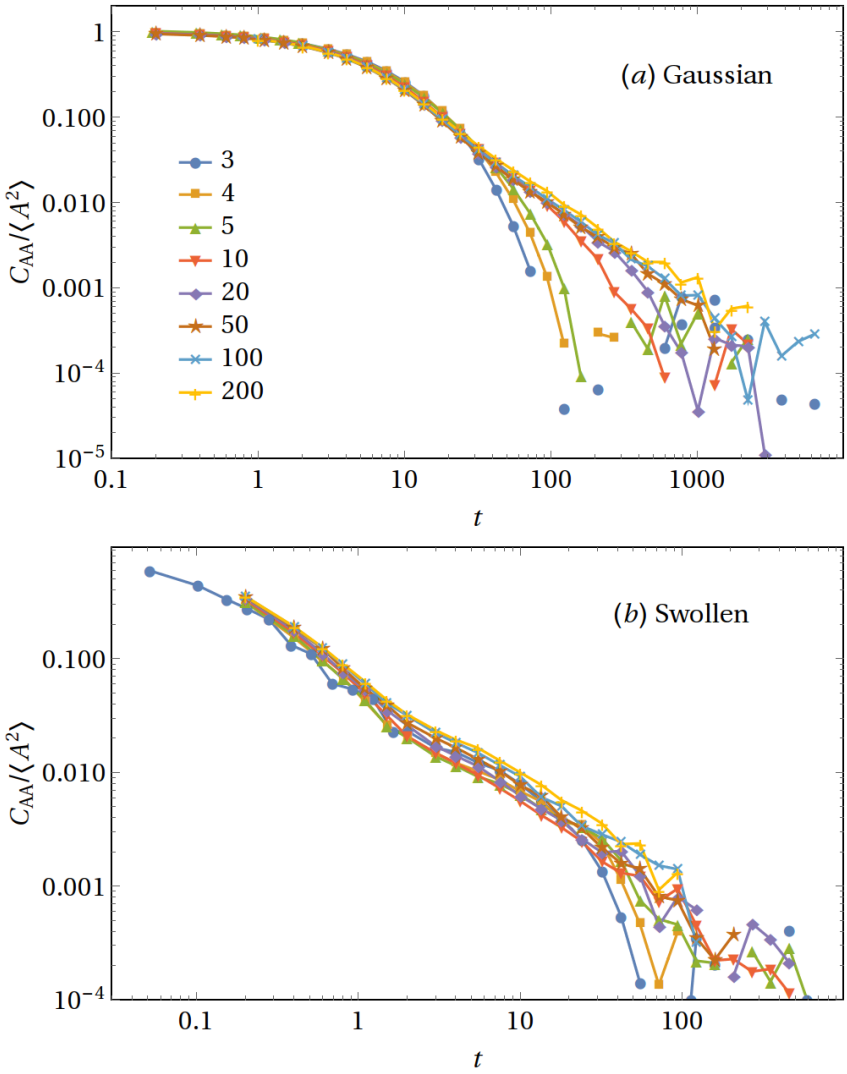

Figure 5.5: Plots of C_{AA} normalised by the values of $C_{AA}(0) = \langle A^2 \rangle$ in Fig. 5.4(a).

short-time regime with an exponential decay out to $t \approx 30$ in the Gaussian data. In the swollen data this regime appears to end at $t \approx 2$ instead, although it will be seen that it also continues out to $t = O(10)$. This short-time regime then segues into a long-time power law decay which does not collapse so well with the present scaling. This long-time regime will be discussed in the next section.

Returning to the short-time regime, it is evident from the continued collapse of the data that the underlying physics is the same in chains of all lengths, including $N = 3$. While calculations with $N = 3$ are still intractable, the tiny size of the system is a blessing when trying to build an intuition of what is going on. While hydrodynamics complicates the story, it does not change the fact everything in $\mathbf{A}$ boils down to particle separations $\mathbf{r}_{ij}$ and forces $\mathbf{F}$. Anticipating a crude investigation of

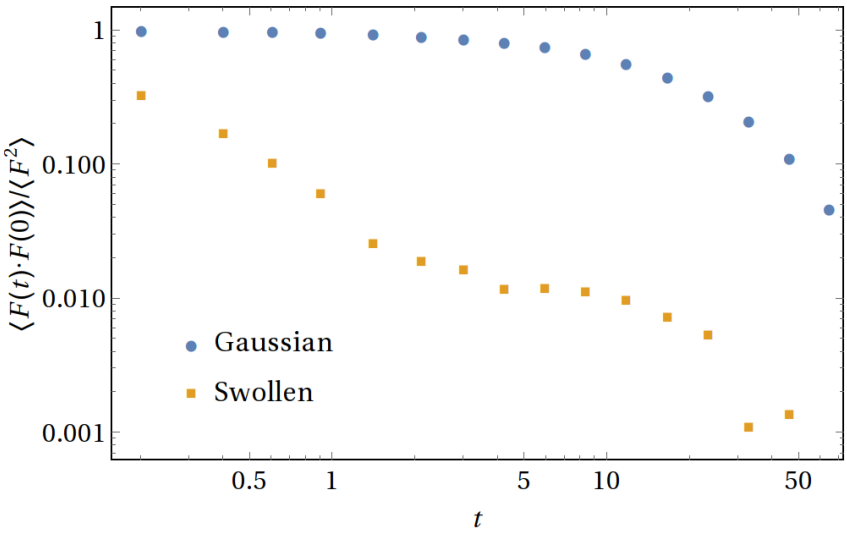

Figure 5.6: Autocorrelation of forces in dimers ($N = 2$) using the same potentials as the usual Gaussian and swollen chains, normalised by the static value.

the latter, Fig. 5.6 plots the force autocorrelation

$$C_{FF}(t) = \langle \mathbf{F}(t) \cdot \mathbf{F}(0) \rangle = \langle |\mathbf{F}(t)|\, |\mathbf{F}(0)| \rangle \, \langle \hat{\mathbf{F}}(t) \cdot \hat{\mathbf{F}}(0) \rangle \tag{5.29}$$

in a dimer. Contrary to expectation this contains almost all the features seen in Fig. 5.5 for $t \lesssim 50$, including: the fast initial decay in the swollen chains; the very slow initial decay in the Gaussian chains; the kink in swollen chain data at $t \approx 2$; and the exponential roll-off in both chain types for $t \gtrsim 10$. Given how important these features are to C_{AA} and how easy they are to make sense of now they can be attributed to forces in a dimer, it is worth describing the physical processes involved in each of these.

First and easiest is the rapid initial decay of C_{FF} in swollen chains, which is simply the result of bonds changing lengths and consequently changing the amplitude of the force. With steep potentials (both repulsive and attractive) the bond lengths at thermal energies sit close to the potential minimum, allowing large changes in the value of $\mathbf{F}$, including its sign, for small displacements. From Fig. 5.6 it is evident that these forces relax on the time scale of 1 time unit. As a reminder this is how long it takes each particle to diffuse by its own radius, which fits with expectation if mean bond lengths are about 3 particle radii. The plateau that follows is a result of having an asymmetric potential well. With a softer wall on one side than the other $\langle |\mathbf{F}| \rangle \neq 0$ and consequently $\langle |\mathbf{F}(\infty)||\mathbf{F}(0)| \rangle = \langle |\mathbf{F}| \rangle^2 \neq 0$.

One might expect to find similar behaviour in the Gaussian chain C_{FF}, but the potential landscape here is very different. By having a much softer quadratic potential with a minimum at $|\mathbf{r}_{ij}| = 0$, large changes in the force require large

displacements. In practice thermal motion dominates over the attractive forces (it is after all what sets the mean bond length to 3 particle radii in this case) and, together with hydrodynamic coupling which already slows a decrease in bond length, prevents forces from changing as they do in swollen chains. C_{FF} is therefore essentially constant, and these last paragraphs can be summarised by

$$\langle |\mathbf{F}(t)|\,|\mathbf{F}(0)| \rangle \approx \langle F^2 \rangle \begin{cases} 1 & \text{Gaussian} \\ f(t) & \text{swollen} \end{cases} \tag{5.30}$$

with $f(t)$ a function decaying from 1 to $\langle |\mathbf{F}| \rangle^2 / \langle F^2 \rangle$.

So far only radial motion has been considered. Bond rotations also decorrelate forces and ultimately send C_{FF} to zero. Because the bond lengths are almost equal for both chain types, the rotational diffusion of each should match. This is indeed what is observed in the exponentially decaying roll-off in both sets of data in Fig. 5.6 for $t \gtrsim 10$. Hence

$$\langle \hat{\mathbf{F}}(t) \cdot \hat{\mathbf{F}}(0) \rangle \approx e^{-t/\tau_{FF}} \tag{5.31}$$

with $\tau_{FF} \approx 10$ for our data.

With all this, the physics of the short-time behaviour of C_{AA} is largely understood as the decorrelation of pair forces (whose hydrodynamic interactions construct $\mathbf{A}$). It will be helpful to suppose the full functional form can be written as a sum of short- and long-time parts as

$$C_{AA}(t) = C_{AA}^{\text{short}}(t) + C_{AA}^{\text{long}}(t), \tag{5.32}$$

for which the above discussion leads to

$$C_{AA}^{\text{short}}(t) \approx \frac{\langle A^2 \rangle}{\langle F^2 \rangle} C_{FF}(t) \approx \langle A^2 \rangle e^{-t/\tau_{FF}} \begin{cases} 1 & \text{Gaussian} \\ f(t) & \text{swollen} \end{cases} . \tag{5.33}$$

Note the full $\langle A^2 \rangle$ will contain some contribution from $C_{AA}^{\text{long}}(0)$ so there is really an extra multiplicative factor less than 1. The exact value is neither known nor needed as it would only be used for finding the scaling with N, and it will be seen that the amplitude of C_{AA}^{long} decreases or remains constant, so C_{AA}^{short} will always scale the same as $\langle A^2 \rangle$.

Importantly for ΔD, τ_{FF} is insensitive to N, so

$$\left. \frac{\Delta D}{D_K} \right|_{\text{short}} = \frac{1}{3N^2 D_K} \int_0^\infty dt\, C_{AA}^{\text{short}}(t) \propto \frac{\langle A^2 \rangle}{N^2 D_K}. \tag{5.34}$$

In the asymptotic limit this will scale as $R_H/N \sim N^{\nu-1}$ (ignoring the sub-dominant logarithm in the Gaussian $\langle A^2 \rangle$) such that this contribution will slowly vanish. However in the moderate length chains simulated with the more rapid increase in $\langle A^2 \rangle$ this contribution can actually be increasing. Returning to Fig. 5.4 this would be expected where the gradient is steeper than the dashed tangent lines.

To see if this is indeed the case, $\Delta D/D_K$ is plotted as a function of cut-off time (i.e. replacing the upper limit ∞ with t in Eq. (5.15)) in Fig. 5.7. For now attention is on the grey arc linking the data in both plots, marking the cross-over between the short and long-time contributions. While this rises with chain length with fairly small N, it is also seen to flatten off and, in the swollen chain data, begin to decrease again (the Gaussian data is expected to do the same with longer chains). This is all consistent with C_{AA}^{short} providing a correction to scaling of the form R_H/N, and the discussion will return to this later.

5.7 Long-time behaviour of $\langle \mathbf{A}(t) \cdot \mathbf{A}(0) \rangle$

The behaviour of C_{AA}^{long} is challenging to describe analytically. Multiple attempts with varying levels of sophistication were made over the course of this project, all using the idea that as time progresses a blob characterising the length of chain that has relaxed since $t = 0$ increases in size, leading to contributions from smaller length scales vanishing and the dominant contribution being associated with length scales of order the blob size. I.e. there is a correspondence between length and time scales. All of these calculations predicted a power law

$$C_{AA}^{\text{long}} \propto t^{-p}; \qquad p = \frac{4\nu - 1}{3\nu} = \begin{cases} 2/3 & \text{Gaussian} \\ 0.766 & \text{swollen} \end{cases}, \tag{5.35}$$

perhaps with an accompanying logarithmic factor in Gaussian chains[5]. As will be seen the existence of a power law fits the data but the exponents disagree, so this section takes a more phenomenological route to reach useful results instead of going through these calculations.

The power law tails in Fig. 5.5 could be fitted straight away, but it will be profitable to first search for a way to scale the data so as to collapse them onto a long-time master plot. To aid this, inspection of Fig. 5.7 reveals ΔD does not converge to its long-time value until τ_z for any chains with sufficiently clean data to reach this time. On the one hand this implies C_{AA}^{long} includes a factor $\exp(-t/\tau_z)$ to cut off the tail at τ_z, which makes sense since the chain configuration fully decorrelates on this time scale and C_{AA} very much depends on the configuration. Seeking a master curve that includes this final roll-off then means C_{AA}^{long} should be a function of t/τ_z.

[5]The origin of this is much the same as the logarithmic factor in $\langle A^2 \rangle$.

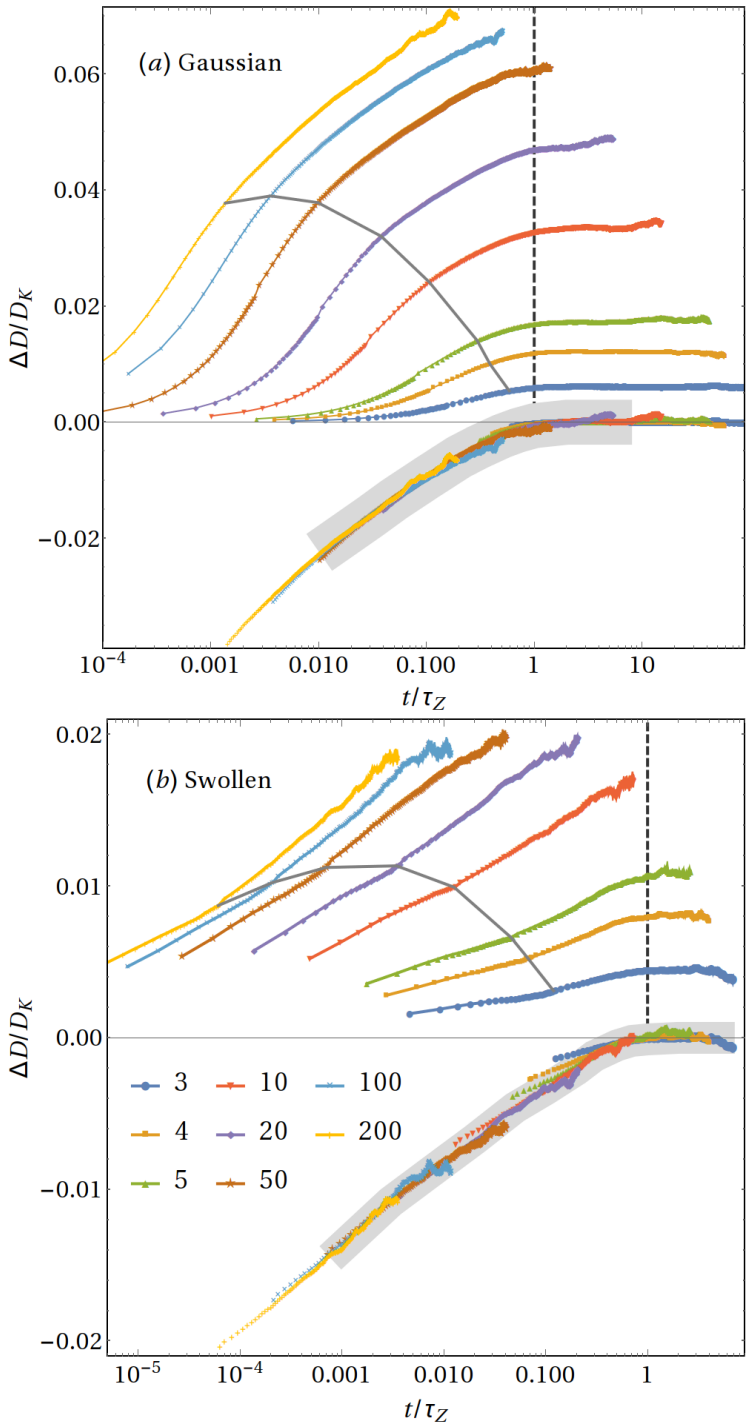

Figure 5.7: Above zero axis: fractional decrease in diffusivity plotted against scaled time so that each curve should plateau at $t/\tau_z = 1$ (marked by vertical dashed lines). The grey arc joining the data in each plot marks the transition between the short-time force dominated regime and long-time power law regime. Below axis: the same data translated down to construct master curves, omitting data from the short-time regime which does not collapse.

Knowing a power law is sought from the straight tails in Fig. 5.5, the ansatz function

$$C_{AA}^{\text{long}}(t) = \alpha \left(\frac{t}{\tau_z} + \psi \right)^{-p} e^{-t/\tau_z} \tag{5.36}$$

is considered, with α, p and ψ unknown positive constants. There is nothing profound about this choice of ansatz; it is simply a generic power-law decay that segues into an exponential roll-off. Even ψ, which might have been omitted, is only included to smooth off the asymptote at $t \to 0$ and hence ensure C_{AA}^{short} dominates here as per the data. In principle both α and ψ have some N-dependence, but reading this from the data is not possible because of the dominant presence of C_{AA}^{short} at short times where they would be apparent. The search for a master curve will address both parameters, and that starts by calculating

$$\left. \frac{\Delta D}{D_K} \right|_{\text{long}} = \frac{1}{3N^2 D_K} \int_0^\infty dt\, C_{AA}^{\text{long}}(t) \tag{5.37}$$

$$= \frac{\alpha \tau_z}{3N^2 D_K} e^{\psi}\, \Gamma(1-p, \psi) \tag{5.38}$$

$$\approx \frac{\alpha \tau_z}{3N^2 D_K} \begin{cases} \Gamma(1-p,0) - \dfrac{\psi^{1-p}}{1-p} & p \neq 1 \\ \ln\left(\psi^{-1}\right) - 0.577215... & p = 1 \end{cases} \tag{5.39}$$

in which $\Gamma(x, y)$ is the incomplete gamma function and the last line assumes $\psi \ll 1$.

From here it is supposed that the ratio $\Delta D/D_K$ will tend towards a finite value as $N \to \infty$, i.e. that it is independent of N once all the quantities contained within reach their scaling regime. Since the contribution from C_{AA}^{short} is expected to vanish, this imposes constraints on both α and ψ. It is useful to first assume ψ has no N-dependence so the messy p-dependent part can be put aside as a constant. This leads to trying

$$\alpha \sim N^2 D_K / \tau_z, \tag{5.40}$$

which in turn means plots of $(\tau_z/N^2 D_K) C_{AA}^{\text{long}}(t/\tau_z)$ ought to collapse onto a master curve, at least for $t/\tau_z \gg \psi$. Fig. 5.8 shows this is indeed the case for both Gaussian and swollen data, and now that the data are extended over several orders of magnitude they can be fitted with greater confidence, finding powers significantly more negative than those in Eq. (5.35). These discrepancies will be discussed shortly, but first the discussion of ψ needs rounding off.

With the observed power laws having $p \geqslant 1$ the question of the N-dependence of ψ cannot be ignored. Aside from being constant (or at least approximately so) ψ can either be an increasing or decreasing function of N. The former will ultimately fail because that will eventually lead to $\psi > 1$, corresponding to C_{AA}^{long} losing its power law altogether and hence must have lost whatever physical processes are

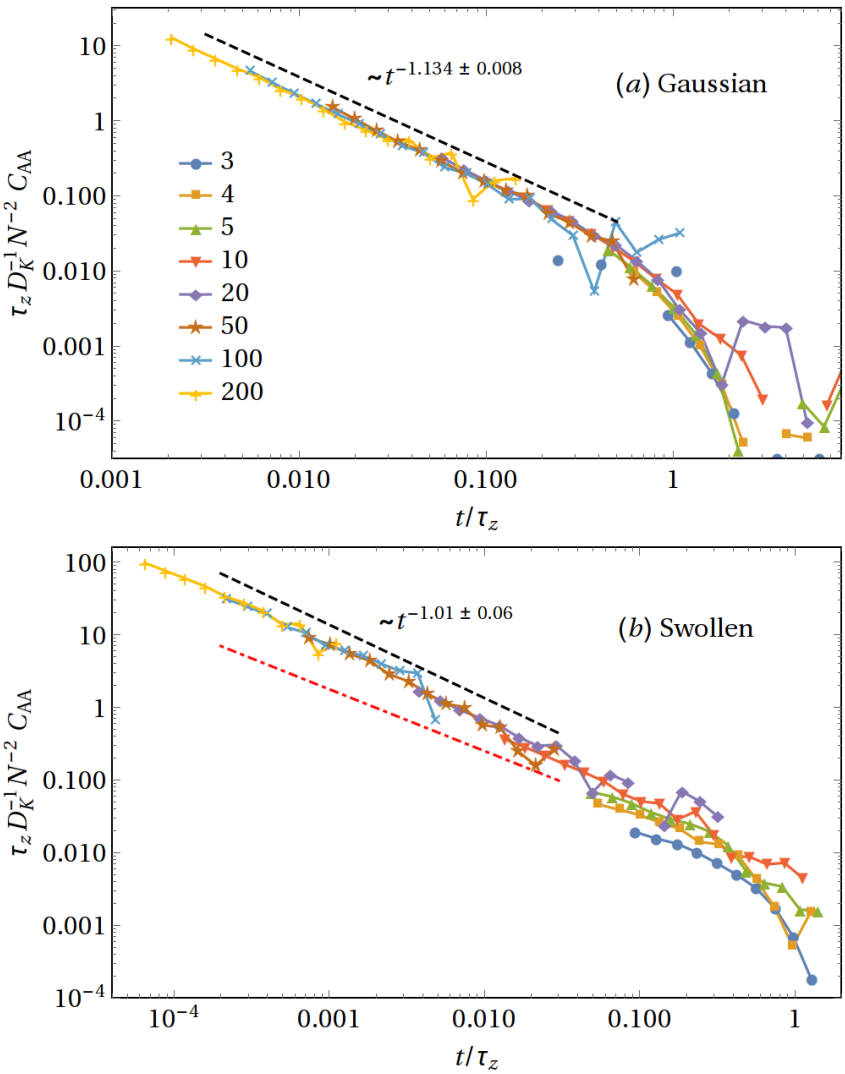

Figure 5.8: Plots of C_{AA}, scaled using Eq. (5.40), against time scaled by the Zimm times in Fig. 5.3. The grey dashed lines in both plots indicate the power law fitted to the data. The red dot-dashed line in (b) shows a shallower gradient plausible in a couple of the data sets.

driving the power law in the first place.

The possibility of ψ being a decreasing function of N hits similar problems. Because the data has already collapsed, $\Delta D/D_K$ would pick up a contributions from ψ. If $p < 1$ this would not be an issue since the constant $\Gamma(1-p, 0)$ would dominate; in fact in this case ψ might as well be taken as a constant since it is sub-dominant anyway. However, for $p \geqslant 1$ as observed in Fig. 5.8, having $\psi \to 0$ as N increases would lead to $\Delta D/D_K$ exceeding 1. That in turn implies a negative diffusivity which is impossible. Therefore ψ must limit to a finite and non-zero value, allowing the discussion to proceed as though it were independent of N.

With that sorted, how do the observed power laws compare to those in Eq. (5.35) and can they conceivably be brought in line? The discrepancy is especially large with Gaussian chains, although the possibility of a logarithmic factor in the analytic result does offer quite a lot of freedom to tinker with the exponent when viewed over a limited domain. However, even the best efforts failed to capture the apparent $t^{-1.134}$ power law over the 2 orders of magnitude it holds over in Fig. 5.8(a). Thus one has to conclude that the observed power law is real if not fully understood.

The existence of a pure power law in the swollen chain case is on firmer ground, but its value is less certain. The fitted exponent of -1.01 ± 0.06 seems to suggest an exact -1 is possible. It is also true that if you stare at Fig. 5.8(b) for long enough you can convince yourself that the data for some values of N is closer to running parallel with the red dot-dashed line than it is the fit of the master curve. This red line has an exponent -0.85, which is close enough to the analytic value of -0.766 to be worth entertaining. However, given not all of the lines seem to have this shallower gradient and wishing to avoid confirmation bias when there is clearly a discrepancy with theory anyway, the fitted exponent is taken as the simulations' result.

5.8 Long-chain limit of ΔD

Returning to $\Delta D/D_K$, an estimate of the long-chain, long-time value is desired. The data in Fig. 5.7 are not good enough to identify this directly as they continue to show N-dependence and are overcome by noise prior to $t/\tau_z = 1$ for the longer chains. A way to bypass throwing more computation time at the problem is to recognise the contribution from C_{AA}^{long} ought to form a master curve in the long-chain limit, and then assume the existing curves lie parallel to that since they are already parallel to each other. Translating each curve down to plateau at zero, and translating those without a plateau down to fall in line with the rest, leads to the estimate master plots shown under the $\Delta D/D_K = 0$ line in Fig. 5.7.

This alone it not enough to obtain the long-chain value as it is unknown

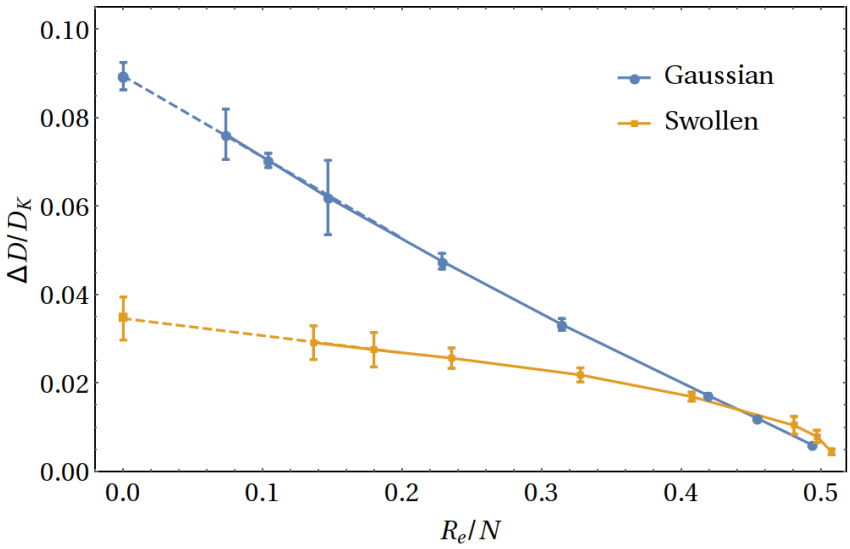

Figure 5.9: Long-time fractional decrease in diffusivity plotted against the expected form of the largest correction to scaling. The values of $\Delta D/D_K$ for long chains, where the data was too noisy to reach $t/\tau_z = 1$, used the master curves in Fig. 5.7 to extrapolate out to the Zimm time. On these data, the error bars show the difference between this master plot extrapolation and the estimated upper bound from a linear extrapolation. For the data that did reach $t/\tau_z = 1$, the error bars show twice the standard deviation of $\Delta D/D_K$ measured across all simulations contributing to each data point. The dashed lines to the left of the data indicate fits with the function $m(R_e/N) + c$, and the data at $R_e/N = 0$ show the extrapolated infinite chain value with error bars calculated as $2\sqrt{\sigma^2_{cc}}$, with σ^2 the covariance matrix for m and c.

how far negative it would go. However, it can be used in reverse to extrapolate the incomplete data sets out to $t/\tau_z = 1$. Figure 5.9 plots the values obtained this way against R_e/N, with R_e the mean distance between the first and last beads and an easily measurable stand in for R_H. R_e/N is expected to encompass the N-dependence of the leading-order correction term, coming from the contribution of local interactions insensitive to the size of the chain. The presence of D_{Rouse} in D_K is such a term, while C_{AA}^{short} has already been argued to contribute such a term too.

The data in Fig. 5.9 forms encouragingly straight lines, further suggesting it is the leading correction being varied, while the slight curvature in each data set comes from higher order terms. Confusingly, these lines have a negative gradient, implying a negative coefficient on the correction despite only a positive contribution being known (C_{AA}^{short} can only *add* to the value). From this one must conclude there are negative correction terms in C_{AA}^{long} that were not captured in the power law ansatz. Regardless, the linear fit of the $N \geqslant 20$ data (shorter chains are omitted to reduce the influence of higher order corrections) in Fig. 5.9 allows extrapolation

back to $R_e/N \to 0$, finding

$$\left.\frac{\Delta D}{D_K}\right|_{N\to\infty} = \begin{cases} 8.9 \pm 0.3\% & \text{Gaussian} \\ 3.5 \pm 0.5\% & \text{swollen} \end{cases} \tag{5.41}$$

with errors quoted to 2σ. These values are in line with Fixman's 8% for Gaussian chains [134] and Liu and Dünweg's 3.5% in their $N = 200$ swollen chains [78].

They can also be compared to the results of Sunthar and Prakash [138], who studied the importance of ΔD on the crossover behaviour of the polymer hydrodynamic radius, $R_H \propto D^{-1} = (D_K - \Delta D)^{-1}$, between θ and good solvents. This was studied via swelling ratios $\alpha_H = R_H/R_H^\theta = D^\theta/D$ and $\alpha_I = D_K^\theta/D_K$, which themselves have the ratio

$$\frac{\alpha_I}{\alpha_H} = \frac{1 - \Delta D/D_K}{1 - \Delta D^\theta/D_K^\theta}. \tag{5.42}$$

Feeding the swollen and Gaussian results in Eq. (5.41) into the numerator and denominator respectively gives $\alpha_I/\alpha_H \approx 1.06$ for the WMCD good solvent data. Since our swollen chains correspond to the good solvent limit, the appropriate comparision is with the high solvent quality values of α_H and α_I in Fig. 2 in Ref. [138], which give $\alpha_I/\alpha_H = 1.14$.

Both results agree on $\alpha_I > \alpha_H$, but the discrepancy in values cannot be accounted for by the error margins in Eq. (5.41) and Sunthar and Prakash's results. The more likely origin is in the details of the interaction potentials used, particularly for excluded volume interactions. The WCA potential used here is quite hard regardless of the values of the parameters used. Meanwhile the narrow Gaussian potential used by Sunthar and Prakash (and indeed the exponential potential used by Liu and Dünweg [78]) is comparatively soft and can be made a smooth function of the solvent quality [139]. Thus it would be important for further work to firmly establish the effect of the choice of potential, and to explore the crossover between θ and good solvent. Not only would this provide a more complete comparison to previous work, but it would also test whether $\Delta D/D_K$ is a decreasing function of solvent quality as the WMCD data suggests.

5.9 Summary

This chapter has showed how to isolate the causal displacements in the Langevin equation from the noise term in WMCD, where they are normally supplied together (smart WMCD) or the causal part is absent from explicit calculations altogether (basic WMCD). This allowed the algorithm to study the change of polymer diffusivity across the time interval measured over, using the time integral of the autocorrelation function of this causal part. In fact, using time reversal arguments the key result

allowing the noise contribution to be ignored, namely Eq. (5.12), is an application of a more general result for the overdamped Langevin equations.

This work looked more closely at the polymer's velocity autocorrelation than previous workers, identifying distinct contributions dominating over short and long times in both Gaussian and swollen chains. On short time scales, the decorrelation of force pairs was seen to dominate, with features dependent on the interaction potentials but ultimately dying off as bonds rotate. While the understanding of the decay on longer time scales remains incomplete, it has been described empirically with a power law that segues into an exponential decay at the Zimm time. Furthermore, the successful collapse of the power law tail onto a master curve implies the fractional reduction in diffusivity will be a universal property of long chains.

The power law itself was found to be close enough to t^{-1} for neither short nor long times to really dominate in ΔD, although the contribution from the force-driven C_{AA}^{short} is expected to become negligible as chains get taken ever larger, leaving behind a master curve. The chains simulated were too short to see this master curve directly, and across their range $\Delta D/D_K$ was seen to increase with chain length, in accordance with the findings of previous work. However they were sufficient to attempt an extrapolation procedure, suggesting $\Delta D/D_K$ will converge on 8.9% in Gaussian chains and the smaller value of 3.5% in swollen chains. This implies $\Delta D/D_K$ decreases with solvent quality, but to build a more complete picture, future work using a continuously varying EV potential would be needed.

6

Motion of passive particles in active systems

This final chapter takes the methodology used in the study of polymer diffusion and applies it to systems of mixed active and passive particles, where the motion conferred to the passive particles by the active ones leads to interesting diffusive phenomena [140, 141]. The length scales are now those of whole micro-organisms, where Brownian motion plays a much smaller role than it does in equilibrium polymer systems and is most often left out of simulation work. Nevertheless, it is still present and this work takes full advantage of WMCD to efficiently add thermal effects to the mix.

Moreover, this work puts the remaining developments in Chapter 4 to use, utilising all of polydispersity, particle rotations, and the quartic wavelet to ensure continuous rotation fields across the wavelet boundaries. While the methods are familiar the systems differ greatly from those in previous chapters.

6.1 Swimmer models

Real swimming micro-organisms (or micro-swimmers, or simply 'swimmers') have a wide range of shapes, sizes and swimming mechanisms. Commonly studied examples include the round, green alga *Chlamydomonas reinhardtii* which swims in a breaststroke-like way with its two flagella [142, 143], and the smaller, rod-like *Escherichia coli* that swims by coiling its flagella into a tail-like bundle [144]. In some instances entire colonies of cells can act as a single swimming object [145], while spermatozoa are a well-known example of swimming cells produced by macroscopic

organisms [146].

Each species has its own swimming technique, but because they are all at low Reynolds number with negligible inertia they are all subject to two major constraints. First is the scallop theorem [41], whereby the time reversal symmetry of the Stokes equation requires time-asymmetric swimming motion. Otherwise the net displacement over the swimming cycle will be zero.

Second is that with negligible inertia and (near-)instantaneous fluid response to motion, the net force and torque on the swimmer is zero at all times [13]. Ignoring external forces such as gravity, the motion derives from internal forces which by Newton's third law balance with the forces on the swimmer from the fluid, such that the net force applied on the fluid is also zero (and likewise for torque). Hydrodynamic flows mean the swimmer still moves, but these flows are now more localised[1], at slowest decaying as the flow generated by a dipole force ($\sim r^{-2}$) [13]. Despite the variation in technique, swimmers can generically be categorised as either 'pushers' or 'pullers'. Figure 6.1 shows a local flow field typical of pusher type swimmers, including sperm and *E. coli*, which pushes fluid in front and behind the cell body, and draws fluid in from the side. The flow fields for puller type swimmers do the reverse.

Swimmer models used in simulations are no less diverse. The simplest represent the swimmer as a particle with an associated flow field generated by a dipole force [147]. More detailed models use a finite set of forces distributed to impart non-zero particle motion while having zero net force and torque on the fluid [148, 149, 150], while the most sophisticated use boundary element methods to define and apply forces around the surface of the swimmer [151, 152]. Other approaches use time asymmetric motion of a small set of connected particles (often 2 [153, 154] or 3 [155, 156, 157]) to achieve swimmer motion.

Of these, the asymmetric chains and the finite sets of forces most readily lend themselves to WMCD simulations. The former could adapt the details for polymers by applying time varying forces on the particles. The primary development here would be the addition of active forces. The same development would be required of the second approach, which could also utilise the rotational side of WMCD since the reorientation of the single particle would equate to the reorientation of the whole swimmer. It is this second approach that is taken here.

Multi-force models can use any number of forces placed in any arrangement so long as they sum to zero with zero torque to mimic constraints in real swimmers. This work uses the simplest of such models, with just two forces placed as shown in Fig. 6.1 giving a representation of a pusher type swimmer. Formally the two forces

[1]This is very much analogous to the centre of mass velocity being non-zero in polymers despite having zero net force. As with the polymer case, thermal flows and any generated by external forces are still long-ranged.

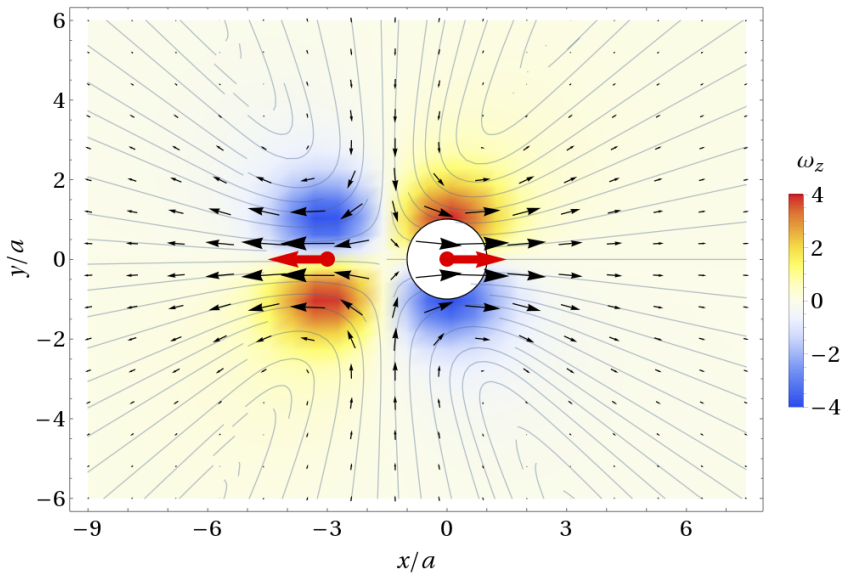

Figure 6.1: Schematic view of the 2-force model for a pusher type swimmer and its associated flow fields. The swimmer's body is indicated by the white disk, which moves in the positive x-direction as per the translational flow field (black arrows) at its centre. Red arrows indicate the position and direction of the swimming forces, while the coloured background shows the rotation field which is always into/out of the plane with negative/positive ω_z respectively. The full 3-dimensional flow field is symmetric under rotations about the x axis. Field strengths are in arbitrary units.

are time averaged representations of beating flagella and the swimmer body pushing backwards and forwards on the fluid respectively, so the simulations work on time scales longer than the microscopic beating of the flagella.

In WMCD the swimming forces bias any move they land inside in exactly the same way as the usual conservative forces do. The main change is the need to keep track of where the swimming forces are since they are no longer all placed at particle centres, and hence to know when they are inside a wavelet move. How this is achieved will depend on the specifics of the WMCD code and won't be discussed, but what will be generically true is that the swimmer model, centre, orientation and size should be enough information to locate all their swimming forces. Accordingly, if the host swimmer is displaced its swimming forces will all follow. The reverse is not necessarily true and this work treats swimming forces purely as move biases without displacing them directly.

By construction, biasing WMCD moves by the swimming forces generates the swimmer's flow field, including the swimmer's own motion equal to the net flow at its centre which is non-zero because the forces are placed asymmetrically. The details of the near-field flows will play a large part in the results to come, but it is recognised that the 2-force model is too simple to provide near-field flows that can be considered representative of real systems [143]. Future work can improve on this

but for the purposes of this thesis, which presents WMCD's first forays into active systems, this model suffices.

Similarly, excluded volume interactions (often called steric interactions in this context) are also omitted and the swimmers are treated as phantom particles. The reason to consider phantom swimmers is that steric interactions complicate the analysis without necessarily making the results more realistic unless the boundary conditions at the swimmer surface are carefully implemented. A discussion of what changes with the inclusion of steric interactions is at the end of the chapter. In the meantime there is an explicit acknowledgement that the results to come are limited to the model used.

What this work does capitalise on is the efficiency of WMCD in simulating many-body systems including the long-ranged correlated motion from thermal fluctuations. Where fluctuations have been included in this field, they have always been spatially white (i.e. neglect the hydrodynamic correlations) and usually retain only rotational diffusion [148, 158, 159]. More often they are neglected altogether. The justification is that, unlike thermally driven polymer systems, the Péclet number

$$Pe = \frac{a_s v_s}{D_{k_B T}} \gg 1 \tag{6.1}$$

with a_s and v_s the swimmer radius and speed respectively. Indeed feeding in typical values for *E. coli* finds $Pe = O(10^2)$ [144], and it would generally be even larger for eukaryotic swimmers. Even so, the effects of a finite $k_B T$ are not absent, which is most evident with rotational diffusion, where even small fluctuations lead to large deviations in swimmer paths as they swim, hence the inclusion of rotational noise in some work.

Active mechanisms for swimmer reorientation also exist, such as the run-and-tumble [160, 161] and reverse-and-flick [162, 163] techniques. These play a role in chemotaxis, allowing micro-organisms to better explore their environment and move up chemical gradients to more nutrient-rich areas. Such strategies are common in biological micro-swimmers, and theoretical and simulation work is starting to incorporate them as well [148, 164].

6.1.1 Run-and-tumble motion in WMCD

The implementation of run-and-tumble described herein separates swimming motion into alternating run phases, when swimmer motion follows the normal algorithm with swimming forces present, and finite time tumble phases during which the swimming forces are turned off. The length of the run step in real swimmers and other simulation work is Poisson distributed [148, 160], but for simplicity the length of

time in both run and tumble phases is constant in this work[2]. The phase of motion of each swimmer can then be determined with

$$\text{Swimming phase} = \begin{cases} \text{run} & \text{if } (t + \Delta t_i)\text{mod}(t_{\text{run}} + t_{\text{tumble}}) < t_{\text{run}} \\ \text{tumble} & \text{otherwise} \end{cases}, \qquad (6.2)$$

with Δt_i a particle-dependent offset used to desynchronise the swimmers. For the results in later sections, each of the N_s swimmers was given the offset $\Delta t_i = (i/N_s)(t_{\text{run}} + t_{\text{tumble}})$.

The aim is to enhance rotational diffusion during the tumble phase using the following procedure during each move:

> Check if any swimmers in the move are in their tumbling phase.
> If any, apply an additional random angular displacement to each of these.

There are two questions to address when determining the details of the additional angular displacement: how large does it need to be for a given rate of tumbling; and what hydrodynamic interactions should result from it? One route to answer the latter is to use the results of Section 4.3.4 to bias moves by a point torque at each tumbling swimmer, which then leads to hydrodynamic flows via $\boldsymbol{\mathcal{G}}^{TR}$ and $\boldsymbol{\mathcal{G}}^{RR}$. However, if the swimmer is to have zero net torque a more complex multi-torque bias would be required. Given the fluid flow would then decay faster than $\boldsymbol{\mathcal{G}}^{TR} \sim r^{-2}$ and be very short-ranged, one can justify cutting out the hydrodynamic interactions with these torques altogether.

Because WMCD moves are hydrodynamically coupled by construction, this requires the additional rotations to be supplied *outside* the usual moves. The task is made easy by having no energy associated with orientation (made possible by modelling swimmers as spheres), so there is no need for an acceptance test to accompany the tumble moves. An iteration on the vague procedure above would then be

> Perform and finish a WMCD move as normal.
> If it was accepted, check for any tumbling swimmers in the move.
> If any, apply uncorrelated random angular displacements to each.
> Then start the next normal WMCD move.

Only doing this after an *accepted* move simply ensures there is a well defined progression of time, and on a related note there is no additional time evolution associated with the extra rotations.

Now it is known *when* the tumbling is added, attention turns to *how*. The

[2]This is not because of any difficulty in using a Poisson distribution, but because it is easier to identify the properties of the model when fewer quantities vary.

task is to match the rotational diffusivity from small diffusive jumps in orientation,

$$D^{RR}_{\text{tumble}} = \frac{\langle \delta\theta^2 \rangle}{4\delta t}, \tag{6.3}$$

with the rotational diffusivity required for longer time decorrelation of swimmer orientation $\hat{\mathbf{v}}$ over the whole tumbling process [165]

$$\langle \hat{\mathbf{v}}(t_{\text{tumble}}) \cdot \hat{\mathbf{v}}(0) \rangle = \langle \cos\theta_{\text{tumble}} \rangle = e^{-2D^{RR}_{\text{tumble}} t_{\text{tumble}}}. \tag{6.4}$$

θ_{tumble} and t_{tumble} can be chosen as input parameters, e.g. to line up with real systems, so D^{RR}_{tumble} is also known and $\delta\theta$ is the main focus.

Over a small jump this can be written as

$$\delta\boldsymbol{\theta} = \hat{\mathbf{v}}(t+\delta t) - \hat{\mathbf{v}}(t) \approx \delta t\, \boldsymbol{\omega}_{\text{tumble}} \times \hat{\mathbf{v}}(t). \tag{6.5}$$

It will be helpful to separate the angular velocity into 3 factors as

$$\boldsymbol{\omega}_{\text{tumble}} = n^{-1/2}\, \omega_0\, \hat{\boldsymbol{\omega}}, \tag{6.6}$$

in which $\hat{\boldsymbol{\omega}}$ is a uniformly distributed random unit vector, ω_0 is a constant akin to A_0 in wavelet moves and $n^{-1/2}$ accounts for the number of particles inside the preceding WMCD move. Note that the use of n results from looking for tumbling swimmers in the WMCD moves, and hence picks up the same biases as the wavelet centre does in $\mathcal{P}_{\mathbf{b}|\lambda}$. Nothing else from the WMCD moves is used.

Keeping that in mind, $\langle n^{-1} \rangle$ in

$$\langle \delta\theta^2 \rangle = \delta t^2 \omega_0^2 \langle n^{-1} \rangle \langle (\hat{\boldsymbol{\omega}} \times \hat{\mathbf{v}})^2 \rangle \tag{6.7}$$

involves an integral over wavelet and Fourier moves to account for the likelihood of tumbling a given swimmer. Only the wavelet centre will actually matter so this boils down to

$$\langle n^{-1} \rangle = P_F N^{-1} + (1 - P_F) \int d^3\mathbf{b}\, \mathcal{P}_{\mathbf{b}|\lambda} n^{-1}\, \Theta(\lambda - |\mathbf{r} - \mathbf{b}|) = N^{-1} \tag{6.8}$$

where the Heaviside step function restricts the domain to those moves that actually contain the swimmer. Since $\hat{\boldsymbol{\omega}}$ and $\hat{\mathbf{v}}$ are uncorrelated, $\langle (\hat{\boldsymbol{\omega}} \times \hat{\mathbf{v}})^2 \rangle = 2/3$ and

$$\langle \delta\theta^2 \rangle = \frac{2\delta t^2 \omega_0^2}{3N}. \tag{6.9}$$

Substituting this and Eq. (6.4) into Eq. (6.3) then solving for ω_0 finds

$$\omega_0 = \left(\frac{3N \ln\left(\langle \cos\theta_{\text{tumble}} \rangle^{-1} \right)}{\delta t \, t_{\text{tumble}}} \right)^{1/2} \tag{6.10}$$

with which diffusive angular displacements in Eq. (6.5) will lead to the desired decorrelation over t_{tumble}.

Having worked through this, it is admitted that the factor of $n^{-1/2}$ was missed in the simulations in Section 6.3. The swimmers therefore tumble more than intended, but the analysis is not concerned with how much the swimmers tumble, only that they do it at all. Consequently the conclusions are unaffected.

6.1.2 The MC acceptance test in active systems

A central part of WMCD that has not yet been touched on in this chapter is the acceptance test. Given the systems of interest here are active, and by definition not in equilibrium, the principle of detailed balance does not apply and one could argue for disregarding the acceptance test altogether. That would then turn smart WMCD into a solver of the Langevin equation.

When looking at a system with passive components this is an imperfect solution, as is clear in the case of a vast system filled with passive particles and only a single swimmer. Certainly the system local to the swimmer will be out of equilibrium and all bets are off there, but far away the system will be in or close to equilibrium and will want an acceptance test to enforce this.

The pragmatic approach taken in this work is to keep the acceptance test but subtract the contribution from the active swimming forces in the move bias $\boldsymbol{\mathcal{F}}$ passed into Eq. (4.6). Any move that contains no active forces will be unaffected and will work towards a local equilibrium state as usual. The effect on moves that do contain active forces is less certain, but P_{acc} will decrease if the system is pushed against any conservative forces present; even if detailed balance makes no comment here it is clear this is a desirable situation that can help prevent the active forces driving the system into unphysical configurations.

Ultimately, the most important part will be the chosen value of A_0 rather than the presence of the acceptance test. This amplitude needs to be quite small anyway to realise smooth swimming and as with the passive polymers smart WMCD typically has $P_{\text{acc}} \gtrsim 98\%$, at which point the dynamics are almost exactly as prescribed by Langevin just with a bit of protection from energetically disallowed moves.

Table 6.1: Parameters used as standard in the simulations in this chapter. The parameters have been grouped into those that set code units, those pertaining to how swimmers move, and relevant system-wide parameters.

Parameter	Value in code	Physical value	Notes
a_s	0.5	1 μm	Swimmer radius.
k_BT	0.0045	26 meV	$T = 300$K.
τ	1	0.64 s	$= \pi\eta a_s^3/k_BT$.
			Uses η_{H_2O} =0.85 mPa s.
F_s	2.72	1.25 pN	Magnitude of swimmer forces.
v_s	12.8	40 μm s^{-1}	Swimmer speed.
Pe_s	154	=	Swimmer Péclet number.
t_{run}	1.56	1 s	
t_{tumble}	0.156	0.1 s	
L	20	40 μm	Periodic system.
A_0/k_BT	0.25	n/a	

6.1.3 System parameters

Multiple systems will be simulated in this chapter, all of which containing both active and passive particles. The (hydrodynamic) radius of the passive particles, a_p, will vary, but the properties of the active swimmers will be kept the same throughout. To give them some physical interpretation, the swimmers are set to mimic *E. coli* with relevant parameters set as per Table 6.1. Default system-wide parameters are also listed here for reference, and any deviations from these will be stated when relevant.

6.2 Low-noise trajectories

One of the best established aspects of swimmer systems is the way fluid elements - along with any tracer particles in the fluid - follow a loop trajectory as a swimmer passes by [147, 149, 152, 166, 167]. The idealised scenario with a single swimmer travelling in a perfectly straight line from $(-\infty, 0, 0)$ to $(\infty, 0, 0)$, passing a tracer initially at $(0, \rho, 0)$ on the way, is of course never observed in real, noisy systems with many swimmers each moving along their own curved paths. Even so, details of the ideal case still provide a good platform for understanding the basic interactions in more complex systems, as well as a cheap and clear test of the representation of swimmers in WMCD. To that end, this section shows and analyses trajectories of passive tracers in such idealised systems.

Since the underlying structure of WMCD is anything but noise-free, the temperature is set to be extremely small[3] so diffusion is dwarfed by causal displace-

[3]Working in the same units as in Table 6.1 the temparature used corresponds to about 0.3mK. This is much too cold for any biological experiments, which would always want to stay comfortably

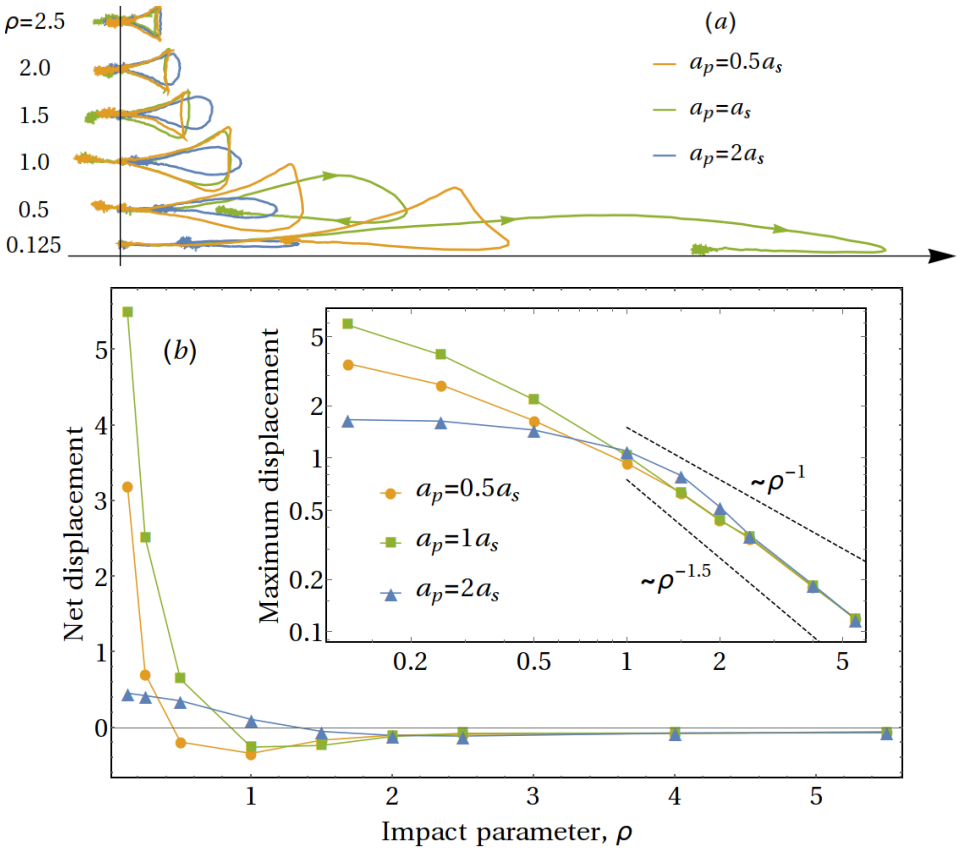

Figure 6.2: (a): Example trajectories of passive particles at different impact parameters with a swimmer travelling straight to the right. All trajectories are travelled in a clockwise sense, as indicated on the largest $a_p = a_s$ paths. (b): Plots of the net displacement over the whole trajectory, and the maximum displacement to the right (inset). All data were averaged over 100 trajectories to remove contributions thermal fluctuations.

ments, raising the swimmer's Péclet number to 1.3×10^8. This alone is not enough as even very slight rotations of the swimmer lead to significant deviations of the swimmer's path over the large distances required to construct full loops. To bypass this complication, rotational motion is turned off for data in this section only. Given the rare inclusion of rotations in other work this does not invalidate the results when comparing to most of the literature, and only goes to show how important thermal effects can be even at large Péclet numbers.

Figure 6.2(a) shows example trajectories of passive particles of different sizes when passed by a swimmer with impact parameter ρ. These trajectories highlight several properties with the aid of Fig. 6.2(b), which plots data averaged over hundreds of trajectories. First is the shape of the trajectory in the far-field where the swimmer's flow field approaches that of a perfect force dipole. The far-field is not

above the freezing point of water (273K), and therefore any experiments would expect to observe far noisier trajectories than the idealised ones in this section.

quite reached for the $a_p = 2a_s$ trajectories, but the smaller particles exhibit the characteristic cusped-triangle path in $\rho = 2, 2.5$ data [166] (as well as $\rho = 1.5$ for $a_p = 0.5a_s$). The distortions at smaller ρ are dependent on the swimmer model [152, 166] and will receive more attention below. In the meantime, the onset of the far-field is clearly defined in WMCD as the mobility tensor becomes exactly Oseen beyond $2\lambda_{\min}$, meaning the trajectories should limit to their far-field form in ascending order of a_p. This matches what is observed in the loops, and is also confirmed more quantitatively by the sequential collapse of data in the inset of Fig. 6.2(b).

The size of and net displacement in the loops also agrees with expectation. The far-field size, here measured by the maximum displacement in the direction of the swimmer's velocity, decays with a power law faster than ρ^{-1}. This fits with the recent study by Shum and Yeomans [152]. Meanwhile the net displacement is large and positive at small ρ, but is small and negative at greater separations [167].

The near-field properties of these trajectories are strongly dependent on a_p, and in preparation for the next section is worth discussing the key details. First and perhaps most strikingly, Fig. 6.2(b) shows the $a_p = a_s$ particles see the largest displacements at small ρ. This is in contrast to Shum and Yeomans' work where the displacement is monotonic with a_p [152]. The difference is attributable to a difference in swimmer model, especially regarding their swimmers having an excluded volume with no-slip boundaries, emphasising the need to be conscious of the properties of any given model.

To help build an understanding of the non-monotonicity in Fig. 6.2, Fig. 6.3 shows near-field flows around the swimmer as observed by particles of different radii, in the swimmer's reference frame. Viewing the flow fields this way is particularly useful for estimating the time over which the passive particle is carried along by the swimmer τ_p, which will correspond to net displacement via

$$\Delta r \approx v_p \tau_p. \tag{6.11}$$

The easiest case to analyse is $a_p = 2a_s$, where the relative flow field is everywhere backwards with little transverse component. The reasons for this are evident when recalling $\boldsymbol{\mathcal{G}}_{ii}^{TT} \propto a^{-1}$, so even when the swimmer and passive particles lie right on top of one another the smaller swimmer will respond to the swimming forces with a greater velocity. Combining the rate of displacement $v_p \sim a_p^{-1}$ with the time taken to pass through the swimmer, estimated to be

$$\tau_{\text{ad}} \sim \frac{a_s}{v_s - v_p} \sim \frac{a_s}{a_s^{-1} - a_p^{-1}}, \tag{6.12}$$

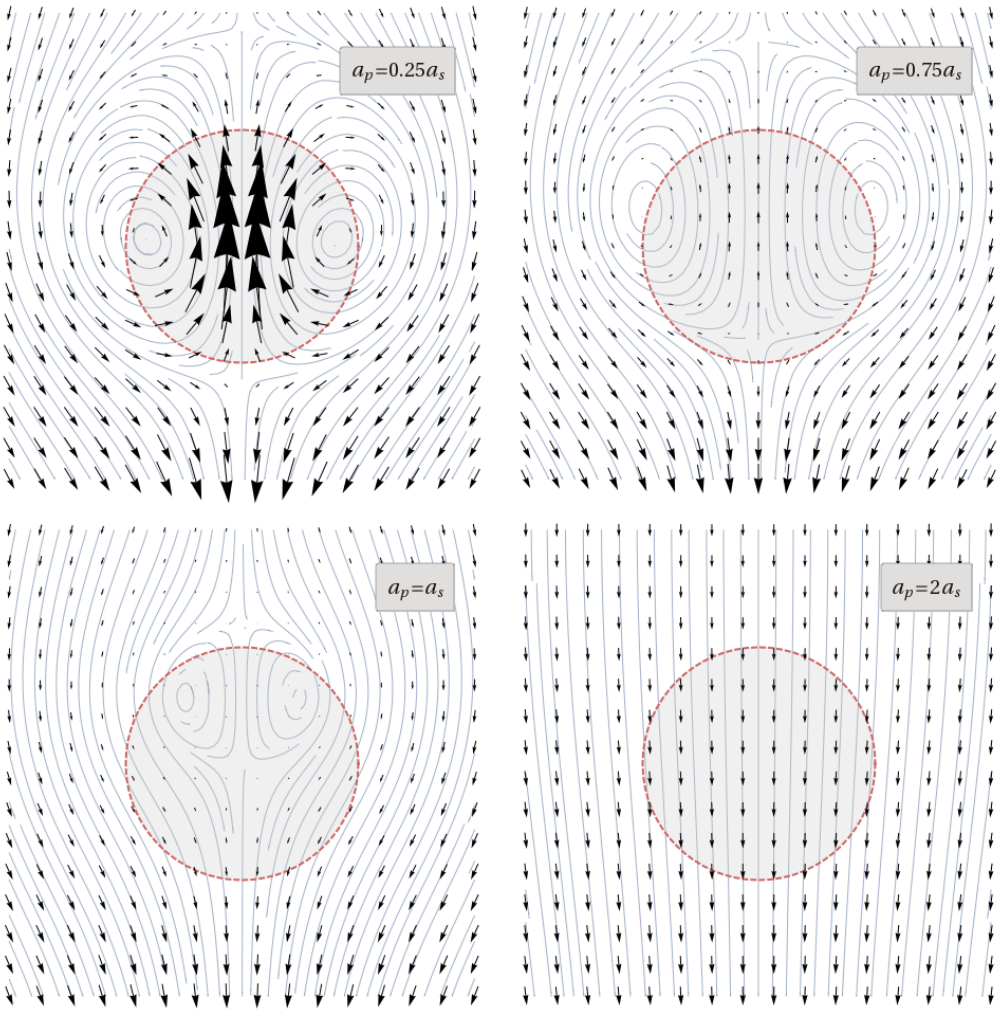

Figure 6.3: Translational flow fields relative to the swimmer velocity (upwards) as seen by particles of different sizes. Vector lengths across all plots have the same (arbitrary) units and can be compared directly. The disk, with radius a_s, indicates the swimmer size, and the flow field everywhere outside the plotted region is downwards.

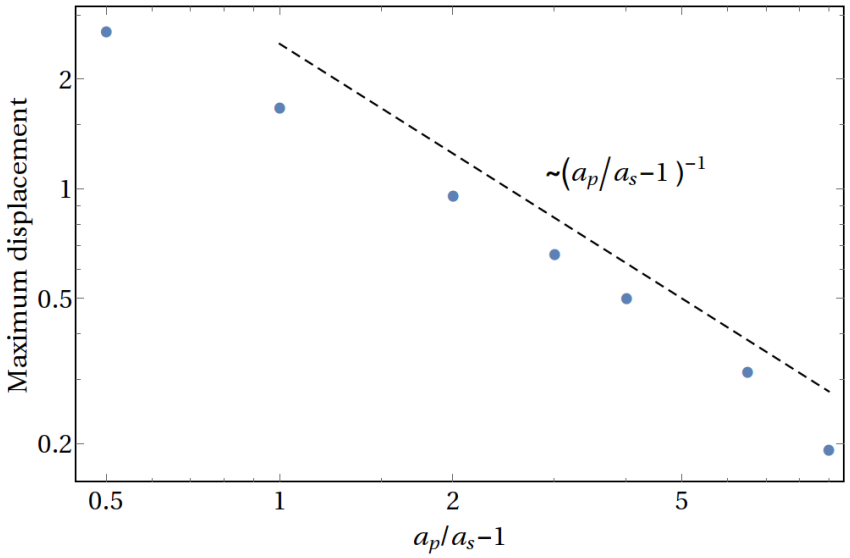

Figure 6.4: Plot of loop sizes for large passive particles when an idealised swimmer passes with impact parameter $\rho = 0.125$. The dashed line indicates the behaviour predicted in Eq. (6.13).

predicts the net displacement to vary with a_p as

$$\Delta r \approx v_p \tau_{\text{ad}} \sim a_s \left(\frac{a_p}{a_s} - 1 \right)^{-1} ; \qquad a_p > a_s. \tag{6.13}$$

This is confirmed in Fig. 6.13 for $a_p/a_s - 1 \gtrsim 2$. Below this the prediction inevitably breaks down as the loop size has already been seen to be finite at $a_p = a_s$, and Δr must be a smooth function of particle size.

A similar rough analysis for small particles is more difficult because of the more complex flow fields, while an accurate calculation would just be the long way to get to the simulation results. The important details can nevertheless be understood at a qualitative level. While the $a_p = 0.25a_s$ and $a_p = 0.75a_s$ flow fields in Fig. 6.3 are significantly more complex around the swimmer, the stream lines form closed loops meaning any particle that starts outside the swimmer will simply follow external stream lines and be carted around the outside of the swimmer. It is notable that the shapes of the stream lines do not change significantly as a_p is reduced between the two values shown, and nor will they as they decrease below $0.25a_s$. This can be understood in terms of λ_{min} and its effect on the wavelets present, but in essence only the flow very local to the swimming forces changes. As a consequence, the trajectories do not significantly change either leading to the $a_p = 0.5a_s$ data in Fig. 6.2 being representative of all smaller particle sizes.

The final question is then why the $a_p = a_s$ case has the largest loops. Looking at the corresponding flows in Fig. 6.3 one finds the stream lines to exhibit both

closed vortices and negative internal flows, bridging the two cases above. However, the most important part is how small the relative velocities are inside and in front of the swimmer. The passive particle therefore passes the swimmer very slowly, during which time it has been displaced a significantly larger distance than either the large or small particles.

6.3 Diffusivity of passive particles in active systems

Beyond the idealised and isolated interactions of the previous section, much of the work in active-passive suspensions has investigated the dependence of passive particle motion on the concentration of swimmers [141, 147, 148, 168, 169, 170, 171]. Drawing parallels to the treatment of polymer diffusion, the total long-time diffusion of passive particles can be split into contributions from thermal fluctuations and displacements caused by passing active particles:

$$D_{\text{tot}} = D_{k_BT} + D_A. \tag{6.14}$$

D_A can itself be split into contributions from hydrodynamic and steric effects [147].

Many-body interactions are rare at low swimmer concentrations, leading to D_A being proportional to the frequency of small impact parameter interactions, and hence proportional to the swimmer concentration [168]. At sufficiently high concentrations multi-swimmer interactions can enhance motion yet further [148, 172], but this will not feature here.

Recent experimental work by Patteson et al. extended investigations to the relative sizes of passive and active particles [141]. In suspensions of *E. coli* they varied the size of passive colloids both below and well above the size of the bacteria. In doing so they found that colloid diffusivity is non-monotonic with radius in systems with moderate swimmer concentrations, peaking at colloids of similar size to the bacteria.

They linked this to Kasyap's theoretical work in which run-and-tumble motion was included in the calculation of D_A [164]. Important for the consideration of particle size was the incorporation of tracer D_{k_BT}, and in turn the Péclet number, into their calculations. Specifically they found

$$D_A \sim \begin{cases} Pe^{1/2} & \text{for } Pe \ll 1 \\ \text{constant} & \text{for } Pe \gg 1 \end{cases} \tag{6.15}$$

with a peak at $Pe \approx 10$. This does not yet take full account of the particle's size however, as the Oseen tensor was used and therefore any important near-field hydrodynamic effects would have been missed.

It is here that the present work aims to take another step forwards, using WMCD's particle size-dependent mobility tensor to better (if still not perfectly) represent near-field flows and its inclusion of thermal fluctuations to explicitly include the effects of a finite Péclet number.

Despite the vast set of differences between the systems, this work is very similar to the study of polymer diffusion in the previous chapter at a mathematical level, with D_{k_BT} being a very simple version of the Kirkwood diffusivity and D_A being akin to ΔD. Not only is D_{k_BT} significantly simpler, being

$$D_{k_BT} = \frac{k_BT}{6\pi\eta a_p}\left(1 - 2.837\,\frac{a_p}{L}\right) \tag{6.16}$$

for a sphere in a periodic system as per Eq. (2.34), but the time scales are much shorter than the polymer Zimm time (at least once the time unit has been rescaled to the larger objects), *and* the signal is much larger than a few per cent. Consequently there is no difficulty in integrating over the full velocity autocorrelation directly to converge on D_A.

The one trick that is employed is to disregard the zero-time value of $C_{vv}(t) = \langle \mathbf{v}(t) \cdot \mathbf{v}(0) \rangle$ on the basis that this contains all of D_{k_BT}. The active contribution to $C_{vv}(0)$ is then obtained by extrapolating back from the small finite t data. Hence

$$D_A = \frac{1}{3}\int_{0^+}^{\infty} dt\, C_{vv}(t), \tag{6.17}$$

where the lower limit 0^+ denotes $t = 0$ as obtained by this extrapolation.

6.3.1 D_A with varying a_p and swimmer concentration

The best place to start with the simulation results is D_A itself for a range of particle sizes and swimmer concentrations, as shown in Fig. 6.5. Part (a) of this figure sees both D_A and D_{k_BT} dominating with different parameter choices and in general they are comparable in size. This is true even with parameters set to mimic room temperature bacterial systems as per Table 6.1, so one cannot always neglect thermal effects despite it being common practice to do so.

In Fig. 6.5(b) D_A is instead scaled by the swimmer concentration, which collapses the data and confirms the systems are dilute enough for multi-swimmer interactions to be negligible. In these data two distinct regimes are seen and D_A is clearly non-monotonic with a peak just below $a_p = a_s$. Hence the data qualitatively agrees with the experimental results of Patteson et al. [141].

So far, so good. The task now is to understand the underlying processes, for which it is fortunate that multi-swimmer interactions can be neglected and the

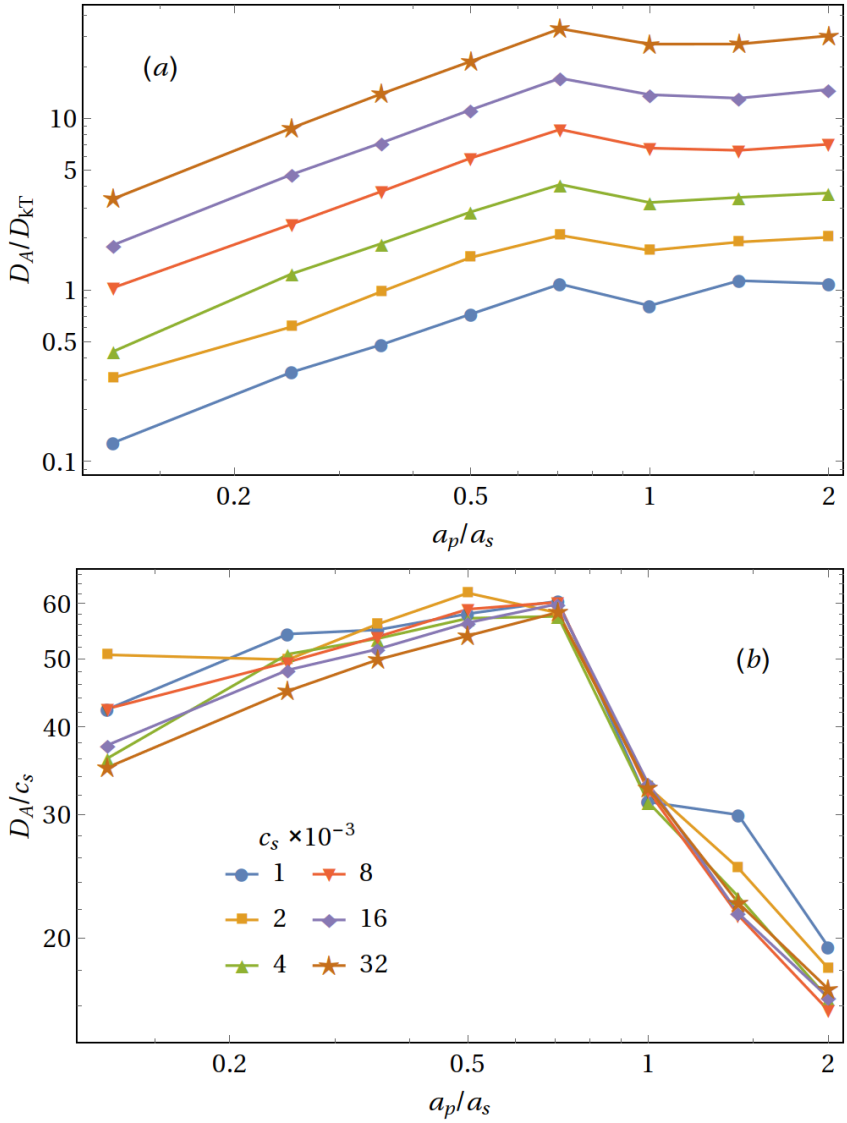

Figure 6.5: Active diffusivity measured in systems with different size passive particles across a range of swimmer concentrations. (a): Data scaled by thermally driven diffusivity to show their relative sizes. (b): The same data scaled by swimmer concentration $c_s = N_s/L^3$.

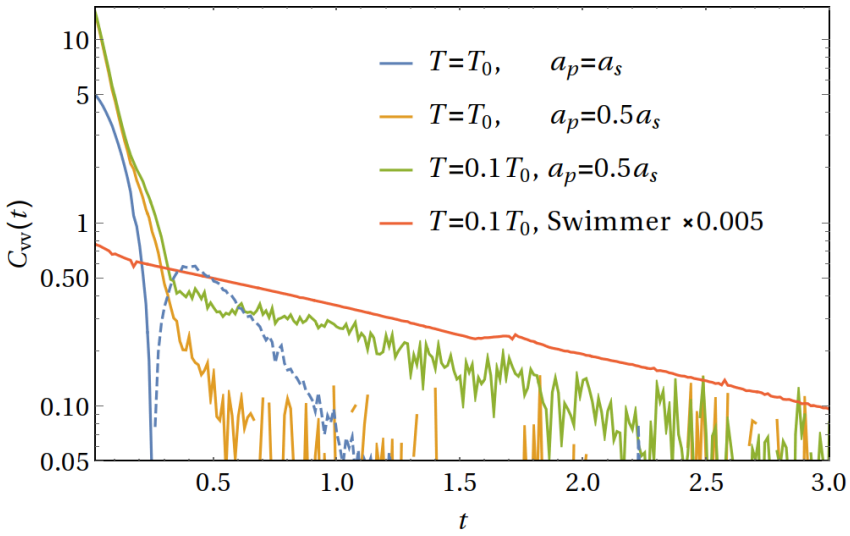

Figure 6.6: Example velocity autocorrelations exhibiting the three distinct forms observed across all data collected. The blue dashed line is the continuation of the solid blue line but with a sign change. All data are at swimmer concentration $c_s = 8 \times 10^{-3}$.

focus is solely on interactions between single swimmers and passive particles. Furthermore the a_p-independence of the far-field mobility tensors directs attention onto near-field interactions, such that the discussion of near-field loops in Section 6.2 is expected to play a role here as well. Since D_A will scale with the jump size in these interactions, the same ideas carry over and the approach of identifying the time scales of interactions is again adopted.

The main differences between the present discussion and that in Section 6.2 are the presence of thermal fluctuations and swimmer tumbling. With these in mind three mechanisms for breaking out of interactions are identified, each associated with a time scale.

Transits: The first mechanism has already been encountered, and is simply the following of the stream lines in Fig. 6.3. For this to dominate the thermal effects must be small, so this is expected to be important for large particles with $a_p > a_s$, where τ_{ad} in Eq. (6.12) applies. These interactions still lead to loop-like trajectories, which should manifest itself in the passive particle's velocity autocorrelation with a strong negative part characterising the forwards-backwards motion. This is observed in the blue curve in Fig. 6.6, which is representative of all $a_p \geqslant a_s$ data. This is as predicted, although the definition of τ_{ad} is too simplistic to capture the transit time for $a_p \leqslant a_s$.

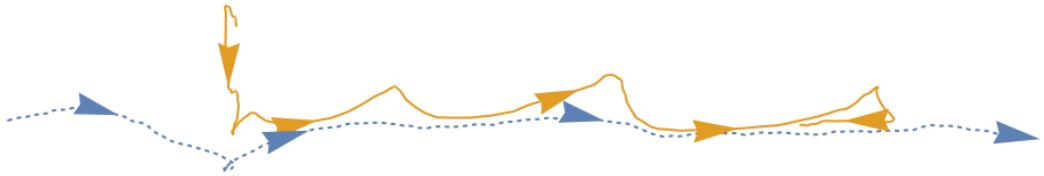

Figure 6.7: Example trajectory of a passive particle (solid, orange line) that became trapped in a closed stream line and transported by a swimmer (dashed, blue line) over an extended distance.

Incomplete loops: When thermal fluctuations are significant the idealised interactions of Section 6.2 get distorted and broken, both by the passive particles diffusing across stream lines and the reorientation of swimmers. This then leads to incomplete loops which have received some attention in previous work [147, 148].

Estimating the interaction time to be of order the time taken to diffuse the swimmer size (since this sets the length scale in the interaction) gives

$$\tau_{k_BT} = \frac{a_s^2}{6D_{k_BT}} \approx \frac{\pi\eta a_s^2 a_p}{k_BT}. \tag{6.18}$$

This will dominate over the transit mechanism if $\tau_{k_BT} < \tau_{\text{ad}}$, which would be expected with small a_p. Given the definition of τ_{ad} does not apply when the passive particles are smaller than the swimmers, it is most useful to look at the properties of C_{vv}, which for incomplete loops is expected to lose its negative part. This is observed in the orange data in Fig. 6.6, whose form is common in small particle data.

Swimmer rotations: The final time scale comes from the swimmers themselves. Because they reorient over time, both by tumbling and rotational diffusion, swimmers' trajectories are diffusive on long time scales. To see the importance of this to the passive particles, consider placing a particle in the closed stream lines in Fig. 6.3. As the swimmer swims, the passive particle will cycle around in this vortex until it crosses the stream lines and escapes the swimmer, all the while it will have adopted the swimmer's own motion. Of course if it is possible to diffuse out, it is also possible to diffuse in and become trapped in the closed loops, leading to a similar phenomenon to Taylor dispersion where diffusion across stream lines can lead to enhanced diffusion in shear flows [173, 174]. An example of this type of motion observed in simulations is shown Fig. 6.7.

While it is true that the escape mechanism is again diffusive, the time scale of the interaction is so drastically different to the incomplete loop case that it should

be treated separately. Thus the third time scale is that of swimmer rotations:

$$\tau_{\text{rot}} = \left(2D_s^{RR} + 2D_{\text{tumble}}^{RR} \frac{t_{\text{tumble}}}{t_{\text{run}} + t_{\text{tumble}}} \right)^{-1} \tag{6.19}$$

with $D_s^{RR} = k_B T/(8\pi\eta a_s^3)$ and the tumble time is spread across the full run-and-tumble cycle.

This type of motion is reliant on τ_{rot} dominating over τ_{k_BT}, otherwise particles escape the closed loops too quickly to inherit the swimmer's behaviour. Fortunately all the tumbling parameters are independent of particle sizes and temperature, so it is possible to change τ_{k_BT} while leaving τ_{rot} approximately fixed. Using large particles is little help because τ_{ad} then takes over, motivating investigations at different k_BT. The discussion will return to these shortly, but for now the green reduced temperature data in Fig. 6.6 shows exactly the sought behaviour, with a tail in C_{vv} matching the exponential decay of the swimmers' own data.

Having now discussed the three primary types of active-passive interactions and characterised them by the dominant time scales, attention now returns to understanding the non-monotonicity in Fig. 6.5(b). Using the behaviour of C_{vv} at each data point (not shown since they are just repeats of the curves in Fig. 6.6) finds all data with $a_p \leqslant a_s/\sqrt{2}$ to be dominated by τ_{k_BT} and all data for larger particles to be dominated by τ_{ad}. From this alone the non-monotonicity can be inferred on the basis that D_A will increase with the typical jump size in each interaction, which will itself increase with the dominant time scale. Since τ_{k_BT} and τ_{ad} are increasing and decreasing functions of a_p respectively, the existence of a maxima where they cross over is expected.

While this is no substitute for a full calculation and does not try to quantify the form of D_A, it is nonetheless valuable to identify the key mechanisms at a qualitative level as a basis for calculations in future work.

6.3.2 D_A with varying a_p and k_BT

With the absence of any τ_{rot}-dominated data in Fig. 6.5, this section repeats the simulations at different temperatures to shift the value of τ_{k_BT}. By reducing k_BT the regime with $\tau_{\text{rot}} < \tau_{k_BT}$ can be explored, while increasing k_BT tests whether it is really τ_{k_BT} that matters or another quantity that increases with a_p. The temperature used thus far is now set as the reference temperature T_0, and Fig. 6.8 shows data at temperatures raised and lowered by a factor of 10.

Part (a) of this figure has two particularly notable features. First is the collapse of data above $a_p/a_s = 1$, implying the dominant time scale for large particles is independent of temperature. This matches expectation as k_BT does not appear in

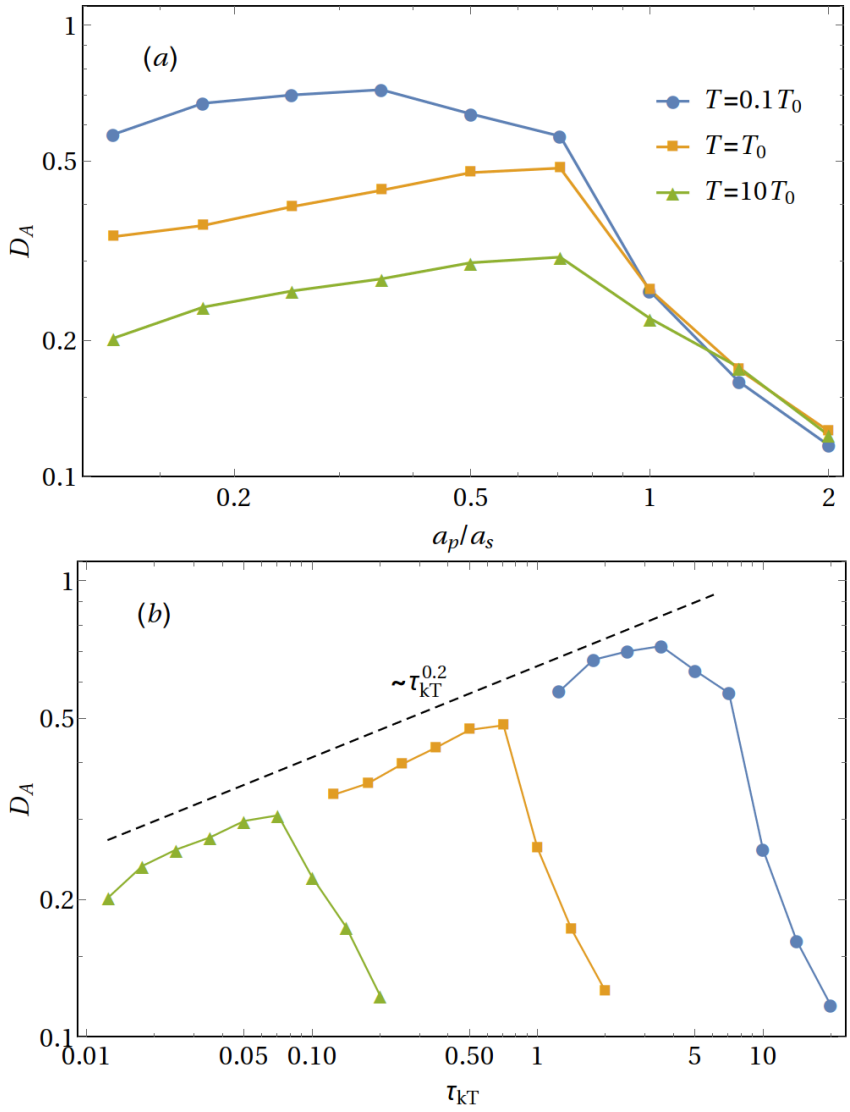

Figure 6.8: Active diffusivity in systems at different temperatures. The horizontal axes are scaled to reveal collapse in different regimes, with the dashed line in (b) giving only a rough indication of a possible underlying power law. All data are at swimmer concentration $c_s = 8 \times 10^{-3}$.

τ_{ad}, while the $T = 10T_0$ data point at $a_p/a_s = 1$ differing from the lower temperature data suggests thermal effects are starting to play a larger role here.

The second key feature is the difference in shape between the $T = 0.1T_0$ and higher T data at $a_p/a_s < 1$. Where the latter run parallel with a shallow positive gradient, the $T = 0.1T_0$ data is almost constant. It is this data that exhibits the long tail in Fig. 6.6. The reason for it being constant is simply that C_{vv}, and in turn D_A, is determined only by the swimmer properties, not the size of a_p. This was characterised by τ_{rot} being independent of a_p.

Figure 6.8(b) takes a different approach to plotting this data, aiming to see if D_A is indeed a function of τ_{k_BT} in the diffusion-dominated regime. Here the small-particle branches are seen to line up (allowing for the known differences in the $T = 0.1T_0$ data), effectively forming a master curve, and thus confirming expectation. The power law indicated in this figure is only a rough guide and not a fit to the data, but it does provide a comparison to Kasyap, Koch and Wu's result in Eq. (6.15) [164]. As $\tau_{k_BT} \propto Pe$ with fixed swimmer parameters, the data obtained in this work suggests a significantly smaller power of approximately 0.2 as opposed to 0.5. The reason for the difference is unknown, but differences in the details of short-range hydrodynamics (where Kasyap used the unregularised Oseen tensor) and the swimmer model could be the origin. It is also possible the Péclet numbers in Fig. 6.8(b) (154 times the value of τ_{k_BT}) are not yet small enough to see the $Pe^{1/2}$ behaviour, as Kasyap's Fig. 1 suggests a slower increase for $Pe \gtrsim 0.1$.

6.3.3 Comments on the effect of including excluded volume interactions

The final comments on this work regard the effect of excluded volume potentials on D_A, and why the focus has been on systems without them despite the previous chapter showing WMCD is quite capable of implementing them.

The obvious question to answer is what happens if the EV potentials used with good solvent polymers are carried over to the swimmers. The flow field generated by the swimmer is unchanged, but the part of it within $a_p + a_s$ of the swimmer's centre is no longer accessible. That cuts out the internal flows which is good, and instead the passive particles get pushed around the outside of the swimmer as is seen in all flows in Fig. 6.3. From here one soon realises that unless there is a perfect head-on collision the passive particles are quickly delivered to the back of the swimmer and left behind. This behaviour is akin to that of a squirmer-type swimmer [175, 176, 177].

That is all well and good if you want an efficient representation of a squirmer, and real world examples do exist [145], but most real swimmers have no-slip boundary conditions allowing particles at their surface to undergo significant entrainment

before being left behind [171]. Consequently the swimmers without EVs end up providing a more realistic version of events, despite the internal flows being unphysical.

When a_p is large enough the story changes as the particle centre is too distant to be quickly skirted around the swimmer. Here you find the WMCD swimmers slow down and shunt the passive particle along until the swimmer rotates to swim away, which is a more physical picture. However, no-slip boundary conditions would still be absent.

In all instances the boundary conditions are the main obstacle limiting the accuracy of WMCD in active systems. The end of Chapter 4 suggested a couple of routes which may be successful here if developed.

6.4 Summary

This chapter has seen WMCD applied to mixed active and passive systems, using the polydispersity and rotations sections in Chapter 4. As a proof of concept for active particles in WMCD, a simple 2-force swimmer model has been used and shown to successfully reproduce the cusped-loop trajectories of passive particles in the far-field of idealised systems. The sizes of loops at small impact parameters depend more strongly on the model, and exhibit non-monotonic dependence on a_p, understood in terms of the time taken for the particles to pass by each other and hence for the interaction to end.

This philosophy was carried over to explain a related non-monotonicity in the active diffusion of passive particles in systems with small but not insignificant thermal fluctuations. Three important time scales were identified and observed with the help of the velocity autocorrelation. The first time scale, τ_{ad}, was the same as that involved in the idealised interactions and is characteristic of large particles with negligible Brownian motion. Due to having nearly complete (if not closed) loops, these can be identified in C_{vv} by the strong negative tail.

The second time scale, τ_{k_BT}, is associated instead with small particles with significant Brownian motion breaking the loop trajectories by crossing stream lines. Consequently C_{vv} loses its negative tail.

The final time scale, τ_{rot}, is more subtle and relies on having enough Brownian motion to cross into closed stream lines but to diffuse slowly enough to be trapped and transported over large distances, such that these events dominate even if they are rare. The passive particle C_{vv} then mimics that of the swimmers, with a slow decay driven by rotational diffusion and tumbling.

The key idea is that the shortest time scale dominates interactions, and D_A is monotonic increasing with time scale. Only when plotting against a_p, with which τ_{k_BT} increases, τ_{rot} is constant and τ_{ad} decreases, does D_A pick up the non-

monotonicity. Carrying this over to explain the experimental results of Patteson et al. requires care because the swimmer model used had several features that differ from real swimmers. Most notably, phantom swimmers were used, which completely changes the picture for large a_p and at the very least requires a re-interpretation of τ_{rot}-dominated dynamics. Nevertheless, the approach taken offers insights into the physical origin of the experimentally observed behaviour.

The final finding is a simple one, but one that highlights the usefulness of WMCD: thermal fluctuations matter. Despite the large Péclet numbers involved, simple Brownian motion played an important role in D_A and even small amounts of rotational diffusion in swimmers leads to large differences in position, and this is not diminished by using more realistic shapes for swimmers than perfect spheres. Theoretical and simulation work is beginning to include these effects, but implementations have so far lacked the long-range hydrodynamic coupling known to be important by the polymer community. Once accounting for this, considerations return to the choice of algorithm for efficient simulation as described back in Chapter 1 and Section 2.5, and it is here the WMCD stands at the forefront of current simulation algorithms.

Bibliography

[1] O. T. Dyer and R. C. Ball. Wavelet Monte Carlo dynamics: A new algorithm for simulating the hydrodynamics of interacting Brownian particles. *The Journal of Chemical Physics*, 146(12):124111, 2017.

[2] O. T. Dyer and R. C. Ball. Time reversal of the overdamped Langevin equation and Fixman's law. `http://wrap.warwick.ac.uk/116261/`. Uploaded: 15/04/2019.

[3] P.-G. de Gennes. Soft matter (Nobel lecture). *Angewandte Chemie International Edition in English*, 31(7):842–845, 1992.

[4] T. A. Witten and P. A. Pincus. *Structured Fluids*. Oxford University Press, 2004.

[5] L. S. Hirst. *Fundamentals of Soft Matter Science*. CRC Press, 2013.

[6] G. G. Stokes. On the effect of internal friction of fluids on the motion of pendulums. *Transactions of the Cambridge Philosophical Society*, 9:8–106, 1851.

[7] S. Kim and S. J. Karrila. *Microhydrodynamics: Principles and Selected Applications*. Butterworth - Heinemann series in chemical engineering. Dover Publications, 2005.

[8] M. Doi and S.F. Edwards. *The Theory of Polymer Dynamics*. International series of monographs on physics. Clarendon Press, 1986.

[9] P.J. Flory. *Principles of Polymer Chemistry*. Baker lectures 1948. Cornell University Press, 1953.

[10] T. Sanchez, D. T. N. Chen, S. J. DeCamp, M. Heymann, and Z. Dogic. Spontaneous motion in hierarchically assembled active matter. *Nature*, 491:431, 2012.

[11] C. A. Whitfield, T. C. Adhyapak, A. Tiribocchi, G. P. Alexander, D. Marenduzzo, and S. Ramaswamy. Hydrodynamic instabilities in active cholesteric liquid crystals. *The European Physical Journal E*, 40(4):50, 2017.

[12] A. Doostmohammadi, J. Ignés-Mullol, J. M. Yeomans, and F. Sagués. Active nematics. *Nature Communications*, 9(1):3246, 2018.

[13] E. Lauga and T. R. Powers. The hydrodynamics of swimming microorganisms. *Reports on Progress in Physics*, 72(9):096601, 2009.

[14] D. Huber, A. Oskooei, X. Casadevall i Solvas, A. deMello, and G. V. Kaigala. Hydrodynamics in cell studies. *Chemical Reviews*, 118(4):2042–2079, 2018.

[15] D. Houtman, I. Pagonabarraga, C. P. Lowe, A. Esseling-Ozdoba, A. M. C. Emons, and E. Eiser. Hydrodynamic flow caused by active transport along cytoskeletal elements. *Europhysics Letters (EPL)*, 78(1):18001, 2007.

[16] A. Callan-Jones, M. Durand, and J.-B. Fournier. Hydrodynamics of bilayer membranes with diffusing transmembrane proteins. *Soft Matter*, 12:1791–1800, 2016.

[17] M. Vögele, J. Köfinger, and G. Hummer. Hydrodynamics of diffusion in lipid membrane simulations. *Phys. Rev. Lett.*, 120:268104, 2018.

[18] K. Kremer and G. S. Grest. Dynamics of entangled linear polymer melts: A molecular-dynamics simulation. *The Journal of Chemical Physics*, 92(8): 5057–5086, 1990.

[19] P. J. Hoogerbrugge and J. M. V. A. Koelman. Simulating microscopic hydrodynamic phenomena with Dissipative particle dynamics. *Europhysics Letters*, 19(3):155, 1992.

[20] P. Español and P. Warren. Statistical mechanics of Dissipative particle dynamics. *Europhysics Letters*, 30(4):191–196, 1995.

[21] P. Español. Hydrodynamics from Dissipative particle dynamics. *Phys. Rev. E*, 52:1734–1742, 1995.

[22] R. D. Groot and P. B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of Chemical Physics*, 107(11):4423–4435, 1997.

[23] P. Español. Fluid particle model. *Phys. Rev. E*, 57:2930–2948, 1998.

[24] A. Malevanets and R. Kapral. Mesoscopic model for solvent dynamics. *The Journal of Chemical Physics*, 110(17):8605–8613, 1999.

[25] T. Ihle and D. M. Kroll. Stochastic rotation dynamics. I. Formalism, Galilean invariance, and Green-Kubo relations. *Phys. Rev. E*, 67:066705, 2003.

[26] T. Ihle and D. M. Kroll. Stochastic rotation dynamics. II. Transport coefficients, numerics, and long-time tails. *Phys. Rev. E*, 67:066706, 2003.

[27] G. Gompper, T. Ihle, D. M. Kroll, and R. G. Winkler. *Multi-Particle Collision Dynamics: A Particle-Based Mesoscale Simulation Approach to the Hydrodynamics of Complex Fluids*, pages 1–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[28] A. J. C. Ladd. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *Journal of Fluid Mechanics*, 271:285–309, 7 1994.

[29] A. J. C. Ladd. Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 2. Numerical results. *Journal of Fluid Mechanics*, 271:311–339, 7 1994.

[30] D. A. Perumal and A. K. Dass. A review on the development of lattice Boltzmann computation of macro fluid flows and heat transfer. *Alexandria Engineering Journal*, 54(4):955 – 971, 2015.

[31] D. L. Ermak and J. A. McCammon. Brownian dynamics with hydrodynamic interactions. *The Journal of Chemical Physics*, 69(4):1352–1360, 1978.

[32] T. Geyer and U. Winter. An $O(N^2)$ approximation for hydrodynamic interactions in brownian dynamics simulations. *The Journal of Chemical Physics*, 130(11):114905, 2009.

[33] T. Ando, E. Chow, Y. Saad, and J. Skolnick. Krylov subspace methods for computing hydrodynamic interactions in Brownian dynamics simulations. *The Journal of Chemical Physics*, 137(6):064106, 2012.

[34] A. Jain, Sunthar P., B. Dünweg, and J. Ravi Prakash. Optimization of a Brownian-dynamics algorithm for semidilute polymer solutions. *Physical Review E*, 85(6):066703, 2012.

[35] A. Saadat and B. Khomami. Matrix-free Brownian dynamics simulation technique for semidilute polymeric solutions. *Phys. Rev. E*, 92:033307, 2015.

[36] L. Miao, C. D. Young, and C. E. Sing. An iterative method for hydrodynamic interactions in Brownian dynamics simulations of polymer dynamics. *The Journal of Chemical Physics*, 147(2):024904, 2017.

[37] M. Fixman. Construction of Langevin forces in the simulation of hydrodynamic interaction. *Macromolecules*, 19(4):1204–1207, 1986.

[38] T. T. Pham, U. D. Schiller, J. Ravi Prakash, and B. Dünweg. Implicit and explicit solvent models for the simulation of a single polymer chain in solution: Lattice Boltzmann versus Brownian dynamics. *The Journal of Chemical Physics*, 131(16):164114, 2009.

[39] E. Guyon, J.-P. Hulin, L. Petit, and C. D. Mitescu. *Physical Hydrodynamics*. Oxford University Press, 2 edition, 2015.

[40] J. P. Heller. An unmixing demonstration. *American Journal of Physics*, 28 (4):348–353, 1960.

[41] E. M. Purcell. Life at low Reynolds number. *American Journal of Physics*, 45(1):3–11, 1977.

[42] C. Pozrikidis. *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 1992.

[43] J. K. G. Dhont. *An Introduction to Dynamics of Colloids*. Studies in Interface Science. Elsevier, 1996.

[44] R. E. DeWames, W. F. Hall, and M. C. Shen. On the molecular theories of polymer solutions. *The Journal of Chemical Physics*, 46(7):2782–2794, 1967.

[45] R. Zwanzig, J. Kiefer, and G. H. Weiss. On the validity of the Kirkwood-Riseman theory. *Proceedings of the National Academy of Sciences*, 60(2): 381–386, 1968.

[46] J. Rotne and S. Prager. Variational treatment of hydrodynamic interaction in polymers. *The Journal of Chemical Physics*, 50(11):4831–4837, 1969.

[47] H. Yamakawa. Transport properties of polymer chains in dilute solution: Hydrodynamic interaction. *The Journal of Chemical Physics*, 53(1):436–443, 1970.

[48] R. Cortez. The method of regularized Stokeslets. *SIAM Journal on Scientific Computing*, 23(4):1204–1225, 2001.

[49] R. Cortez, L. Fauci, and A. Medovikov. The method of regularized Stokeslets in three dimensions: analysis, validation, and application to helical swimming. *Physics of Fluids*, 17(3):031504, 2005.

[50] H. Risken. *The Fokker-Planck Equation*. Springer series in synergetics. Springer-Verlag Berlin Heidelberg, 2 edition, 1996.

[51] D. S. Lemons and A. Gythiel. Paul Langevins 1908 paper On the theory of Brownian motion [Sur la théorie du mouvement Brownien, C. R. Acad. Sci. (Paris) 146, 530-533 (1908)]. *American Journal of Physics*, 65(11):1079–1081, 1997.

[52] N. G. van Kampen. Itô versus Stratonovich. *Journal of Statistical Physics*, 24 (1):175–187, 1981.

[53] E. Wajnryb, P. Szymczak, and B. Cichocki. Brownian dynamics: divergence of mobility tensor. *Physica A: Statistical Mechanics and its Applications*, 335 (3):339 – 358, 2004.

[54] R. Kubo. The fluctuation-dissipation theorem. *Reports on Progress in Physics*, 29(1):255–284, 1966.

[55] B. Noetinger. Fluctuating hydrodynamics and Brownian motion. *Physica A: Statistical Mechanics and its Applications*, 163(2):545 – 558, 1990.

[56] H. Brenner. Coupling between the translational and rotational Brownian motions of rigid particles of arbitrary shape: II. general theory. *Journal of Colloid and Interface Science*, 23(3):407 – 436, 1967.

[57] M. Reichert. *Hydrodynamic interactions in colloidal and biological systems.* PhD thesis, Universität Konstanz, 2006.

[58] P. Mazur and W. van Saarloos. Many-sphere hydrodynamic interactions and mobilities in a suspension. *Physica A: Statistical Mechanics and its Applications*, 115(1):21 – 57, 1982.

[59] J. Happel and H. Brenner. *Low Reynolds Number Hydrodynamics.* Noordhoff,Leyden, 1973.

[60] A. Einstein. *Investigations on the Theory of the Brownian Movement.* Dover Books on Physics Series. Dover Publications, 1956.

[61] M. S. Green. Markoff random processes and the statistical mechanics of timedependent phenomena. II. Irreversible processes in fluids. *The Journal of Chemical Physics*, 22(3):398–413, 1954.

[62] R. Kubo. Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems. *Journal of the Physical Society of Japan*, 12(6):570–586, 1957.

[63] J.-P. Hansen and I. R. McDonald. Chapter 7 - time-dependent correlation and response functions. In J.-P. Hansen and I. R. McDonald, editors, *Theory*

of Simple Liquids, pages 265 – 310. Academic Press, Oxford, fourth edition edition, 2013.

[64] A. Einstein. The elementary theory of the Brownian motion. *Zeitschrift für Elektrochemie*, 14:235–239, 1908.

[65] B. Dünweg and K. Kremer. Molecular dynamics simulation of a polymer chain in solution. *The Journal of Chemical Physics*, 99(9):6983–6997, 1993.

[66] P. E. Rouse. A theory of the linear viscoelastic properties of dilute solutions of coiling polymers. *The Journal of Chemical Physics*, 21(7):1272–1280, 1953.

[67] B. H. Zimm. Dynamics of polymer molecules in dilute solution: Viscoelasticity, flow birefringence and dielectric loss. *The Journal of Chemical Physics*, 24(2): 269–278, 1956.

[68] J. G. Kirkwood and J. Riseman. The intrinsic viscosities and diffusion constants of flexible macromolecules in solution. *The Journal of Chemical Physics*, 16(6):565–573, 1948.

[69] P. Ahlrichs, R. Everaers, and B. Dünweg. Screening of hydrodynamic interactions in semidilute polymer solutions: A computer simulation study. *Phys. Rev. E*, 64:040501, 2001.

[70] A. Jain, B. Dünweg, and J. R. Prakash. Dynamic crossover scaling in polymer solutions. *Phys. Rev. Lett.*, 109:088302, 2012.

[71] I. Nishio, S.-T. Sun, G. Swislow, and T. Tanaka. First observation of the coil-globule transition in a single polymer chain. *Nature*, 281:208–209, 1979.

[72] R. Wang and Z.-G. Wang. Theory of polymers in poor solvent: Phase equilibrium and nucleation behavior. *Macromolecules*, 45(15):6266–6271, 2012.

[73] G. S. Grest and K. Kremer. Molecular dynamics simulation for polymers in the presence of a heat bath. *Phys. Rev. A*, 33:3628–3631, 1986.

[74] J. D. Weeks, D. Chandler, and H. C. Andersen. Role of repulsive forces in determining the equilibrium structure of simple liquids. *The Journal of Chemical Physics*, 54(12):5237–5247, 1971.

[75] S. Havlin and D. Ben-Avraham. Fractal dimensionality of polymer chains. *Journal of Physics A: Mathematical and General*, 15(6):L311–L316, 1982.

[76] N. Clisby. Accurate estimate of the critical exponent ν for self-avoiding walks via a fast implementation of the pivot algorithm. *Phys. Rev. Lett.*, 104:055702, 2010.

[77] C. Stoltz, J. J. de Pablo, and M. D. Graham. Concentration dependence of shear and extensional rheology of polymer solutions: Brownian dynamics simulations. *Journal of Rheology*, 50(2):137–167, 2006.

[78] B. Liu and B. Dünweg. Translational diffusion of polymer chains with excluded volume and hydrodynamic interactions by Brownian dynamics simulation. *The Journal of Chemical Physics*, 118(17):8061–8072, 2003.

[79] R. R. Schmidt, J. G. H. Cifre, and J. G. de la Torre. Comparison of Brownian dynamics algorithms with hydrodynamic interaction. *The Journal of Chemical Physics*, 135(8):084116, 2011.

[80] A. Saadat and B. Khomami. Computationally efficient algorithms for incorporation of hydrodynamic and excluded volume interactions in Brownian dynamics simulations: A comparative study of the Krylov subspace and Chebyshev based techniques. *The Journal of Chemical Physics*, 140(18):184903, 2014.

[81] P. P. Ewald. Die berechnung optischer und elektrostatischer gitterpotentiale. *Annalen der Physik*, 369(3):253–287, 1921.

[82] C. W. J. Beenakker. Ewald sum of the Rotne-Prager tensor. *The Journal of Chemical Physics*, 85(3):1581–1582, 1986.

[83] A. J. Banchio and J. F. Brady. Accelerated Stokesian dynamics: Brownian motion. *The Journal of Chemical Physics*, 118(22):10323–10332, 2003.

[84] D. Fincham. Optimisation of the Ewald sum for large systems. *Molecular Simulation*, 13(1):1–9, 1994.

[85] J. P. Hernández-Ortiz, J. J. de Pablo, and M. D. Graham. $N \log N$ method for hydrodynamic interactions of confined polymer systems: Brownian dynamics. *The Journal of Chemical Physics*, 125(16):164906, 2006.

[86] G. Grimmett and D. Stirzaker. *Probability and random processes*. Oxford university press, 2001.

[87] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[88] R. J. Glauber. Timedependent statistics of the Ising model. *Journal of Mathematical Physics*, 4(2):294–307, 1963.

[89] S. Nedelcu and J.-U. Sommer. Single chain dynamics in polymer networks: A Monte Carlo study. *The Journal of Chemical Physics*, 130(20):204902, 2009.

[90] L. R. Dodd, T. D. Boone, and D. N. Theodorou. A concerted rotation algorithm for atomistic Monte Carlo simulation of polymer melts and glasses. *Molecular Physics*, 78(4):961–996, 1993.

[91] M. R. Betancourt. Efficient Monte Carlo trial moves for polypeptide simulations. *The Journal of Chemical Physics*, 123(17):174905, 2005.

[92] S. Zamuner, A. Rodriguez, F. Seno, and A. Trovato. An efficient algorithm to perform local concerted movements of a chain molecule. *PLoS ONE*, 10(3): 1–27, 2015.

[93] A. Uhlherr, V. G. Mavrantzas, M. Doxastakis, and D. N. Theodorou. Directed bridging methods for fast atomistic Monte Carlo simulations of bulk polymers. *Macromolecules*, 34(24):8554–8568, 2001.

[94] N. Ch. Karayiannis, A. E. Giannousaki, and V. G. Mavrantzas. An advanced Monte Carlo method for the equilibration of model long-chain branched polymers with a well-defined molecular architecture: Detailed atomistic simulation of an H-shaped polyethylene melt. *The Journal of Chemical Physics*, 118(6): 2451–2454, 2003.

[95] E. P. Bernard, W. Krauth, and D. B. Wilson. Event-chain Monte Carlo algorithms for hard-sphere systems. *Phys. Rev. E*, 80:056704, 2009.

[96] T. A. Kampmann, H.-H. Boltz, and J. Kierfeld. Monte Carlo simulation of dense polymer melts using event chain algorithms. *The Journal of Chemical Physics*, 143(4):044105, 2015.

[97] K. Kikuchi, M. Yoshida, T. Maekawa, and H. Watanabe. Metropolis Monte Carlo method as a numerical technique to solve the Fokker-Planck equation. *Chemical Physics Letters*, 185(3):335 – 338, 1991.

[98] D. M. Heyes and A. C. Branka. Monte Carlo as Brownian dynamics. *Molecular Physics*, 94(3):447–454, 1998.

[99] X. Z. Cheng, M. B. A. Jalil, H. K. Lee, and Y. Okabe. Mapping the Monte Carlo scheme to langevin dynamics: A Fokker-Planck approach. *Phys. Rev. Lett.*, 96:067208, 2006.

[100] S. Whitelam and P. L. Geissler. Avoiding unphysical kinetic traps in monte carlo simulations of strongly attractive particles. *The Journal of Chemical Physics*, 127(15):154101, 2007.

[101] E. Sanz and D. Marenduzzo. Dynamic Monte Carlo versus Brownian dynamics: A comparison for self-diffusion and crystallization in colloidal fluids. *The Journal of Chemical Physics*, 132(19):194102, 2010.

[102] A. Patti and A. Cuetos. Brownian dynamics and dynamic Monte Carlo simulations of isotropic and liquid crystal phases of anisotropic colloidal particles: A comparative study. *Phys. Rev. E*, 86:011403, 2012.

[103] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. *Journal of Computational Physics*, 17 (1):10–18, 1975.

[104] P. J. Rossky, J. D. Doll, and H. L. Friedman. Brownian dynamics as smart Monte Carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978.

[105] C. Heil and D. F. Walnut, editors. *Fundamental Papers in Wavelet Theory*. Princeton University Press, 2006.

[106] A. Grossmann and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736, 1984.

[107] A. R. Ochadlick Jr., H. N. Kritikos, and R. Giegengack. Variations in the period of the sunspot cycle. *Geophysical Research Letters*, 20(14):1471–1474, 1993.

[108] J. A. Montoya Zegarra, N. J. Leite, and R. da Silva Torres. Wavelet-based fingerprint image retrieval. *Journal of Computational and Applied Mathematics*, 227(2):294 – 307, 2009.

[109] S. Anfinogentov and V. M. Nakariakov. Motion magnification in coronal seismology. *Solar Physics*, 291(11):3251–3267, 2016.

[110] L. Hudgins and J. H. Kaspersen. *Wavelets and detection of coherent structures in fluid turbulence*, pages 201–226. Cambridge University Press, 1999.

[111] G. Novati, S. Verma, D. Alexeev, D. Rossinelli, W. M. van Rees, and P. Koumoutsakos. Synchronisation through learning for two self-propelled swimmers. *Bioinspiration & Biomimetics*, 12(3):036001, 2017.

[112] J. C. van den Berg, editor. *Wavelets in Physics*. Cambridge University Press, 2004.

[113] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.

[114] P.S. Addison. *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance, Second Edition*. CRC Press, 2017.

[115] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998.

[116] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[117] S. Bacchelli and S. Papi. Filtered wavelet thresholding methods. *Journal of Computational and Applied Mathematics*, 164-165:39 – 52, 2004.

[118] J. M. Gere. *Mechanics of Materials*. Brooks/Cole, 2001.

[119] L. Durlofsky, J. F. Brady, and G. Bossis. Dynamic simulation of hydrodynamically interacting particles. *Journal of Fluid Mechanics*, 180:21–49, 1987.

[120] J. G. Kirkwood. The general theory of irreversible processes in solutions of macromolecules. *Journal of Polymer Science*, 12(1):1–14, 1954.

[121] G. K. Batchelor. Sedimentation in a dilute polydisperse system of interacting spheres. Part 1. General theory. *Journal of Fluid Mechanics*, 119:379–408, 1982.

[122] D. J. Kraft, R. Wittkowski, B. ten Hagen, K. V. Edmond, D. J. Pine, and H. Löwen. Brownian motion and the hydrodynamic friction tensor for colloidal particles of complex shape. *Phys. Rev. E*, 88:050301, Nov 2013.

[123] R. S. Strichartz. *A Guide to Distribution Theory and Fourier Transforms*. World Scientific, 2003.

[124] J. R. Blake. A note on the image system for a stokeslet in a no-slip boundary. *Mathematical Proceedings of the Cambridge Philosophical Society*, 70:303–310, 1971.

[125] M. Revenga, I. Zúñiga, and P. Español. Boundary conditions in dissipative particle dynamics. *Computer Physics Communications*, 121-122:309 – 311, 1999.

[126] J. B. Avalos, J. M. Rubi, and D. Bedeaux. Dynamics of polymers in solution: the role of time-dependent hydrodynamic interactions. *Macromolecules*, 24 (22):5997–6005, 1991.

[127] V. Lisy, J. Tothova, and A. V. Zatovsky. The effects of hydrodynamic noise on the diffusion of polymers in dilute solutions. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(01):P01024, 2008.

[128] J. Farago, H. Meyer, and A. N. Semenov. Anomalous diffusion of a polymer chain in an unentangled melt. *Phys. Rev. Lett.*, 107:178301, 2011.

[129] C.-C. Huang, G. Gompper, and R. G. Winkler. Effect of hydrodynamic correlations on the dynamics of polymers in dilute solution. *The Journal of Chemical Physics*, 138(14):144902, 2013.

[130] B. J. Alder and T. E. Wainwright. Decay of the velocity autocorrelation function. *Phys. Rev. A*, 1:18–21, 1970.

[131] M. Fixman. Inclusion of hydrodynamic interaction in polymer dynamical simulations. *Macromolecules*, 14(6):1710–1717, 1981.

[132] M. Selby. An exact calculation of the ratio of the short and long time diffusion coefficients for long polymer chains. Undergraduate thesis, University of Warwick (unpublished), 2018.

[133] M. Fixman. Effects of fluctuating hydrodynamic interaction. *The Journal of Chemical Physics*, 78(3):1594–1599, 1983.

[134] M. Fixman. Translational diffusion of chain polymers. I. Improved variational bounds. *The Journal of Chemical Physics*, 84(7):4080–4084, 1986.

[135] A. Jain and K. D. Dorfman. Evaluation of the Kirkwood approximation for the diffusivity of channel-confined DNA chains in the de Gennes regime. *Biomicrofluidics*, 9(2):024112, 2015.

[136] J. F. James. *A Student's Guide to Fourier Transforms: With Applications in Physics and Engineering*. Student's Guides. Cambridge University Press, 3 edition, 2011.

[137] B. Dünweg, D. Reith, M. Steinhauser, and K. Kremer. Corrections to scaling in the hydrodynamic properties of dilute polymer solutions. *The Journal of Chemical Physics*, 117(2):914–924, 2002.

[138] P. Sunthar and J. Ravi Prakash. Dynamic scaling in dilute polymer solutions: The importance of dynamic correlations. *Europhysics Letters*, 75(1):77–83, 2006.

[139] J. Ravi Prakash and H. C. Öttinger. Viscometric functions for a dilute solution of polymers in a good solvent. *Macromolecules*, 32(6):2028–2043, 1999.

[140] X.-L. Wu and A. Libchaber. Particle diffusion in a quasi-two-dimensional bacterial bath. *Phys. Rev. Lett.*, 84:3017–3020, 2000.

[141] A. E. Patteson, A. Gopinath, P. K. Purohit, and P. E. Arratia. Particle diffusion in active fluids is non-monotonic in size. *Soft Matter*, 12:2365–2372, 2016.

[142] R. Jeanneret, M. Contino, and M. Polin. A brief introduction to the model microswimmer Chlamydomonas reinhardtii. *The European Physical Journal Special Topics*, 225(11):2141–2156, 2016.

[143] K. Drescher, R. E. Goldstein, N. Michel, M. Polin, and I. Tuval. Direct measurement of the flow field around swimming microorganisms. *Phys. Rev. Lett.*, 105:168101, 2010.

[144] K. Drescher, J. Dunkel, L. H. Cisneros, S. Ganguly, and R. E. Goldstein. Fluid dynamics and noise in bacterial cell-cell and cell-surface scattering. *Proceedings of the National Academy of Sciences*, 108(27):10940–10945, 2011.

[145] T. J. Pedley, D. R. Brumley, and R. E. Goldstein. Squirmers with swirl: a model for Volvox swimming. *Journal of Fluid Mechanics*, 798:165–186, 2016.

[146] K. Ishimoto, H. Gadêlha, E. A. Gaffney, D. J. Smith, and J. Kirkman-Brown. Coarse-graining the fluid flow around a human sperm. *Phys. Rev. Lett.*, 118: 124501, 2017.

[147] D. O. Pushkin and J. M. Yeomans. Fluid mixing by curved trajectories of microswimmers. *Phys. Rev. Lett.*, 111:188101, 2013.

[148] D. Krishnamurthy and G. Subramanian. Collective motion in a suspension of micro-swimmers that run-and-tumble and rotary diffuse. *Journal of Fluid Mechanics*, 781:422–466, 2015.

[149] J. de Graaf and J. Stenhammar. Lattice-Boltzmann simulations of microswimmer-tracer interactions. *Phys. Rev. E*, 95:023302, 2017.

[150] T. C. Adhyapak and S. Jabbari-Farouji. Flow properties and hydrodynamic interactions of rigid spherical microswimmers. *Phys. Rev. E*, 96:052608, 2017.

[151] H. Shum, E. A. Gaffney, and D. J. Smith. Modelling bacterial behaviour close to a no-slip plane boundary: the influence of bacterial geometry. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466 (2118):1725–1748, 2010.

[152] H. Shum and J. M. Yeomans. Entrainment and scattering in microswimmer-colloid interactions. *Phys. Rev. Fluids*, 2:113101, 2017.

[153] P. T. Underhill, J. P. Hernandez-Ortiz, and M. D. Graham. Diffusion and spatial correlations in suspensions of swimming particles. *Phys. Rev. Lett.*, 100:248101, 2008.

[154] V. B. Putz and J. Dunkel. Low Reynolds number hydrodynamics of asymmetric, oscillating dumbbell pairs. *The European Physical Journal Special Topics*, 187(1):135–144, 2010.

[155] A. Najafi and R. Golestanian. Simple swimmer at low Reynolds number: Three linked spheres. *Phys. Rev. E*, 69:062901, 2004.

[156] G. P. Alexander, C. M. Pooley, and J. M. Yeomans. Hydrodynamics of linked sphere model swimmers. *Journal of Physics: Condensed Matter*, 21 (20):204108, 2009.

[157] M. S. Rizvi, A. Farutin, and C. Misbah. Three-bead steering microswimmers. *Phys. Rev. E*, 97:023102, 2018.

[158] R. Golestanian. Anomalous diffusion of symmetric and asymmetric active colloids. *Phys. Rev. Lett.*, 102:188305, 2009.

[159] M. E. Cates and J. Tailleur. When are active brownian particles and run-and-tumble particles equivalent? consequences for motility-induced phase separation. *Europhysics Letters*, 101(2):20010, 2013.

[160] H.C. Berg. *Random Walks in Biology*. Princeton paperbacks. Princeton University Press, 1993.

[161] M. Polin, I. Tuval, K. Drescher, J. P. Gollub, and R. E. Goldstein. Chlamydomonas swims with two 'gears' in a eukaryotic version of run-and-tumble locomotion. *Science*, 325(5939):487–490, 2009.

[162] L. Xie, T. Altindal, S. Chattopadhyay, and X.-L. Wu. Bacterial flagellum as a propeller and as a rudder for efficient chemotaxis. *Proceedings of the National Academy of Sciences*, 108(6):2246–2251, 2011.

[163] R. Stocker. Reverse and flick: Hybrid locomotion in bacteria. *Proceedings of the National Academy of Sciences of the United States of America*, 108: 2635–2636, 2011.

[164] T. V. Kasyap, D. L. Koch, and M. Wu. Hydrodynamic tracer diffusion in suspensions of swimming bacteria. *Physics of Fluids*, 26(8):081901, 2014.

[165] J. Saragosti, P. Silberzan, and A. Buguin. Modeling E. coli tumbles by rotational diffusion. Implications for chemotaxis. *PLOS ONE*, 7(4):1–6, 2012.

[166] J. Dunkel, V. B. Putz, I. M. Zaid, and J. M. Yeomans. Swimmer-tracer scattering at low Reynolds number. *Soft Matter*, 6:4268–4276, 2010.

[167] D. O. Pushkin, H. Shum, and J. M. Yeomans. Fluid transport by individual microswimmers. *Journal of Fluid Mechanics*, 726:5–25, 2013.

[168] G. Miño, T. E. Mallouk, T. Darnige, M. Hoyos, J. Dauchet, J. Dunstan, R. Soto, Y. Wang, A. Rousselet, and E. Clement. Enhanced diffusion due to active swimmers at a solid surface. *Phys. Rev. Lett.*, 106:048102, 2011.

[169] H. Kurtuldu, J. S. Guasto, K. A. Johnson, and J. P. Gollub. Enhancement of biomixing by swimming algal cells in two-dimensional films. *Proceedings of the National Academy of Sciences*, 108(26):10391–10395, 2011.

[170] J. J. Molina and R. Yamamoto. Diffusion of colloidal particles in swimming suspensions. *Molecular Physics*, 112(9-10):1389–1397, 2014.

[171] R. Jeanneret, D. O. Pushkin, V. Kantsler, and M. Polin. Entrainment dominates the interaction of microalgae with micron-sized objects. *Nature Communications*, 7:12518, 2016.

[172] G. V. Soni, B. M. Jaffar Ali, Y. Hatwalne, and G. V. Shivashankar. Single particle tracking of correlated bacterial dynamics. *Biophys. J.*, 84:2634–2637, 2003.

[173] G. I. Taylor. Dispersion of soluble matter in solvent flowing slowly through a tube. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 219(1137):186–203, 1953.

[174] A. J. T. M. Mathijssen, R. Jeanneret, and M. Polin. Universal entrainment mechanism controls contact times with motile cells. *Phys. Rev. Fluids*, 3: 033103, 2018.

[175] M. J. Lighthill. On the squirming motion of nearly spherical deformable bodies through liquids at very small Reynolds numbers. *Communications on Pure and Applied Mathematics*, 5(2):109–118, 1952.

[176] Z. Lin, J.-L. Thiffeault, and S. Childress. Stirring by squirmers. *Journal of Fluid Mechanics*, 669:167–177, 2011.

[177] J.-L. Thiffeault. Distribution of particle displacements due to swimming microorganisms. *Phys. Rev. E*, 92:023023, 2015.