Unravelling the Dynamics of Semidilute Polymer Solutions Using Brownian Dynamics

A THESIS
SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

by

Aashish Jain

Department of Chemical Engineering
Monash University

Clayton, Australia

June 28, 2013

Aashish Jain

In loving memory of my father

and

to my wife, Pallavi

for your love and support,
and your efforts in keeping my spirits up

Acknowledgments

I would like to express my sincere gratitude to Prof. Ravi Prakash Jagadeeshan, my PhD supervisor, for his constant support, encouragement and advice throughout my thesis work. Working with him, helped me to get an insight into the areas of "Polymer physics" and "Brownian dynamics simulations". I thank him further for his guidance in writing this thesis. The most effective learning period of my life became successful because of him, which will stay in my memories forever.

I am grateful to Prof. Burkhard Dünweg who has been an incredible mentor for many parts of my research. His excellent hold on theoretical physics helped my understanding whenever he visited Monash university. It was my pleasure to attend two very fascinating lecture series given by him at Monash. I am indebted to Dr. Prabhakar Ranganathan and Dr. P. Sunthar who provided their insightful comments and valuable suggestions throughout my PhD. I thank Prof. Billy Todd and Mr. Remco Hartkamp for helping me in understanding the Mixed Flow algorithm.

I would also like to acknowledge my colleagues and friends without whom this journey would not have been as fruitful and enjoyable as it has been. A big thanks to Debadi, Sarah, Sharad, Amo, Andi, Chamath, Emma, Anuja, Chandi and Gopesh for the several complicated but fascinating technical discussions, and for all the fun that I had in Molecular Rheology Group meetings. I shall always cherish the sweet memories attached with Debadi, Parama, Chiranjib and Mandar for their perpetual and unending support and making my journey at Monash a memorable one.

I thank the staff at the Department of Chemical Engineering, in particular Lilyanne Price and Jill Crisfield, for their help on administrative matters. My work would not have been possible without the MGS and IPRS scholarships from the MIGR, and the Finkel top-up scholarship. Travel grants provided by the MIGR, Engineering faculty and the department of Chemical Engineering also enabled me to present at conferences in Adelaide and Lisbon (Portugal).

I take this opportunity to thank the National Computational Infrastructure, Victorian Life Sciences Computation Initiative and Victorian Partnership for Advanced Computing for allocations of CPU-time on their supercomputers. I also thank Philip, Simon and Shahaan from Monash SunGrid facility for their help and effort in setting-up the computer programs.

Finally, I thank my parents, my brother and my wife for their unconditional support and encouragement. Without their constant support, it would not have been possible for me to embark upon this amazing journey.

Abstract

A polymer solution has three concentration regimes: (i) dilute (ii) semidilute and (iii) concentrated. There are a number of contexts involving polymer solutions, such as in the spinning of nanofibers or in ink jet printing, where in order to achieve the most optimal outcome the concentration of polymers must be in the semidilute regime. In many biological contexts as well, such as the diffusion of protein and other biomolecules, the essential physics occur in the semidilute regime. Therefore, it is extremely important to understand the behavior of semidilute polymer solutions from the fundamental and also from the technological point of view. A significant amount of research has been carried out in the dilute and concentrated regimes in the past by means of experiments, theories and computer simulations. These two regimes have been explored successfully because the behavior of polymer solutions in the dilute and concentrated regimes can be understood by studying the behavior of single molecules. In the dilute case the motivation for this is obvious, while in the concentrated case, by treating all the molecules that surround a particular molecule as obstacles that constrain its motion, the entire problem is reduced to understanding the motion of a polymer in a tube. This approximation, however, is not valid in the *semidilute regime*, which lies between the dilute and concentrated regimes, because of all the many-body interactions, that arise in this regime.

The main focus of this thesis is to develop an optimized Brownian dynamics (BD) simulation algorithm for semidilute polymer solutions at and far from equilibrium, that is capable of accounting for the many-body interactions. The goal is to use this algorithm to predict various physical properties for a range of concentrations and temperatures and to interpret the results in terms of the blob scaling theory.

The development of a BD simulation algorithm for multi-chain systems requires the consideration of a large system of polymer chains coupled to one another through excluded volume interactions (which are short-range in space) and hydrodynamic interactions (which are long-range in space). In the presence of periodic boundary conditions, long-ranged hydrodynamic interactions are frequently summed with the Ewald summation technique (Beenakker, 1986; Stoltz et al., 2006). By performing detailed simulations that shed light on the influence of several tuning parameters involved both in the Ewald summation method, and in the efficient treatment of Brownian forces, we describe the development of a BD algorithm in this thesis, in which the computational cost scales as $O(N^{1.8})$, where N is the number of monomers in the simulation box. It is also shown that Beenakker's original implementation of the Ewald sum, which is only valid for systems without bead overlap, can be modified so that θ solutions can be simulated by switching off excluded volume interactions. Comparison of the predictions by the BD algorithm of the gyration radius, the end-to-end vector, and the self-diffusion coefficient with the hybrid lattice Boltzmann-Molecular dynamics (LB-MD) method (Ahlrichs and Dünweg, 1999) shows excellent agreement between the two methods. This study has been published in the paper Jain et al. (2012).

The behavior of semidilute polymer solutions at equilibrium varies significantly with concentration and solvent quality. These effects are reflected in the concentration driven crossover from the dilute to the concentrated regime, and in the solvent quality driven crossover from theta solvents to good solvents in the phase diagram of polymer solutions. This double crossover region for concentration above the overlap concentration, is explored by Brownian dynamics simula-

tions to map out the universal crossover scaling functions for the gyration radius and the single-chain diffusion constant. Scaling considerations (Rubinstein and Colby, 2003), our simulation results, and recently reported experimental data (Pan, Nguyen, Sunthar, Sridhar & Prakash, Pan et al.) on the polymer contribution to the zero-shear rate viscosity obtained from rheological measurements on DNA systems support the assumption that there are simple relations between these functions, such that they can be inferred from one another. This study has been published in the paper Jain et al. (2012).

Unlike the simulation of equilibrium systems where periodic boundary conditions (PBCs) are used in an orthogonal cell to get rid of wall effects, for the simulation of far from equilibrium systems, appropriate PBCs need to be used such that they are compatible with any particular imposed flow. One should also be able to carry out the simulation for an arbitrary amount of time. Commonly, the Lees Edwards PBC (Lees and Edwards, 1972) is used for planar shear flow and the Kraynik-Reinelt PBC (Kraynik and Reinelt, 1992) is used for planar elongational flow. These PBCs have been used and tested in molecular dynamics simulations (Bhupathiraju et al., 1996; Todd and Daivis, 1998) and multi-chain BD simulations (Stoltz et al., 2006). In this thesis PBCs that can handle a *planar mixed flow* (which is a linear combination of planar elongational flow and planar shear flow) (Hunt et al., 2010) is implemented in a multi-chain BD simulation algorithm for semidilute polymer solutions. Preliminary results on the validation of the planar mixed flow algorithm are presented.

Contents

List of Figures

List of Tables

List of Notations

α	Ewald parameter 71
α_g	Swelling of a polymer molecule 18
$\langle R_e^2 \rangle$	Mean-square end-to-end distance of a polymer chain 55
$\left\langle R_g^2 \right\rangle$	Mean-square gyration radius of a polymer chain 56
$\boldsymbol{\mathcal{B}}$	$3N \times 3N$ matrix to denote the square root of $\boldsymbol{\mathcal{D}}$ 51
$\boldsymbol{\mathcal{D}}$	$3N \times 3N$ diffusion matrix 51
$\boldsymbol{\mathcal{L}}$	Cell basis matrix 153
β	The angle between the elongation and contraction axes in PMF 147
$\boldsymbol{\sigma}$	Stress tensor 60
$\boldsymbol{\delta}$	Unit tensor 53
$\dot{\boldsymbol{\Gamma}}$	Rate of strain tensor 60
$\boldsymbol{\kappa}$	Transpose of a velocity gradient tensor 51
$\boldsymbol{\Lambda}$	Time evolution matrix for lattice vectors 142

$\mathbf{M}^{(1)}$	Matrix involved in real space part of the Ewald sum..........71
$\mathbf{M}^{(2)}$	Matrix involved in reciprocal space part of the Ewald sum....71
$\mathbf{N}$	Integer matrix used in Kraynik-Reinelt PBC 143
$\mathbf{T}$	Transformation matrix used in PMF 148
χ	Mixedness parameter...62
Δt	Time step size 51
δ	Dirac delta function..18
$\delta_{\mu\nu}$	Kronecker delta...53
$\dot{\epsilon}$	Elongation rate 59
$\dot{\Gamma}$	Characteristic strain rate..................................62
$\dot{\gamma}$	Shear rate 59
ϵ	Repulsive energy of the enthalpic origin between two monomers 25
η	Planar mixed flow viscosity 63
η_H	Generalized viscosity defined by Hounkonnou et al. (1992)....60
η_p	Polymer contribution to the viscosity.........................6
η_p^θ	Polymer contribution to the viscosity at theta condition......28
η_s	Solvent contribution to the viscosity..........................6
η_{PEF}	Planar elongational flow viscosity defined by Hunt et al. (2010)61

η_{PMF}	Planar mixed flow viscosity defined by Hunt et al. (2010)	60
η_{PSF}	Planar shear flow viscosity defined by Hunt et al. (2010)	61
$\eta_{\mathrm{p}}^{\star}$	Polymer contribution to the viscosity at overlap concentration	43
λ_H	Characteristic time scale used for nondimensionalizations	51
λ_{max}	Maximum eigenvalue of $\boldsymbol{\mathcal{D}}$	90
λ_{min}	Minimum eigenvalue of $\boldsymbol{\mathcal{D}}$	90
$\mathcal{N}_A$	Avogadro number	118
μ	Index used to indicate a bead	18
ν	Index used to indicate a bead	18
ν_{eff}	Effective exponent	121
ϕ_η	Universal crossover scaling function for viscosity	43
ϕ_D	Universal crossover scaling function for diffusivity	43
ϕ_R	Universal crossover scaling function for gyration radius	43
τ_1	The longest relaxation time of the polymer chain	28
τ_c	Crossover time used in the context of HI screening	34
τ_p	Strain period	137
τ_u	Universal scaling variable combining T and M_{w}	17
Θ	Theta temperature	17
θ_s	Angle used in planar shear flow	136

$\mathbf{b_i}$	Basis lattice vector in PEF $(i = 1, 2, 3)$	142
$\mathbf{b_i}'$	Basis lattice vector in PMF $(i = 1, 2, 3)$	149
$\mathbf{k}$	Wave vector	71
$\mathbf{L_i}$	Lattice vector $(i = 1, 2, 3)$	135
$\mathbf{n}$	Integer triplet used to account for periodic images	71
$\mathbf{r}_{\nu\mu}$	A vector connecting beads ν and μ	18
$\mathbf{r}_{\text{cm}}$	Center of mass of a polymer chain	56
$\mathbf{v}_{\nu}$	Velocity of the solvent at the bead ν	80
ξ_c	Size of a concentration blob	3
ξ_H	Hydrodynamic screening length	33
ξ_T	Size of a thermal blob	25
ζ	Hydrodynamic friction coefficient associated with a bead	51
a	Dimensionless bead radius	54
b	Size of a monomer	21
b_p	FENE parameter	52
c	Monomer concentration	4
$c^{\star}$	Overlap concentration	5
$c^{\star\star}$	Concentration to demarcate between the semidilute and the concentrated regimes	35

c_p	Chebyshev coefficients 89
c_{M}	Chemistry dependent parameter 18
D	Diffusivity 24
$D^{\star}$	Diffusivity at overlap concentration 43
$d^{\star}$	Dimensionless parameter used to measure the range of the excluded volume interaction 21
D^{θ}	Diffusivity at theta condition 28
D_0	Diffusivity at infinite dilution 6
D_0^{θ}	Diffusivity at infinite dilution and at theta condition 41
D_{K}	Kirkwood diffusivity 57
D_{min}	Minimum lattice spacing 152
D_{s}	Short time diffusivity 57
D_{Z}	Diffusivity in the dilute limit 114
E	Excluded volume potential 18
g	Number of monomers in a concentration blob 38
H	Spring constant in a bead-spring chain 18
$h^{\star}$	Hydrodynamic interaction parameter 54
k_B	Boltzmann constant 18
K_r	Reciprocal space cutoff in the Ewald sum 76

k_z	Chemistry dependent parameter	21
L	Edge length of a cubic simulation box	50
l_H	Characteristic length scale used for nondimensionalizations	20
M	Accuracy parameter used in the Ewald sum method	76
m	Number of thermal blobs within the concentration blob	36
M_d	The Madelung constant (2.8373)	79
M_{w}	Molecular weight of a polymer molecule	4
N	Total number of beads in a simulation box	50
n	Number of monomers within a thermal blob	25
$N^\star$	Number of overlapping beads	73
N_b	Number of beads in a bead-spring chain	18
N_c	Number of polymer chains in a simulation box	50
n_p	Number of polymer chains per unit volume	60
N_{Ch}	Number of terms used in the Chebyshev polynomial approximation	89
$N_{\mathrm{r_c}}$	Number of beads in a sphere of radius r_{c}	83
q_0	Maximum stretch of a spring	52
R_g	Gyration radius of a polymer molecule	5
R_g^0	Gyration radius at infinite dilution	6

$R_g^\star$	Gyration radius at overlap concentration	43
$R_g^{0,\theta}$	Gyration radius at infinite dilution and at theta condition	38
R_g^θ	Gyration radius at theta condition	27
r_c	A cutoff radius in the real space part of the Ewald sum	76
$S(k)$	Static Structure factor at wave number k	56
T	Temperature	4
t	Time	51
T_F	CPU cost to calculate the reciprocal space sum	83
T_R	CPU cost to calculate the real space sum	83
V	Volume of a simulation box	50
$v(T)$	Temperature dependent excluded volume parameter	18
Z	Total friction experienced by a polymer chain	23
z	Solvent quality parameter	20
$z^\star$	Dimensionless strength of the excluded volume interaction	20
$(\boldsymbol{\nabla v})$	Velocity gradient tensor	51
$\bar{\eta}_1$	Conventional definition of planar elongational flow viscosity	63
$\boldsymbol{\Omega}$	Hydrodynamic interaction tensor	53
$\mathbf{B}_{\nu\mu}$	3×3 matrix responsible for multiplicative noise for a fixed pair of beads μ and ν	51

$\mathbf{D}_{\nu\mu}$	3×3 diffusion matrix for a fixed pair of beads μ and ν	51
$\mathbf{F}^{c}$	Connector force between two beads	52
$\mathbf{F}_{\mu}$	Sum of all the non-hydrodynamic forces on bead μ	51
$\mathbf{F}_{\mu}^{\text{exv}}$	Excluded volume force on bead μ	51
$\mathbf{F}_{\mu}^{\text{spr}}$	Spring force on bead μ	51
$\varOmega_1$	RPY function	54
$\varOmega_2$	RPY function	54
$\boldsymbol{\Delta}\mathbf{W}$	Vector to denote Wiener process	51
$\mathbf{r}_{\nu}$	Position vector of a bead ν	51
MSD	Mean square displacement	58

Chapter 1

Introduction

1.1 Semidilute Polymer Solutions

Polymer molecules are building blocks for manufacturing paints, fibers, films, glues and many other products. Polymer molecules, when immersed in suitable solvents, form polymer solutions. Theoretical investigations of polymeric solutions have been extensively carried out for many years because of their interesting physical and chemical properties, and for their technological applications. At present, we have an excellent understanding of the dynamics of infinitely dilute polymer solutions, and of concentrated polymer solutions and melts. However, there is very little known about the vast regime of concentrations that lie in between; a regime that has, because of its unique behavior, a special name all of its own, the so-called *semidilute* concentration regime. Semidilute solutions are of significant interest in many practical applications and display highly interesting behavior. For example, the flow resistance is quite strong when a semidilute solution is subjected to an elongational flow. This has potential ramifications in many applications in which there is a strong elongational component to the deformation of the solution, including fiber spinning and coating flows. Studies of nanoscale motion such as the diffusion of globular proteins through media crowded with macromolecules has become an important area of research in cell biology since in a typical cell proteins function in the crowded cytoplasmic environment where thirty percent of the space is occupied by polymers of varying size and nature (Kozer et al., 2007; Mangenota et al., 2003). Obtaining a quantitative understanding of semidilute polymer solution dynamics is consequently not only of fundamental importance, but is also vitally important for a number of practical applications.

Today, a significant amount of literature can be found on the studies of dilute and concentrated polymer solutions because in either case, their behavior can be understood by understanding the behavior of single molecules. In the dilute case this is obvious since polymer molecules are far apart from each other. In the concentrated case, the prohibition of lateral motion of any particular molecule due to the presence of surrounding molecules can be represented as a tube surrounding the chain (Doi and Edwards, 1986). In the vast regime of semidilute polymer concentrations, the simplifications (of having to deal only with a single chain) made for dilute and concentrated regime are not afforded because of the inherently many-body interactions in the problem. It is important to appreciate that the onset of the semidilute regime occurs at surprisingly low concentrations because even though the monomer concentration is very low, their being strung together into polymers that are extended objects in space gives rise to the early emergence of interactions (Rubinstein and Colby, 2003).

In essence there are only two interactions that are significant. The first is the excluded volume interaction, which simply states that two monomers cannot occupy the same place at the same time. While this interaction is short-ranged in space, it is long-ranged along the backbones of the polymer chains since any two monomers on any two chains can interact with each other. The other significant interaction is hydrodynamic interaction, which is a solvent-mediated interaction that is long-ranged in space. When one part of a polymer moves, it disturbs the solvent close to it, and this disturbance is propagated to all the other chains in the system, leading to a coupling of all their motions. Any theoretical effort towards understanding the behavior of semidilute solutions must be able to account for and satisfactorily treat these two interactions.

Excluded volume and hydrodynamic interactions exist in dilute solutions as well, therefore it is well known how to incorporate them into molecular theories (Prabhakar and Prakash, 2004). In this case, however, it is sufficient to account for only intra-chain interactions because only a single polymer chain is considered. It is an amazing fact of polymer solution behavior that both excluded volume and hydrodynamic interactions *disappear* in concentrated solutions and melts (Rubinstein and Colby, 2003). As a consequence, one may think of semidilute solutions as the regime in which excluded volume and hydrodynamic interactions are gradually *screened* with increasing concentration.

Current understanding of semidilute solution behavior is mainly due to the theoretical scaling laws (de Gennes, 1976a,b, 1979; Muthukumar and Edwards, 1982a,b, 1983; Muthukumar, 1984; Richter et al., 1984; Edwards and Muthukumar, 1984; Doi and Edwards, 1986; Shiwa et al., 1988; Fredrickson and Helfand, 1990; R. H. Colby et al., 1994; Rubinstein and Colby, 2003) developed around the concepts of (i) *blobs models* (Daoud et al., 1975) and (ii) *screening* of different interactions. The blob model is based on identifying physically relevant length scales, and formulating different scaling laws for different observables using these length scales. Basically, a semidilute solution is characterized by one such length scale ξ_c, representing the size of a *concentration blob*. Within a concentration blob, segments on a polymer chain are ignorant of the presence of other chains and behave as though they are in a dilute solution. As a result, both excluded volume and hydrodynamic interactions are present. On length scales larger than ξ_c, both these interactions are screened and the chains behave as though they are in a concentrated solution or melt. This simple demarcation of a semidilute solution through the concept of a concentration blob into two known regimes of dilute and concentrated solution behavior enables the development, particularly at equilibrium and close to equilibrium, of a number of scaling predictions in terms of the key independent variables.

The key variables that determine the behavior of a polymer solution at equilibrium are the molecular weight of the polymer molecule M_{w}, the monomer concentration c (which is defined as number of monomers per unit volume), and the temperature T. Therefore, one must study and understand various physical properties as functions of these key variables.

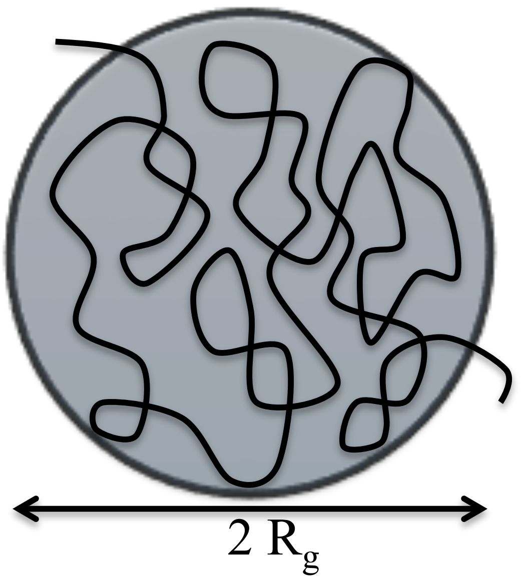


Figure 1.1: Volume of a sphere of radius R_g can be used as an approximation for the volume of a polymer chain

The volume of a linear polymer chain can be approximated as the volume of a sphere of radius R_g as illustrated in Fig. 1.1, where R_g is the gyration radius of a polymer molecule. At low polymer concentrations, as shown in Fig. 1.2 (a), these spheres of radius R_g are far apart and do not interact with each other. Upon increasing the polymer concentration, polymer chains become congested and interact with each other. At a certain polymer concentration, the monomer concentration c becomes equal to the so called *overlap concentration* $c^\star$. At the overlap concentration, individual polymer chains just begin to *touch* each other, and the whole volume of the system is filled with spheres of radius R_g, as depicted in Fig. 1.2 (b). In this scenario, the monomer concentration inside a sphere is equal to the monomer concentration of the whole system. The onset of

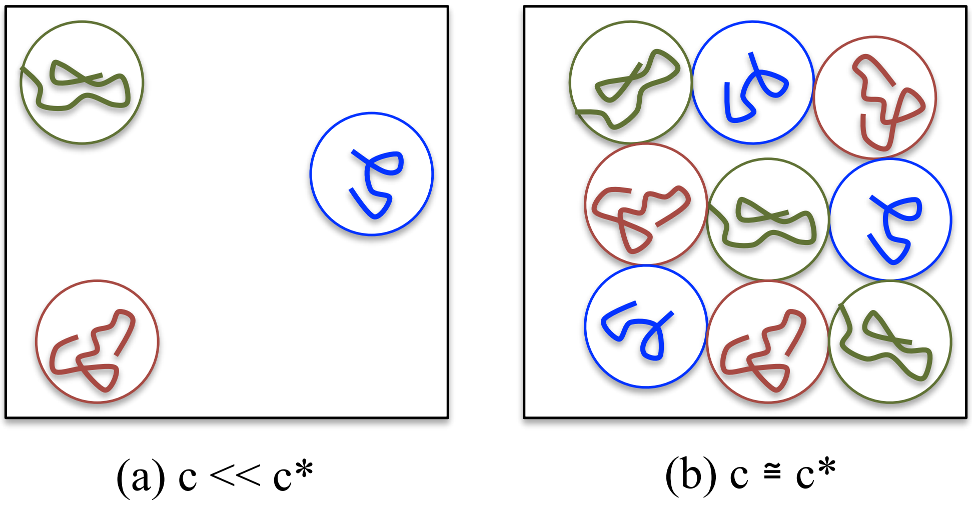


Figure 1.2: State of polymer chains in solutions with different polymer concentrations

the semidilute solution regime is believed to occur at the overlap concentration $c^\star$. Clearly, the larger the molecular weight, the larger the chain size, and the smaller the value of $c^\star$. In fact, it turns out it is possible to eliminate the explicit dependence on M_w altogether, and describe the behavior completely in terms of the scaled variable $c/c^\star$ (Rubinstein and Colby, 2003).

The dependence on temperature is subtler. At high enough temperatures,

Property	Theta solvent	Good solvent
Gyration radius	$\dfrac{R_g}{R_g^0} \sim \left(\dfrac{c}{c^\star}\right)^{0}$	$\dfrac{R_g}{R_g^0} \sim \left(\dfrac{c}{c^\star}\right)^{-0.125}$
Zero-shear rate viscosity	$\dfrac{\eta_p}{\eta_s} \sim \left(\dfrac{c}{c^\star}\right)^{2}$	$\dfrac{\eta_p}{\eta_s} \sim \left(\dfrac{c}{c^\star}\right)^{1.25}$
Diffusion coefficient	$\dfrac{D}{D_0} \sim \left(\dfrac{c}{c^\star}\right)^{-1}$	$\dfrac{D}{D_0} \sim \left(\dfrac{c}{c^\star}\right)^{-0.5}$

Table 1.1: Scaling predictions for semidilute solutions in theta and good solvents.

the enthalpic interactions between monomers and solvent molecules are favorable, so a polymer coil swells to occupy as large a size as is permitted within the constraints of monomer connectivity. This good solvent regime vanishes as the temperature is lowered and polymer-solvent interactions become less favorable. At a unique temperature called the *theta* temperature, the desire for the polymer coil to swell due to entropic considerations is just balanced by the unfavorable enthalpic interactions with the solvent, giving rise to a unique scaling regime called the theta regime. Using the concept of a concentration blob, scaling predictions in theta and good solvents can be developed. For instance, the gyration radius, the zero-shear rate viscosity and the self-diffusion coefficient in semidilute solutions are predicted to scale with $c/c^\star$ as shown in Table 1.1. In this Table, R_g^0 is the gyration radius at infinite dilution, η_p and η_s are the polymer and solvent contributions to viscosity, and D and D_0 are self-diffusion coefficients, with D_0 representing the value in a dilute solution at theta condition. These scaling laws, that govern the static and near-equilibrium dynamic properties, have been subjected to a wealth of carefully executed experiments (Rubinstein and Colby, 2003; Wiltzius et al., 1984; Ewen and Richter, 1997) and computer simulations

(W. Paul et al., 1991; Müller et al., 2000; Huang et al., 2010; Ahlrichs et al., 2001).

Unfortunately, these scaling laws are only valid in the limits of the theta and good solvent regimes, with no predictions available in the large crossover regime between theta and good solvents (Grosberg and Khokhlov, 1994; Schäfer, 1999). Therefore, it is clearly also important to study the crossover driven by temperature in semidilute solutions in more detail. Indeed there have been only few investigations (based on scaling theory) that have tried to study the *double* crossover driven by both concentration and temperature, in particular when dynamic properties are concerned (Daivis and Pinder, 1990). These have not been examined by any computer simulations either because of the complexity of the problem.

The aim of this thesis is to explore the universal crossover scaling functions in the semidilute regime by studying both the concentration driven crossover and the temperature driven crossover simultaneously, with the help of computer simulations. In particular, we wish to discover the specific form of these scaling functions, and address the question: do these scaling functions bear any resemblance to the purely concentration driven crossover scaling functions? We address this question by developing a mesoscopic Brownian dynamics simulation algorithm, and examining universal behavior in the long-chain limit.

1.2 Mesoscopic Simulations

In recent years, significant progress in the development of mesoscopic simulation techniques, which allow the exploitation of underlying theories without the need for approximations, (Ahlrichs et al., 2001; Stoltz et al., 2006; Huang et al., 2010) has made it possible for the first time to obtain detailed predictions of equilibrium and nonequilibrium properties that can be compared with experimental observations. The successful implementation of mesoscopic simulations has been made possible through the use of algorithms that enable an accurate depiction of the semidilute regime. Essentially, this requires the ability to describe long polymers that overlap with each other, while maintaining a low segment density. Further, the segments must be capable of interacting with each other through solvent mediated hydrodynamic interactions (Kirkwood and Riseman, 1948; Freed and Edwards, 1974; de Gennes, 1976b; Bixon, 1976; Ahlrichs et al., 2001). Three different mesoscopic simulation methods, all of which use coarse-grained bead-spring chain models for polymer molecules, have been developed recently that achieve these objectives. Two of these techniques, namely, the hybrid Lattice Boltzmann/Molecular Dynamics (LB/MD) method (Ahlrichs and Dünweg, 1999; Dünweg and Ladd, 2009) and the hybrid Multi-particle Collision Dynamics/Molecular Dynamics (MPCD) method treat the solvent explicitly (Malevanets and Kapral, 1999; Kapral, 2008; Gompper et al., 2009). As a consequence, hydrodynamic interactions between polymer segments arise naturally through the exchange of momentum between the beads on a chain and solvent molecules. In the third approach (Stoltz et al., 2006), which is based on Brownian dynamics (BD) simulations (Ottinger, 1996), the solvent degrees of freedom are removed completely, but their effect is taken into account through long-range dynamic correlations in the stochastic displacements of the beads.

The very nature of semidilute polymer solutions, particularly the need to use periodic boundary conditions to describe homogeneous polymer solutions in unbounded domains, necessitates the simulation of a large number of particles. As a result, the computational efficiency of a simulation technique becomes an important consideration. One of the aim of this thesis is to implement an *optimized* BD algorithm for semidilute polymer solutions by efficiently treating the long-ranged hydrodynamic interactions. Moreover, the BD algorithm developed by Stoltz et al. (2006) is not applicable when there is particle-particle overlap, and hence is restricted to the simulation of semidilute solutions in good solvent conditions. The aim here is to develop a BD algorithm capable of simulating semidilute solutions at the θ temperature, which is essential in order to study the temperature driven crossover regime, and also in addressing the question regarding the form of the crossover scaling functions in the double crossover regime.

1.3 Far-From-Equilibrium Simulations

While it is important to study the behavior of semidilute polymer solutions at equilibrium, it is equally important to understand far from equilibrium behavior, because for many practical applications, polymer solutions are processed under different flow conditions. The great advantage of BD simulations is that the same algorithm can be used for simulating both equilibrium systems, and systems subject to flow. In the case of dilute solutions, this is typically accomplished by incorporating a term in the stochastic differential equation that accounts for flow. However, this is highly nontrivial for polymer solutions at finite concentrations, in particular for infinite systems treated with periodic boundary conditions. Unlike the simulation of equilibrium system where periodic boundary conditions (PBCs) are used in an orthogonal cell to get rid of wall effects, for simulating far from equilibrium systems, appropriate PBCs need to be used such that the following two requirements are met: (i) The PBCs should be compatible with any particular flow and (ii) the simulation should be capable of running for an arbitrary amount of time. PBCs for planar shear flow and planar elongational flow were developed by Lees and Edwards (1972) and Kraynik and Reinelt

(1992), respectively, such that the two requirements mentioned above were fulfilled. Lees-Edwards and Kraynik-Reinelt PBCs have been used by Bhupathiraju et al. (1996); Todd and Daivis (1998) in their nonequilibrium molecular dynamics (NEMD) simulation algorithm. Other than NEMD simulations, these PBCs have also been implemented in a BD algorithm by Stoltz et al. (2006) to simulate semidilute polymer solutions undergoing planar shear or planar elongational flows.

In real flow situations, however, rather than only shear or elongational flow, a combination of these flows is often observed. Woo and Shaqfeh (2003); Dua and Cherayil (2003) and Hoffman and Shaqfeh (2007) have simulated dilute polymer solutions in planar mixed flow using a BD algorithm in which PBCs were not required. In a recent paper, Hunt et al. (2010) have derived suitable PBCs for *planar mixed flow* (which is a linear combination of planar elongational flow and planar shear flow) and implemented them in their nonequilibrium molecular dynamics (NEMD) simulation algorithm. To our knowledge, mixed flows of semidilute polymer solutions have not been studied so far. We aim to implement, for the first time, PBCs for planar mixed flow (Hunt et al., 2010) in a multi-chain Brownian dynamics simulation algorithm, which will enable us to simulate semidilute polymer solutions undergoing different kinds of flows.

1.4 Objectives

The broad objectives of this thesis can be listed as the following set of tasks:

1. Derivation of universal crossover scaling functions based on the blob model.

2. Development of a Brownian dynamics simulation algorithm for semidilute polymer solutions at equilibrium.

3. Verification of the crossover scaling functions predicted by scaling theory using Brownian dynamics simulations.

4. Implementation of mixed flow in a Brownian dynamics simulation algorithm.

The structure of the thesis is as follows: In order to set the stage for deriving universal crossover scaling laws for semidilute solutions, various concepts such as the blob model, excluded volume and hydrodynamic interactions etc . . . are introduced in Chapter 2. In Chapter 3, these concepts are used to derive scaling functions for semidilute solutions in the double crossover regime. The governing equations for the BD simulation algorithm are described in Chapter 4, while Chapter 5 focuses on the implementation of different terms in the BD simulation algorithm at equilibrium. Optimization studies that have been carried out to obtain a fast BD simulation algorithm are also discussed in Chapter 5. The various predictions of scaling theory in the double crossover regime are verified with the help of BD simulations in Chapter 6. Chapter 7 describes the implementation of mixed flows in a BD simulation algorithm, and the preliminary results we have obtained for the viscosity of a semidilute solution under mixed flow conditions are discussed. Finally, conclusions of the thesis are presented in Chapter 8.

Chapter 2

Scaling Theory for Dilute Polymer Solutions Near Equilibrium

The behavior of dilute polymer solutions near equilibrium has been extensively studied by experiments, theory and computer simulations, and there are many monographs that provide an excellent summary of our current understanding of these systems (Bird et al., 1987; de Gennes, 1979; Doi and Edwards, 1986; Rubinstein and Colby, 2003). In this chapter, we briefly discuss the scaling theory used to derive scaling laws for various properties in dilute polymer solutions at equilibrium. Though we are interested in semidilute solutions, a discussion of scaling concepts in the context of dilute solutions is required to set up the basic framework for developing scaling laws in the semidilute regime, presented in Chapter 3.

Scaling laws for dilute solutions account for two important pieces of physics that occur on the microscopic scale, namely, excluded volume (EV) interactions and hydrodynamic interactions (HI) (Bird et al., 1987; de Gennes, 1979; Doi and

Edwards, 1986; Larson, 1988; Prakash and Öttinger, 1999; Prakash, 1999, 2009). Excluded volume interactions are first briefly discussed in Section 2.1, followed by a discussion of hydrodynamics interactions in Section 2.2. Finally, scaling laws for various properties are derived for dilute polymer solutions in Section 2.3.

2.1 Excluded Volume Interactions

Excluded volume interactions play an important role in determining the static behavior of polymer solutions (de Gennes, 1979; Doi and Edwards, 1986; Rubinstein and Colby, 2003). The physics of excluded volume interactions can be understood as follows. When any two monomers of a polymer molecule come close to each other, a strong mutual repulsion is felt by them. This repulsion arises because a monomer cannot occupy space that is already occupied by another monomer of the same polymer molecule at the same time. As seen in Fig.

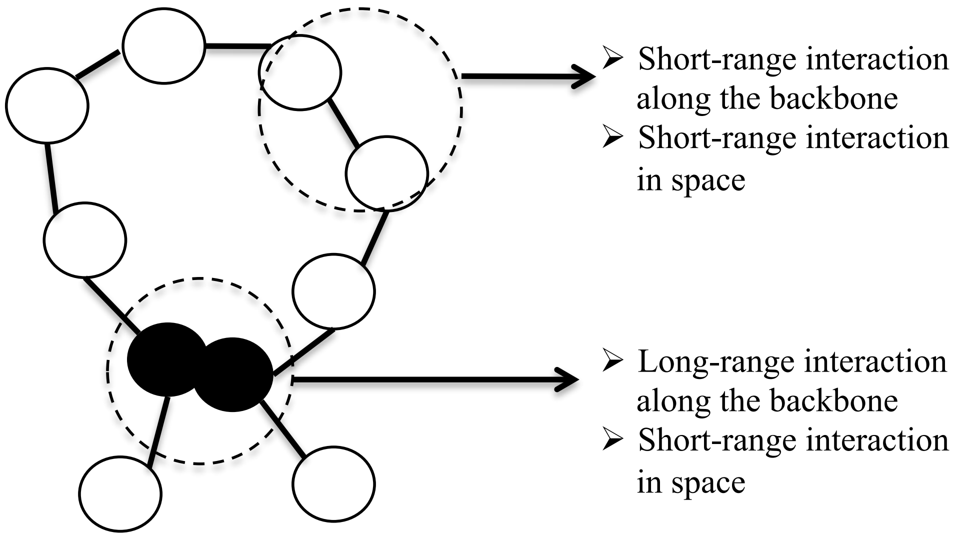

➢ Short-range interaction along the backbone
➢ Short-range interaction in space
➢ Long-range interaction along the backbone
➢ Short-range interaction in space

Figure 2.1: Illustration of excluded volume interactions in a polymer molecule

2.1, the two black monomers cannot overlap each other because of the repulsive interaction between them.

In general, the *effective* interaction between a pair of monomers can be either repulsive or attractive, depending on the difference between the strengths of monomer-monomer and monomer-solvent interactions. When the monomer-monomer energy is lower than the monomer-solvent energy, monomers like to be near each other, and this means that the effective interaction between monomers is attractive. Note that by energy we mean the energy $u(r)$ required to bring two particles (that can be either monomer or solvent particles) from ∞ to within distance r of each other (Rubinstein and Colby, 2003). On the other hand, when the energy between monomer and solvent is lower than that between monomer and monomer, monomers like to be surrounded by solvent molecules, leading to the effective interaction between monomers being repulsive. This effective interaction, which includes the repulsive and attractive parts of monomer-monomer interaction, is known as excluded volume interaction (Flory, 1953). Excluded volume interactions are either short-range or long-range along the backbone of a polymer molecule but they are always short-range in space.

Clearly the structure of a polymer molecule depends on the relative strength of monomer-monomer and monomer-solvent interactions. For example, the average size of a polymer molecule tends to increase due to the monomer-monomer repulsion, but at the same time, this swelling of a molecule is opposed by solvent-mediated attractions. The thermodynamic state of the solvent plays an important role in monomer-solvent interactions, and the temperature is a controlling parameter in such interactions (Flory, 1953; Schäfer, 1999). For instance, monomer-solvent interactions are energetically unfavorable at low temperatures, and the polymer solution is said to be under *poor solvent* conditions. This leads to strong enough monomer-monomer attractions that dominate monomer-monomer repulsions, and the polymer molecule collapses to a globule like structure. Upon increasing the temperature, the strength of monomer-monomer attractions reduces, and therefore, the average size of a polymer molecule increases. The term *good solvent* is used to describe a condition where the temperature is high enough for the repulsive interactions to dominate over attractive interactions. There exists an intermediate temperature between poor solvent and good solvent conditions, the so called *theta* temperature, at which the attractive and repulsive parts of monomer-monomer interactions balance each other. Therefore, at the theta temperature, the size of a polymer molecule is the same as that of an *ideal* polymer molecule (Doi and Edwards, 1986). Depending on the monomer-solvent interactions, or temperature, three different cases can be visualized as shown in Fig. 2.2. It is well known that in the theta limit, where polymer configurations

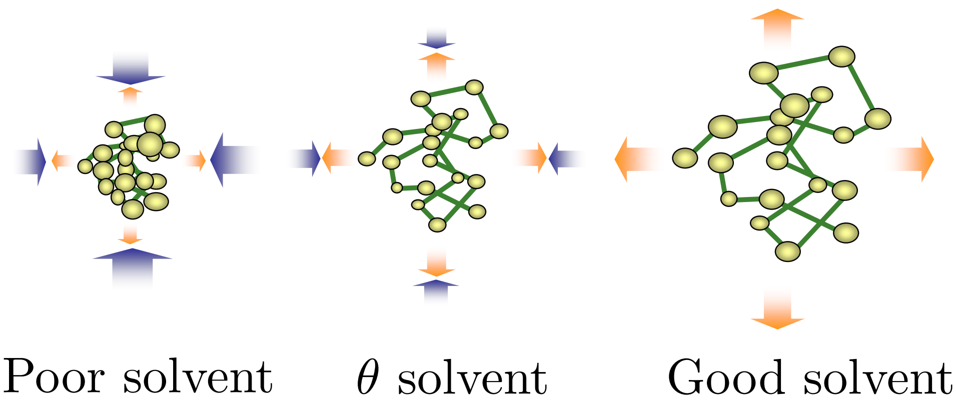


Figure 2.2: Monomer-solvent interactions affect the structure of a polymer molecule

obey random walk (RW) statistics, the size of a polymer molecule scales as $M_{\rm w}^{1/2}$. This power law is also referred to as the Gaussian scaling law for polymer size. The exponent in this power law is universal and hence is (1/2) for all theta solvents. However, the prefactor depends on the chemistry of the polymer-solvent system.

In the good solvent limit, on the other hand, the polymer is considered to follow self avoiding walk (SAW) statistics. In this case, a similar power law governs the polymer size, but with a nontrivial exponent $\nu \cong 3/5$, famously known as the Flory exponent [the precise value is 0.587597 (7) (Clisby, 2010)]. This exponent is again universal in the good solvent limit and hence is independent of temperature or chemistry. The size of a polymer molecule in a good solvent, therefore, scales as M_w^ν. This power law is referred to as the Kuhnian scaling law for polymer size. Similar to the theta case, the prefactor in this power law contains all the information regarding chemistry or temperature. This power law is valid only in the excluded volume limit, when the molecular weight is sufficiently large, with the effective interaction being repulsive or with the temperature T being not too close to the theta temperature. The repulsive interaction decreases as T approaches the theta temperature. In order for the excluded volume power law to be valid at a lower temperature, the polymer must have an even larger molecular weight. In other words, the polymer solution departs from the excluded volume limit if either T is reduced towards the theta temperature at a fix M_w or M_w is decreased by keeping T fixed.

In the temperature crossover regime, between θ and good solvents, where simple power laws are not valid, T and M_w are generally combined to form a single scaling variable τ_u, and a type of scaling still persists (Schäfer, 1999). The scaling variable τ_u has the form

$$\tau_u = \left(1 - \frac{\Theta}{T}\right) M_\mathrm{w}^{1/2} \tag{2.1}$$

where, Θ is the theta temperature.

An example of the advantage of using the scaling variable τ_u can be seen in the work of Miyaki and Fujita (1981), who performed experiments to measure the swelling of polystyrene in Benzene at 25 and 30^0 C, in methyl ethyl ketone (MEK) at 35^0 C, and in Cyclohexane at the temperatures indicated in Fig. 2.3 (a). The variable in the y-axis is the swelling of a polymer molecule $\alpha_g(T, M_\text{w})$ defined as the ratio of the polymer size at temperature T to that in a theta solvent.

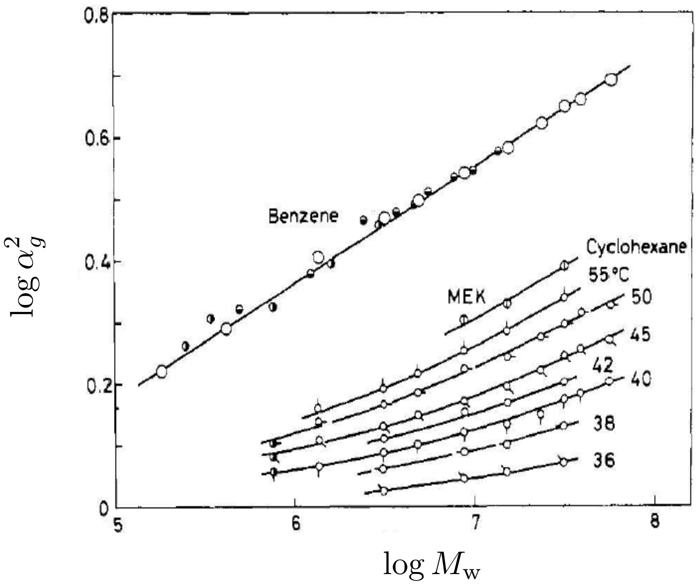
When α_g^2 is plotted as a function of $M^\star$ in a $\log - \log$ plot, where $M^\star = c_\text{M} \tau_u^2$ and c_M is a chemistry dependent parameter, it can be seen in Fig. 2.3 (b) that universal behavior can be obtained for all values of M_w and temperature $T > \Theta$, by merely using proper values of c_M (Miyaki and Fujita, 1981).

Universal behavior in the temperature crossover regime has also been observed using theoretical models. For instance, renormalization group (RG) methods have been applied to successfully predict the entire range of behavior exhibited by static solution properties (Freed, 1987; Schäfer, 1999; Cloizeaux and Jannink, 2010). In these theoretical models, polymer molecules are represented by a coarse-grained model, such as a bead-spring chain (Bird et al., 1987), consisting of N_b beads connected together by $N_b - 1$ Hookean springs (with a spring constant H). Each bead consists of several monomers, and hence N_b is directly proportional to the molecular weight of a polymer molecule.

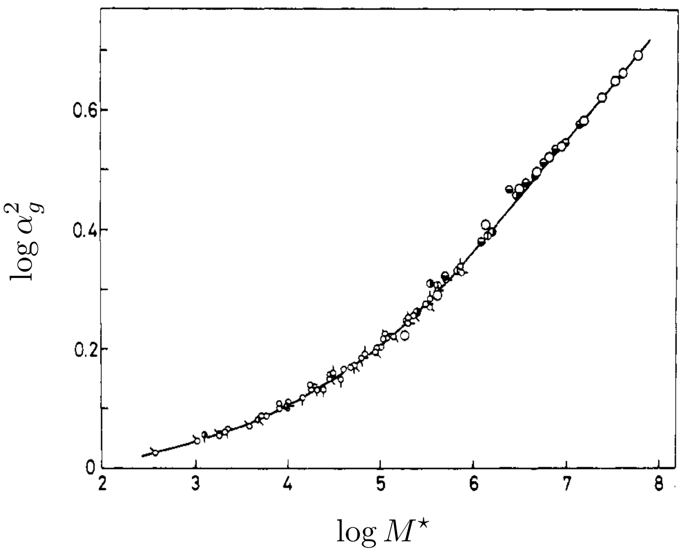
The presence of excluded volume interactions is usually taken into account in theoretical models by assuming the existence of a Dirac delta function repulsive potential, which acts pairwise between the beads of the chain

$$E\left(\mathbf{r}_{\nu\mu}\right) = v(T)\, k_\text{B} T\, \delta(\mathbf{r}_{\nu\mu}) \tag{2.2}$$

where, $\mathbf{r}_{\nu\mu}$ is a vector connecting beads ν and μ, $v(T) = 1 - \Theta/T$ is the temperature dependent excluded volume parameter (Doi and Edwards, 1986) and k_B is Boltzmann's constant. By defining a length scale $l_H = \sqrt{k_B T/H}$, and



(a)



(b)

Figure 2.3: Experimental data on polymer swelling reproduced from Miyaki and Fujita (1981): (a) for various temperatures and molecular weights and (b) collapse of all the data by plotting $\log \alpha_g^2$ as a function of $\log M^\star$, where $M^\star$ is a reduced molecular weight defined by Miyaki and Fujita (1981). Note that the parameter $M^\star$ can be related to τ_u through $M^\star = c_M \tau_u^2$, where c_M is a chemistry dependent parameter, which is used to make a shift in the horizontal axis in order to collapse all the data points onto a master curve. Reprinted (adapted) with permission from Miyaki and Fujita (1981). Copyright (1981) American Chemical Society.

a dimensionless strength of the excluded volume interaction $z^\star = v(2\pi l_H^2)^{-3/2}$, the temperature and the chain length variables can be combined into a single dimensionless parameter $z = z^\star \sqrt{N_b}$ (Yamakawa, 1971) such that $z \propto \tau_u$ (see Eq. (2.1)).

RG theories predict the existence of power laws in the limit $z \to 0$ (corresponding to a theta solvent) and in the good solvent limit $z \to \infty$, completely consistently with experimental observations. Significantly, the existence of scaling functions that depend only on the parameter z, is also predicted by RG theory. These predictions accurately describe the temperature or solvent quality driven crossover behavior of all static properties between these two asymptotic limits, i.e., in the domain $0 < z < \infty$ (Schäfer, 1999).

Though RG theories have been successful in many aspects, they are approximate since they are based on perturbation theory. An exact solution for the excluded volume problem becomes possible by using Monte Carlo simulations (Li et al., 1995; Graessley et al., 1999), based on excluded volume potentials with a finite range of excluded volume interactions, such as the Lennard-Jones potential. However, with such potentials, the parameter z (which is the true measure of solvent quality) does not appear naturally in the equations, making it difficult to study the crossover behavior of static properties.

Kumar and Prakash (2003) have developed a scheme whereby it is possible to obtain the dependence of static properties on the crossover variable z in BD simulations. There are two key aspects to their procedure. The first aspect of their approach is the use of a narrow Gaussian potential (Ottinger, 1996) to represent excluded volume interactions,

$$E\left(\mathbf{r}_{\nu\mu}\right) = z^\star\, k_{\mathrm{B}}T \, \left(\frac{1}{d^{\star 3}}\right) \exp\left\{-\frac{1}{2\,b^2}\,\frac{\mathbf{r}_{\nu\mu}^2}{d^{\star 2}}\right\} \tag{2.3}$$

where, $d^\star$ is a dimensionless parameter that measures the range of the excluded volume interaction, and b is the monomer size. In the limit $d^\star \to 0$, the narrow Gaussian potential reduces to δ-function potential, and is consequently a means of regularizing the δ potential.

The second aspect is their procedure for finding the model's predictions in the limit of infinite chain length. In their simulations, data for larger and larger values of N_b are accumulated, and the infinite chain length limit is then found by extrapolating the finite chain data to $N_b \to \infty$.

Kumar and Prakash (2003) have shown that in this limit, $d^\star$ is an irrelevant variable, and that the only relevant variable is the scaling variable z. The results of their BD simulations for α_g^2 at various values of z are displayed as the solid red line in Fig. 2.4. The experimental data of Miyaki and Fujita (1981), is plotted along with the simulation results in Fig. 2.4. By considering $z = k_z \tau_u$, where k_z is a chemistry dependent parameter, all the experimental data can be collapsed onto the simulation curve by making a simple horizontal shift to the experimental data. Details of the shifting procedure are given in Kumar and Prakash (2003).

The extended discussion of excluded volume interactions in this section has served two purposes. Firstly, it helps introduce the solvent quality variable z, which is used in the scaling theories for dilute and semidilute solutions. Secondly, it helps to introduce the narrow Gaussian potential and the extrapolation procedure that we have used subsequently in our simulations.

2.2 Hydrodynamic Interactions

In a polymer chain, each segment experiences a drag force because of friction with solvent molecules caused by their relative motion. When any one segment of a polymer chain moves, the velocity field of the solvent near all the other segments

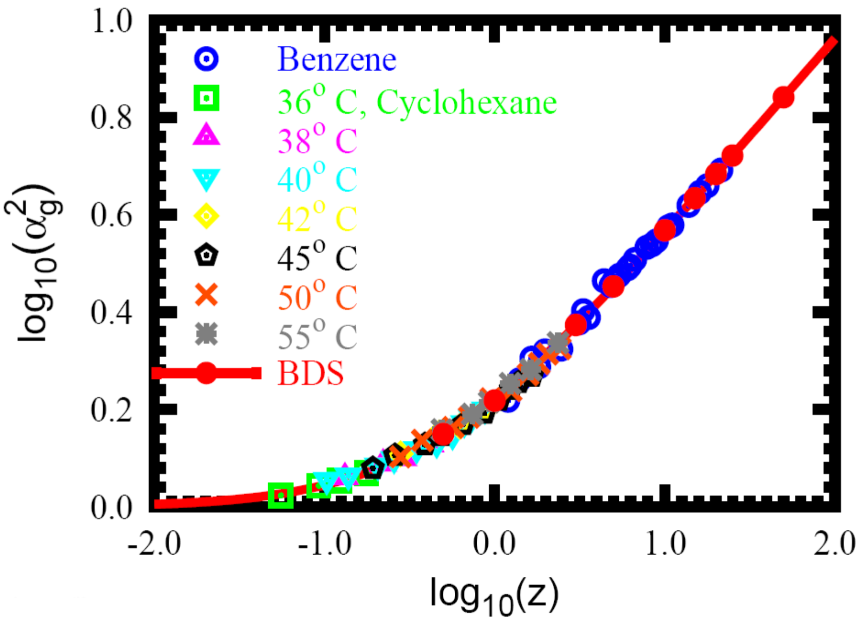


Figure 2.4: Experimental data for swelling obtained by Miyaki and Fujita (1981) collapse onto the BD simulation results obtained by Kumar and Prakash (2003). Reprinted (adapted) with permission from Miyaki and Fujita (1981) and Kumar and Prakash (2003). Copyright (1981), (2003) American Chemical Society.

are disturbed due to fast diffusive momentum transport by solvent molecules. This disturbance in the velocity field modifies the drag force on other segments, which eventually affects the motion of those segments. This interaction is known as hydrodynamic interaction, and is shown schematically in Fig. 2.5. The phenomenon of hydrodynamic interactions is a *long-range* effect, which means that a strong coupling of motions of segments are unavoidable even for those segments that are far apart along the backbone of the chain, and far apart in space as well. An example of a dynamic property is the diffusivity of a polymer molecule, D, which is proportional to the mean squared displacement of the chain with time. Accurate prediction of diffusivity, through theoretical approach, is obtained only when HI is included (Zimm, 1956).

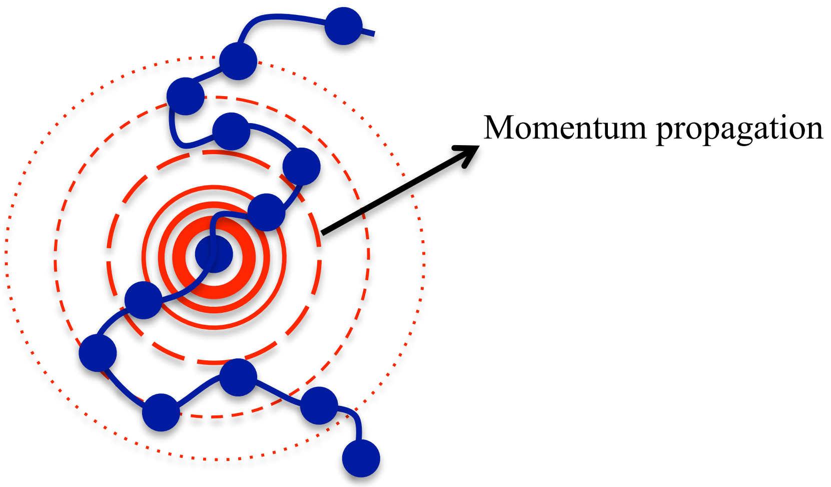


Figure 2.5: Illustration of long-range hydrodynamic interactions for a polymer molecule immersed in a pool of solvent

The simplest theoretical model for dilute polymer solutions, the so-called Rouse model, does not account for HI, and as a result fails to accurately predict dynamic properties. However, since HI is absent in a melt or in concentrated solutions, the Rouse model turns out to be useful to model low molecular weight melts where topological constraints are not yet important. As we shall see subsequently in Chapter 3, it is also useful for the description of semidilute solutions where HI is screened.

It is instructive in this context to consider the prediction of diffusivity (which is a dynamic property) by the Rouse model. By the Nernst-Einstein equation (Bird et al., 1987)

$$D \sim \frac{k_B T}{Z} \tag{2.4}$$

where, Z is the total friction experienced by a polymer chain. The total friction Z is the sum of the friction experienced by parts of the polymer chain exposed to the flow field. Since, in Rouse theory all the monomers are exposed to the flow field, the total friction is the sum of the friction experienced by all the monomers, and as a result Z scales as the molecular weight $M_{\rm w}$ of the chain. This also implies that the diffusivity in Rouse theory scales as $M_{\rm w}^{-1}$. This prediction is contrary to experimental observations, which suggest that D scales as $M_{\rm w}^{-\nu}$.

The introduction by Zimm (1956) of HI in an equilibrium averaged form into a molecular model for polymer solutions led for the first time to an accurate description of dynamic properties close to equilibrium. The Zimm theory does not capture the dynamics of polymer solutions away from equilibrium since it does not account for fluctuations in HI (Öttinger, 1989; Prakash, 1999). As will be discussed in Chapter 4, fluctuations in HI can be taken into account in BD simulations without making any averaging approximations. For the purpose of scaling theory, however, the Zimm theory is perfectly adequate.

The key result of the Zimm theory is that in dilute solutions, the total friction of the polymer chain is that experienced by the polymer coil as a whole, and not that of all the monomers together. In some sense, the monomers inside the polymer molecule are shielded from the flow (de Gennes, 1979; Rubinstein and Colby, 2003). As a result Z is proportional to the size of the polymer molecule R_g, and the diffusivity then takes the form

$$D \sim \frac{k_B T}{\eta_s R_g} \tag{2.5}$$

Since $R_g \sim M_{\rm w}^{\nu}$, the diffusivity in Zimm theory scales as $M_{\rm w}^{-\nu}$, in excellent agreement with experimental observations.

As we shall see in the scaling theory for semidilute solutions developed in Chapter 3, we use Rouse scaling when HI is screened and Zimm scaling when it is present. In the next section however, we use the concepts of solvent quality and Zimm scaling to derive some scaling laws for dilute polymer solutions.

2.3 Scaling Theory

The scaling theory for dilute polymer solutions is based on the concept of *thermal blobs* (de Gennes, 1979; Rubinstein and Colby, 2003; Grosberg and Khokhlov, 1994). The thermal blob is associated with a balance between thermal energy and energetic contributions due to excluded volume interactions . The size of the thermal blob ξ_T characterizes the length scale at which the energy due to excluded volume interactions amongst all the monomers lying within a blob becomes equal to the thermal energy k_BT. This implies that the conformations of the chain inside a thermal blob are unperturbed by excluded volume effects. As a result, the polymer chain obeys random walk (RW) statistics within a thermal blob. On the other hand, for length scales greater than ξ_T, excluded volume interactions become important and as a result, thermal blobs obey self avoiding walk (SAW) statistics, as shown schematically in Fig. 2.6.

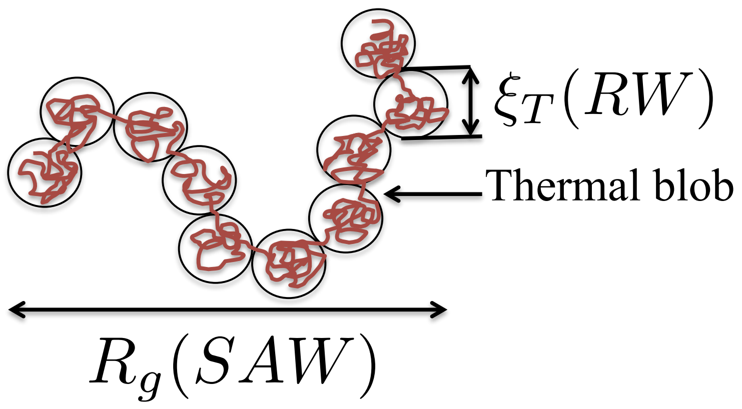


Figure 2.6: Thermal blobs in dilute solution

The size of the thermal blob ξ_T can be estimated as follows. Two monomers are assumed to interact with an energy $\epsilon\left(1-\dfrac{\Theta}{T}\right)$, where ϵ (> 0) is the repulsive energy of enthalpic origin. We assume that there are n monomers within a thermal blob. Since the number of contacts in a RW are of order $n^{1/2}$ in three spatial dimensions, it is straightforward to see that the excluded volume energy due to all the interactions within a thermal blob is of order $n^{1/2}\,\epsilon\left(1-\dfrac{\Theta}{T}\right)$. From the definition of a thermal blob, this energy must be of the same order as k_BT. Therefore, the condition to find n is

$$\epsilon\left(1-\frac{\Theta}{T}\right)n^{1/2} = k_BT \tag{2.6}$$

The excluded volume interaction energy for a pair of monomers, $\epsilon\left(1-\dfrac{\Theta}{T}\right)$, nondimensionalized by k_BT, can be equated to the dimensionless parameter $z^\star$, which is the parameter in the narrow Gaussian potential which represents the non-dimensional strength of pair-wise excluded volume interactions (see Eq. (2.3)). As a result,

$$z^\star = \frac{\epsilon}{k_BT}\left(1-\frac{\Theta}{T}\right) \tag{2.7}$$

From Eqs. (2.6) and (2.7), it follows that,

$$n = (z^\star)^{-2} \tag{2.8}$$

The corresponding blob size is consequently (since RW statistics are obeyed within a blob)

$$\xi_T = bn^{1/2} = b\,(z^\star)^{-1} \tag{2.9}$$

Using the definition of the solvent quality parameter, as mentioned in Section 2.1,

$$z = z^\star N_b^{1/2} \tag{2.10}$$

the nondimensional blob size in terms of the scaled variable z is,

$$\frac{\xi_T}{bN_b^{1/2}} = \frac{\xi_T}{R_g^\theta} = z^{-1} \tag{2.11}$$

where, $R_g^\theta = bN_b^{1/2}$ is the gyration radius at theta condition. The solvent quality can be seen to represent the ratio of the size of the chain in a θ solvent to the size of the thermal blob. Clearly, decreasing z increases ξ_T, and in the θ limit

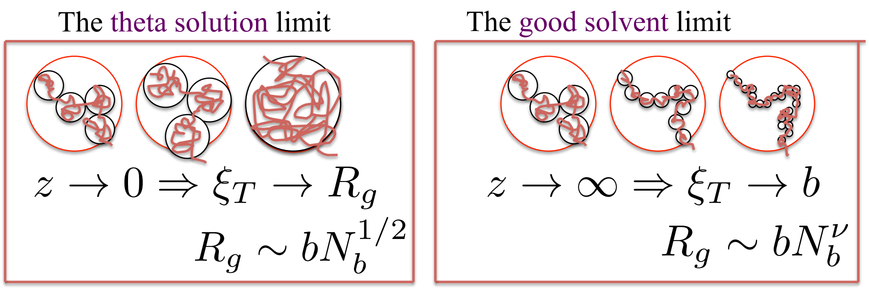


Figure 2.7: Changing z changes the size of a thermal blob ξ_T

the size of the thermal blob becomes larger than the size of a polymer chain. On the other hand, increasing z decreases ξ_T, and in the very good solvent limit ξ_T becomes equal to the size of a monomer b. This can be visualized as shown in Fig. 2.7.

The scaling relation for the gyration radius R_g can be derived in terms of the scaled variable z as follows. Since the chain obeys SAW statistics on length scales larger than ξ_T,

$$R_g = \xi_T \left(\frac{N_b}{n}\right)^\nu \tag{2.12}$$

where, (N_b/n) is the number of thermal blobs. Using Eqs. (2.8) and (2.9), leads to

$$R_g = bN_b^\nu (z^\star)^{2\nu-1} \tag{2.13}$$

The scaling relation in terms of scaled variables can be obtained by making use of Eq. (2.10), and the definition of R_g^θ

$$\frac{R_g}{R_g^\theta} = z^{2\nu-1} \tag{2.14}$$

As discussed earlier, Zimm dynamics are relevant on all length scales in the dilute regime. Using Zimm's scaling law for diffusivity from Eq. (2.5) and R_g from Eq. (2.13), we can write

$$D = \frac{k_B T}{\eta_s \, b N_b^{\nu} (z^{\star})^{2\nu - 1}} \tag{2.15}$$

Defining $D^{\theta} = \dfrac{k_B T}{\eta_s R_g^{\theta}}$, it follows from Eq. (2.10) that the ratio of diffusivities is given in terms of scaled variables by

$$\frac{D}{D^{\theta}} = z^{-(2\nu - 1)} \tag{2.16}$$

Once the expressions for R_g and D are known, the scaling laws for η_p, the polymer contribution to viscosity, can be derived from the relation

$$\eta_p = k_B T (c/N_b) \tau_1 \tag{2.17}$$

where τ_1 is the longest relaxation time of the polymer molecule which can be expressed as $\tau_1 \sim R_g^2/D$. Using Eqs. (2.13), (2.15) and (2.17), it follows that the scaling law for η_p is

$$\eta_p = \eta_s c b^3 N_b^{3\nu - 1} z^{\star 3(2\nu - 1)} \tag{2.18}$$

Defining $\eta_p^{\theta} = \dfrac{c}{N_b} \, \eta_s \, b^3 \, N_b^{3/2}$ and using Eq. (2.10), the scaling law for the ratio of viscosities in terms of scaled variables is

$$\frac{\eta_p}{\eta_p^{\theta}} = z^{3(2\nu - 1)} \tag{2.19}$$

Using the scaling arguments discussed in this section, and the concepts of excluded volume and hydrodynamics interactions, we develop the scaling theory for finite concentration systems in the next chapter.

Chapter 3

Scaling Theory for Semidilute Polymer Solutions Near Equilibrium

Predicting the behavior of polymer solutions becomes more complicated when the polymer concentration becomes finite. When the polymer concentration increases to such an extent that the monomer concentration c becomes equal to the overlap concentration $c^\star$ (where the total volume is equal to the volume occupied by polymer chains), the polymer solution becomes semidilute. Although c may be low, a strong overlap of polymer chains may occur in semidilute solutions because of the extended size of polymer chains. This chapter attempts to develop a scaling theory for semidilute polymer solutions that is applicable to both the static and dynamic crossovers (based on the framework developed in Chapter 2 in the context of dilute polymer solutions).

Depending on the strength of the excluded volume interaction $z^\star$ and the monomer concentration c, different regimes on the phase diagram of a polymer solution (Rubinstein and Colby, 2003) can be drawn as shown in Fig. 3.1. Dilute

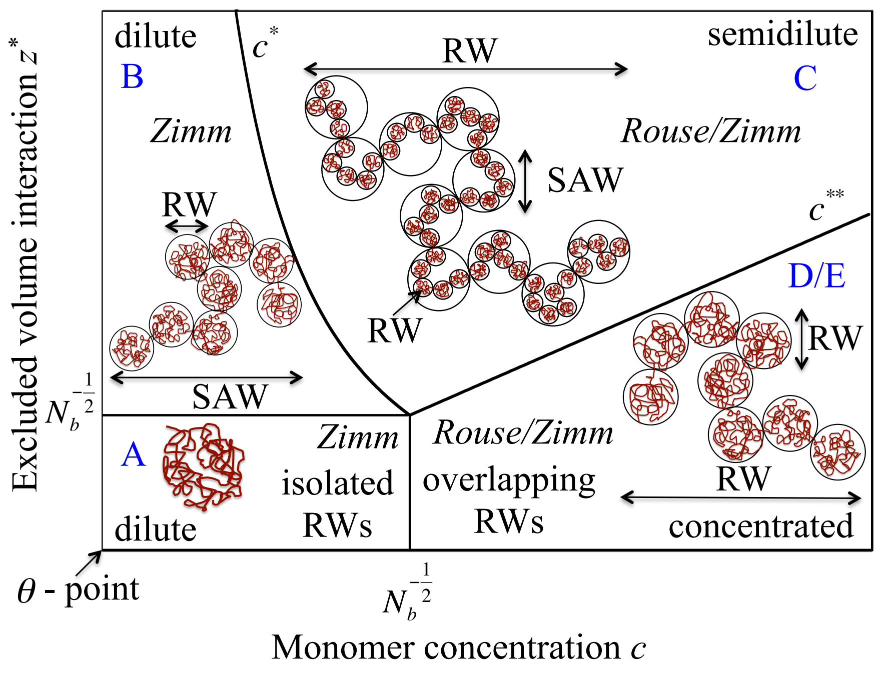


Figure 3.1: Phase diagram of a polymer solution in the plane monomer concentration c and excluded volume strength $z^\star$, as predicted by the standard blob picture. The θ regime occurs for small $z^\star \ll N_b^{-1/2}$, while $z^\star \gg N_b^{-1/2}$ in the good solvent regime. The overlap concentration is given by $c^\star$. For concentrations above $c^{\star\star}$, excluded volume interactions are completely screened.

polymer solutions are indicated by regimes A and B for theta and good solvent regimes, respectively, and the semidilute regime is indicated by regime C. Unlike dilute solutions where excluded volume and hydrodynamics interactions were treated as only intra-chain interactions, in semidilute solutions these interactions are intra-chain as well as inter-chain, which leads to a many-body problem. The challenge of solving the many-body problem becomes even more difficult due to the complicated interplay between excluded volume and hydrodynamic interactions. Using the concepts of scaling blobs, Daoud et al. (1975) developed scaling theories to describe the static and dynamic behavior of semidilute polymer solutions at equilibrium. Scaling theory is based on the idea of a separation of length scales. Apart from the thermal blob (as discussed in Chapter 2), another blob (or another length scale) called the *concentration blob or correlation blob* is introduced to felicitate the development of a scaling theory for semidilute polymer solutions (de Gennes, 1979).

In both static and dynamic scaling theories, the concentration blob size ξ_c corresponds to the length scale at which interactions between polymer chains become significant. On length scales smaller than ξ_c the interaction is not important, and the segment inside a concentration blob essentially follows isolated chain (dilute solution) behavior. Parts of a chain inside the concentration blob are not aware that the solution is semidilute because these interact mainly with solvent molecules and with other monomers from the same segment of the chain. Since the conformations of a polymer chain in a dilute solution in a good solvent follow self-avoiding walk (SAW) statistics, the conformations of parts of a chain that lie within a concentration blob obey SAW statistics of thermal blobs as illustrated in Fig. 3.2. On length scales larger than the concentration blob size, the behavior of a semidilute polymer solution is controlled by interactions between chains. Moreover, since the concentration of polymers in a semidilute solution is such that the concentration blob volumes are space filling, on length scales larger than ξ_c the semidilute solution behaves as a melt of chains whose segments are concentration blobs. Thus, polymer conformations on these scales are random walks (RW) of concentration blobs. This static scaling ansatz is shown schematically in Fig. 3.2. The transition from self-avoiding walk statistics to random walk statistics on the length scale of the concentration blob occurs due to the screening of excluded volume interactions by the overlapping chains. On length scales smaller than ξ_c excluded volume interactions are strong enough to swell

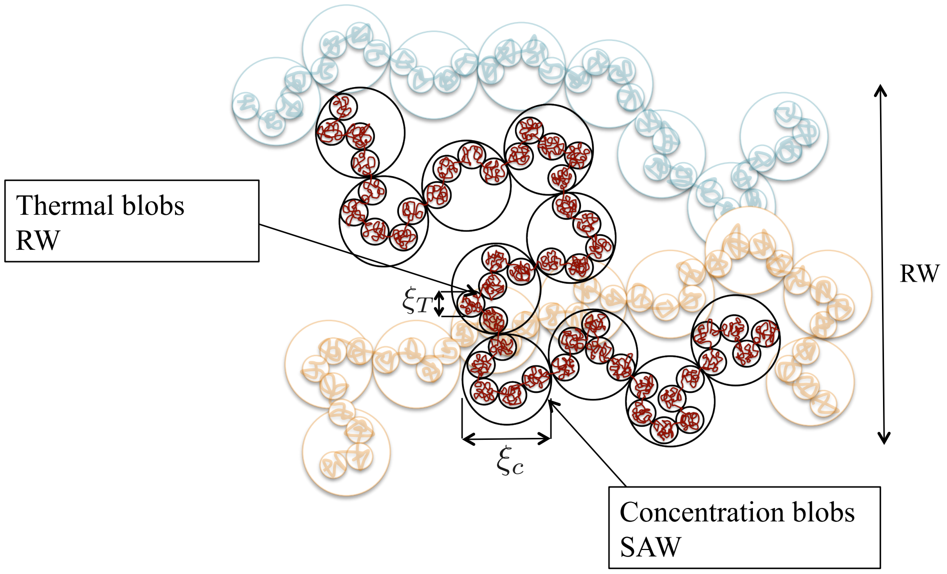


Figure 3.2: Blob picture in a semidilute polymer solution

the chain but are not yet screened by the surrounding chains, while on length scales larger than ξ_c, as in the case of melts, excluded volume interactions are screened.

The blob scaling picture has also been adopted to describe the equilibrium dynamic behavior of semidilute polymer solutions. As discussed in Chapter 2, polymer dynamics in dilute solutions incorporating hydrodynamic interactions, can be described by Zimm dynamics. In concentrated solutions, hydrodynamic interactions are screened (Freed and Edwards, 1974; Shiwa et al., 1988; Richter et al., 1984; Edwards and Muthukumar, 1984; Fredrickson and Helfand, 1990; R. H. Colby et al., 1994) similar to the screening of excluded volume interactions. As a consequence, for short chains that are not entangled, the Rouse model, which neglects hydrodynamic interactions, has been found to provide an accurate description of polymer dynamics in the concentrated regime (Doi and Edwards, 1986). In the special case of semidilute solutions, there exists a length scale ξ_H (which is often called the hydrodynamic screening length), which separates

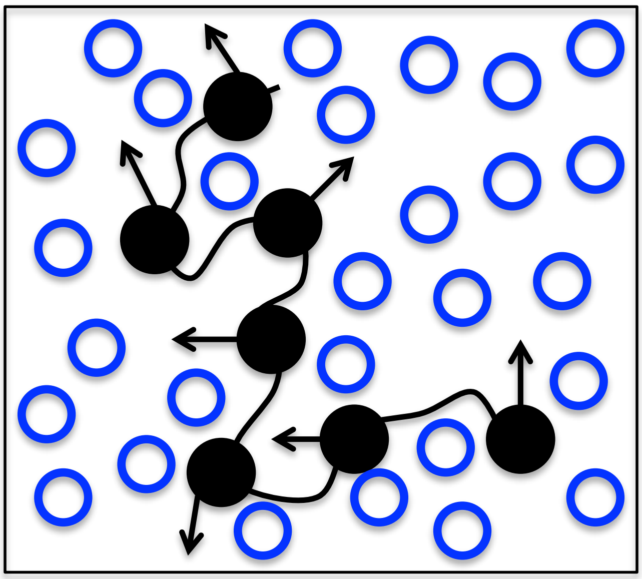


Figure 3.3: Schematic representation of hydrodynamic screening by surrounding chains.

these two types of dynamics. On length scales shorter than ξ_H, hydrodynamic interactions dominate, and the Zimm model describes polymer dynamics. On length scales larger than ξ_H, HI is screened, essentially as a result of chain-chain collisions which tend to randomize the momentum propagation in the system (as shown schematically in Fig. 3.3) (Ahlrichs et al., 2001), and the Rouse model describes polymer dynamics. In the dynamic scaling scenario, as a consequence, the blob size ξ_H represents the length scale at which a crossover occurs from Zimm to Rouse dynamics.

In an important paper, de Gennes (1976b) verified the Freed-Edwards theory (Freed and Edwards, 1974) for the screening of hydrodynamic interactions and set up the connection between static length scale ξ_c and the dynamic length scale or hydrodynamic interaction screening length ξ_H. de Gennes (1976b) proposed convincing scaling arguments to suggest that the static and dynamic correlation lengths should be identical, i.e., $\xi_c = \xi_H$. Experimental results from static and dynamic light scattering studies appear to support de Gennes contention that hydrodynamic interactions and excluded volume interactions are indeed screened on similar length scales (Wiltzius and Cannell, 1986; Zhang et al., 1999).

In another important study, Ahlrichs et al. (2001) have revealed an interesting aspect about the screening of hydrodynamics interactions in semidilute solutions at equilibrium. They found from their simulations (using a coupled Molecular Dynamics/Lattice Boltzmann mesoscopic simulation scheme (Ahlrichs and Dünweg, 1999)) that hydrodynamic screening is not simply a matter of length scales, but must necessarily be viewed as a dynamic time-dependent phenomenon. In particular, they found that HI is not screened up to the *crossover time* τ_c on all length scales, after which screening sets in, leading to Rouse-like motion. The crossover time τ_c is the Zimm time of a concentration blob, i.e., roughly the time needed for a blob to move its own size. After τ_c, the chain will, on average, feel the constraints by the temporary matrix of other chains. From then on it is unable to follow the flow, but rather lags behind, and momentum transport occurs mainly along the chain backbones, due to connectivity forces. The computer simulation results of Ahlrichs et al. (2001) are in complete accord with the neutron spin echo measurements on labeled polymer chains by Richter et al. (1984), which show that hydrodynamic screening is a dynamic effect which becomes relevant only after the crossover time. However, it should be noted that their findings are not relevant if we are interested in understanding only long-time dynamics.

The central message of the discussion above is that for $c/c^\star > 1$, and at sufficiently low time scales, screening of both HI and EV interactions occur at a certain length scale ξ_c (the size of a concentration blob), beyond which polymer chains obey RW statistics. A semidilute solution (regime C in Fig. 3.1) is consequently characterized by $b \ll \xi_T \ll \xi_c \ll R_g$, i.e., by only a finite window of length scales between ξ_T and ξ_c where SAW statistics apply. Upon further increasing c, the system enters the concentrated regime $c \gg c^{\star\star}$, where this window has shrunk to zero and chains obey RW statistics on all length scales. The reason the window shrinks to zero is because, (i) ξ_T is independent of the monomer concentration, as discussed in Chapter 2, and (ii) ξ_c decreases by increasing concentration (as will be discussed shortly), and becomes equal to ξ_T at $c = c^{\star\star}$ (Rubinstein and Colby, 2003). As a result, on length scales smaller than ξ_c, thermal energy is stronger than the energy of excluded volume interactions, and on length scales greater than ξ_c, excluded volume interactions are screened by overlapping chains. Therefore, ideal chain like behavior is observed at all length scales when the monomer concentration is equal to or greater than $c^{\star\star}$ (Rubinstein and Colby, 2003).

The lines drawn in Fig. 3.1 do not indicate sharp phase transitions but rather smooth crossovers. Clearly, there is a concentration driven crossover from the dilute regime (SAW statistics) to the concentrated regime (RW statistics) via the semidilute regime for any given solvent quality. The effect of solvent quality on the behavior of semidilute solutions can be studied in a similar manner as was discussed for dilute solutions in Chapter 2. In order to understand the effect of concentration and solvent quality together on the behavior of semidilute solutions at equilibrium, the blob scaling picture can be applied to regime C, and scaling relations for ξ_c, R_g and D can be derived (in terms of concentration and solvent quality) as discussed below.

Thermal blobs within the concentration blob obey SAW statistics, hence the size of a concentration blob ξ_c is given by

$$\xi_c = \xi_T \, m^{\nu} \tag{3.1}$$

where, m is the number of thermal blobs within the concentration blob. The size ξ_c is found by the knowledge that the solution is homogeneous on length scales larger than ξ_c, and that the concentration blobs are space filling. The total number of monomers within ξ_c is nm, therefore

$$nm = c\,\xi_c^3 \tag{3.2}$$

Using Eqs. (2.8), (2.9), and (3.1), it is straight forward to show that

$$m = \left(z^\star c^{-1} b^{-3}\right)^{1/(3\nu-1)} \tag{3.3}$$

As a result, using Eqs. (2.9) and (3.3) in Eq. (3.1)

$$\xi_c = b\,(z^\star)^{-\frac{2\nu-1}{3\nu-1}} \left(cb^3\right)^{-\frac{\nu}{3\nu-1}} \tag{3.4}$$

As seen in Chapter 2, for dilute solutions, the universal crossover scaling relationships for different properties can be expressed in terms of the scaled solvent quality parameter z. It is convenient to study the concentration driven crossover in terms of the ratio of monomer concentration c to the overlap concentration $c^\star$. It turns out (as will be seen subsequently) that the scaled variables $c/c^\star$ and z completely absorb the dependence of N_b, and the phase diagram can be expressed in terms of z and $c/c^\star$ as shown in Fig. 3.4.

In order to represent the size of the concentration blob in terms of scaled variables, it is necessary to derive expression for the overlap concentration $c^\star$, which separates regimes B and C in the phase diagram. The overlap concentration can be found from the condition that at $c = c^\star$, the concentration blob contains just all monomers of the chain, N_b. From Eqs. (2.9), (3.1) and (3.2), since $nm = N_b$

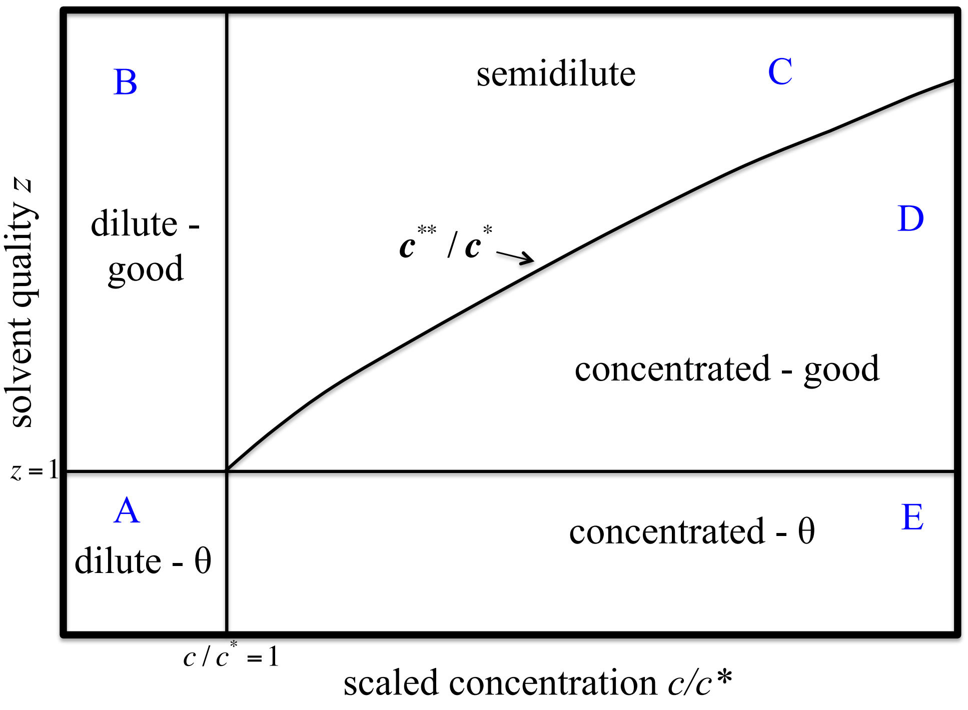


Figure 3.4: Same as Fig. 3.1 but now using the scaled concentration $c/c^{\star}$ and the solvent quality $z = z^{\star} N_b^{1/2}$ as variables. Note that in this representation the loci of the crossovers do not depend on the chain length N_b.

at $c = c^{\star}$

$$N_b = c^{\star} b^3 z^{\star -3} m^{3\nu} \tag{3.5}$$

Substituting for m from Eq. (3.3), and solving for $c^{\star}$, leads to

$$c^{\star} = b^{-3} N_b^{-(3\nu-1)} z^{\star -3(2\nu-1)} \tag{3.6}$$

When solved for $z^{\star}$, this leads to the equation of the line that separates regimes B and C shown in Fig. 3.1. It is convenient to represent cb^3 as $(c/c^{\star})(c^{\star}b^3)$ when converting from unscaled to scaled variables. One can then show from Eqs. (2.10), (3.4) and (3.6) that the dimensionless size of the concentration blob in terms of scaled variables is

$$\frac{\xi_c}{R_g^{0,\theta}} = \left(\frac{c}{c^\star}\right)^{-\nu/(3\nu-1)} z^{2\nu-1} \tag{3.7}$$

where, $R_g^{0,\theta} = bN_b^{1/2}$ is the gyration radius in dilute solution at θ condition. We can also derive an expression for ξ_c in the concentrated regime. Upon systematically increasing the monomer concentration c, the concentration blob shrinks until at a threshold concentration $c = c^{\star\star}$, $\xi_c = \xi_T$, and there is no longer any length scale where SAW statistics apply. The locus of points $c^{\star\star}$ separates regime C from regimes D and E in the unscaled variables phase diagram. It is straight forward to calculate $c^{\star\star}$ by equating Eqs. (2.9) and (3.4)

$$c^{\star\star} = b^{-3}\, z^\star \tag{3.8}$$

In terms of scaled variables, it is similarly straight forward to show by equating Eqs. (2.11) and (3.7) that

$$c^{\star\star} = c^\star\, z^{2(3\nu-1)} \tag{3.9}$$

As a result,

$$\frac{c^{\star\star}}{c^\star} = z^{2(3\nu-1)} \tag{3.10}$$

It is clear that when represented in terms of z and $c/c^\star$, the loci of crossover between the various regimes no longer depend on chain length N_b. In regimes D and E, where only RW statistics apply, the concentration blob size is given by

$$\xi_c = bg^{1/2} \tag{3.11}$$

where, g is the number of monomers in a concentration blob. Since the solution is homogeneous on all length scales larger than ξ_c, and the concentration blobs are space filling,

$$c = \frac{g}{\xi_c^3} = \frac{g}{(bg^{1/2})^3} \tag{3.12}$$

which implies

$$g = (cb^3)^{-2} \tag{3.13}$$

and as a result

$$\xi_c = b\,(cb^3)^{-1} \tag{3.14}$$

The expression for $c^\star$ given by Eq. (3.6) is only valid away from the Θ regime, *i.e.*, for $z^\star > N_b^{-1/2}$. On the other hand, in the Θ regime, since at $c = c^\star$ the concentration blob contains all the monomers in the chain, which implies $\xi_c = bN_b^{1/2}$, Eq. (3.14) leads to

$$c^\star = b^{-3}\,N_b^{-1/2} \tag{3.15}$$

This difference in the expressions for $c^\star$ makes it necessary to distinguish between regimes D (where EV effects are important) and E (where EV effects are negligible) when representing the size of the concentration blob (and indeed all other observables) in terms of scaled variables. Figure 3.4 shows the clear demarcation between regime D and E. In regime D, Eqs. (2.10), (3.6) and (3.14) imply

$$\frac{\xi_c}{R_g^{0,\theta}} = \left(\frac{c}{c^\star}\right)^{-1} z^{3(2\nu-1)} \tag{3.16}$$

while in regime E, Eqs. (3.14) and (3.15) imply

$$\frac{\xi_c}{R_g^{0,\theta}} = \left(\frac{c}{c^\star}\right)^{-1} \tag{3.17}$$

Once the blob scaling laws for ξ_T are known in regimes B and C, and those for ξ_c are known in regimes C to E, the scaling laws for all other observables in these regimes can be derived. Even though these laws have been derived and discussed in detail previously (de Gennes, 1979; Grosberg and Khokhlov, 1994; Rubinstein and Colby, 2003), here, as examples, we derive the scaling laws for the static chain size R_g, and the single-chain diffusion coefficient D, in order to represent them in terms of the notation used in this work. Also, as discussed in Chapter 2, knowing the expressions for R_g and D, the scaling laws for $\eta_{\rm p}$, the polymer contribution to viscosity, can be derived using Eq. (2.17). The scaling laws for regime A are not discussed since the conventional notation in this regime is followed here, and regime B has been discussed in Chapter 2.

Regime C

Since RW statistics are obeyed on length scales larger than ξ_c

$$R_g = \xi_c \left(\frac{N_b}{nm} \right)^{1/2} \tag{3.18}$$

where, (N_b/nm) is the number of concentration blobs in a chain. Using Eqs. (2.8), (3.3) and (3.4), leads to

$$R_g = b N_b^{1/2} \left[z^\star (cb^3)^{-1} \right]^{\frac{1}{2}\frac{2\nu-1}{3\nu-1}} \tag{3.19}$$

The scaling relation in terms of scaled variables is obtained by using Eqs. (2.10) and (3.6)

$$\frac{R_g}{R_g^{0,\theta}} = \left(\frac{c}{c^\star} \right)^{-\frac{1}{2}\frac{2\nu-1}{3\nu-1}} z^{2\nu-1} \tag{3.20}$$

Rouse dynamics are obeyed on length scales larger than ξ_c. Therefore, the total friction of the chain is the sum of the friction experienced by all the concentration blobs. This leads to the following expression for the diffusivity

$$D = \frac{k_B T}{\eta_{\rm s}\, \xi_c \, (N_b/nm)} \tag{3.21}$$

Using Eqs. (2.8), (3.3) and (3.4), leads to

$$D = \frac{k_B T}{\eta_s b N_b} (z^\star)^{-2\frac{2\nu-1}{3\nu-1}} (cb^3)^{-\frac{1-\nu}{3\nu-1}}. \tag{3.22}$$

The nondimensional diffusivity in terms of scaled variables follows from using Eqs. (2.10) and (3.6) in Eq. (3.22)

$$\frac{D}{D_0^\theta} = (c/c^\star)^{-\frac{1-\nu}{3\nu-1}}\, z^{-(2\nu-1)} \tag{3.23}$$

where, $D_0^\theta = (k_B T/\eta_s b N_b^{1/2})$ is the diffusivity in the dilute solution at θ condition. The polymer contribution to the viscosity η_p $\left(k_B T \dfrac{c}{N_b} \dfrac{R_g^2}{D}\right)$ can be written as

$$\eta_p = \eta_s (cb^3)^{\frac{1}{3\nu-1}} N_b (z^\star)^{3\frac{2\nu-1}{3\nu-1}} \tag{3.24}$$

Using Eq. (3.6) and (3.24), the polymer contribution to the viscosity η_p can be expressed in terms of scaled variables as,

$$\eta_p/\eta_s = (c/c^\star)^{\frac{1}{3\nu-1}} \tag{3.25}$$

Regime D

In regime D, RW statistics are obeyed on all length scales, both within the concentration blob, and by the concentration blobs themselves. As a result

$$R_g = \xi_c \left(\frac{N_b}{g}\right)^{1/2} = b N_b^{1/2} \tag{3.26}$$

where Eqs. (3.13) and (3.14) have been used for simplification. The nondimensional chain size in this regime is trivially $R_g/R_g^{0,\theta} = 1$.

Rouse dynamics are obeyed on length scales larger than ξ_c. Using Eq. (3.21) (with $g = nm$), and Eqs. (3.13) and (3.14) leads to

$$D = \frac{k_B T}{\eta_s b N_b} (c b^3)^{-1} \tag{3.27}$$

The nondimensional diffusivity in terms of scaled variables follows from Eqs. (2.10) and (3.6)

$$\frac{D}{D_0^\theta} = \left(\frac{c}{c^\star}\right)^{-1} z^{3(2\nu-1)} \tag{3.28}$$

Regime E

In regime E, the scaling relations for R_g and D in terms of unscaled variables are identical to those in regime D, since in both regimes, RW statistics are obeyed on all length scales, and Rouse dynamics apply on length scales larger than ξ_c. In terms of scaled variables, while the distinction between the two regimes is irrelevant for the static chain size, the difference in the expression for $c^\star$ due to the presence and absence of EV effects, respectively, manifests itself as a difference in the expression for the nondimensional diffusivity. Using Eqs. (2.10) and (3.15), Eq. (3.27) reduces in terms of scaled variables to

$$\frac{D}{D_0^\theta} = \left(\frac{c}{c^\star}\right)^{-1} \tag{3.29}$$

Table 3.1 summarizes all the unscaled equations derived in this chapter, including those for the polymer contribution to viscosity. Table 3.2 summarizes the equations in terms of scaled variables. As can be seen, in this representation the dependence on the chain length N_b has been completely absorbed in the N_b dependence of $c^\star$ and z, and the only remaining variables are $c/c^\star$ and z. It is therefore natural to generalize these results to the relations

$$\frac{R_g}{R_g^\star} = \phi_R\left(\frac{c}{c^\star}, z\right) \tag{3.30}$$

$$\frac{D}{D^\star} = \phi_D\left(\frac{c}{c^\star}, z\right) \tag{3.31}$$

and

$$\frac{\eta_\mathrm{p}}{\eta_\mathrm{p}^\star} = \phi_\eta\left(\frac{c}{c^\star}, z\right) \tag{3.32}$$

where, $R_g^\star$, $D^\star$, $\eta_\mathrm{p}^\star$ denote the values of R_g, D, η_p at $c = c^\star$, and ϕ_R, ϕ_D, and ϕ_η are universal crossover scaling functions defined on the whole plane of Fig. 3.4, and which, up to numerical prefactors, are known deep in the asymptotic regimes (these are just the results listed in Table 3.2), but whose behavior needs to be calculated or measured near the crossover lines.

Finally, an interesting observation can be made when looking at the last two lines of Table 3.2, i.e.,

$$(c/c^\star)^{1/4}\left(R_g/R_g^\star\right)(D/D^\star)^{1/4} = \begin{cases} \left(\dfrac{c}{c^\star}\right)^{1/4} & \text{in regimes A and B} \\ 1 & \text{in regimes C, D and E} \end{cases} \tag{3.33}$$

and

$$(c/c^\star)^{1/3}\left(R_g/R_g^\star\right)\left(\eta_p/\eta_p^\star\right)^{-1/6} = \begin{cases} \left(\dfrac{c}{c^\star}\right)^{1/6} & \text{in regimes A and B} \\ 1 & \text{in regimes C, D and E} \end{cases} \tag{3.34}$$

In all regimes above the overlap concentration, we find that the functions $R_g/R_g^\star$, $D/D^\star$, and $\eta_\mathrm{p}/\eta_\mathrm{p}^\star$ are not independent, but can be calculated as soon as one of them is known. It is then natural to assume that the same property also holds in the crossover regimes, i.e. that the relations

$$(c/c^\star)^{1/4}\phi_R\phi_D^{1/4} = \left(\frac{c}{c^\star}\right)^{1/4} \tag{3.35}$$

and

$$(c/c^\star)^{1/3}\phi_R\phi_\eta^{-1/6} = \left(\frac{c}{c^\star}\right)^{1/6} \tag{3.36}$$

are valid in regimes A and B, while the relations

$$(c/c^\star)^{1/4}\phi_R\phi_D^{1/4} = \text{const.} \tag{3.37}$$

and

$$(c/c^\star)^{1/3}\phi_R\phi_\eta^{-1/6} = \text{const.} \tag{3.38}$$

are valid in regimes C, D and E.

By carrying out a scaling analysis of the semidilute regime has led to the following predictions:

1. There exist universal scaling functions ϕ_R, ϕ_D and ϕ_η that depend only on $(c/c^\star)$ and z.

2. These functions obey power laws whose specific forms are known away from the crossover regions, but scaling theory is unable to provide further information in the crossover regimes.

3. Combinations of the scaling functions assume particularly simple forms that span several regimes in the phase diagram, suggesting that they are valid even in the crossover regimes.

4. Equations (3.35) - (3.38) suggest that there is only a single crossover scaling function and the others can be inferred from them.

These predictions of scaling theory can be verified by carrying out BD simulations. Further, the specific forms of ϕ_R, ϕ_D and ϕ_η in the crossover regimes can also be ascertained . This is a substantial part of the goals of this thesis. In Chapters 4 and 5 we develop the BD algorithm and in Chapter 6 we verify the prediction of scaling theory.

regime	A	B	C	D / E
ξ_T	–	$b\,(z^\star)^{-1}$	$b\,(z^\star)^{-1}$	–
ξ_c	–	–	$b\,(z^\star)^{-\frac{2\nu-1}{3\nu-1}}(cb^3)^{-\frac{\nu}{3\nu-1}}$	$b\,(cb^3)^{-1}$
R_g	$bN_b^{1/2}$	$bN_b^{\nu}(z^\star)^{2\nu-1}$	$bN_b^{1/2}\left[z^\star(cb^3)^{-1}\right]^{\frac{1}{2}\frac{2\nu-1}{3\nu-1}}$	$bN_b^{1/2}$
D	$\dfrac{k_BT}{\eta_s bN_b^{1/2}}$	$\dfrac{k_BT}{\eta_s bN_b^{\nu}(z^\star)^{2\nu-1}}$	$\dfrac{k_BT}{\eta_s bN_b}(z^\star)^{-2\frac{2\nu-1}{3\nu-1}}(cb^3)^{-\frac{1-\nu}{3\nu-1}}$	$\dfrac{k_BT}{\eta_s bN_b}(cb^3)^{-1}$
η_p	$\eta_s cb^3 N_b^{1/2}$	$\eta_s cb^3 N_b^{3\nu-1}(z^\star)^{3(2\nu-1)}$	$\eta_s(cb^3)^{\frac{1}{3\nu-1}}N_b(z^\star)^{3\frac{2\nu-1}{3\nu-1}}$	$\eta_s(cb^3)^2N_b$

Table 3.1: Various quantities of the polymer system (thermal blob size ξ_T, overlap blob size ξ_c, gyration radius R_g, single-chain diffusion constant D, polymer part of the viscosity η_p) in the regimes indicated in Fig. 3.1, as a function of monomer size b, chain length N_b, excluded volume interaction strength $z^\star$, thermal energy k_BT, solvent viscosity η_s, and monomer concentration c, within the framework of scaling theory. Blob sizes are not indicated in cases where they are irrelevant. Numerical prefactors of order unity have been ignored. The scaling laws are valid in the asymptotic regimes sufficiently far away from the crossover boundaries.

regime	A	B	C	D	E
$\xi_T/R_g^{0,\theta}$	–	z^{-1}	z^{-1}	–	–
$\xi_c/R_g^{0,\theta}$	–	–	$(c/c^\star)^{-\frac{\nu}{3\nu-1}}\, z^{2\nu-1}$	$(c/c^\star)^{-1}\, z^{3(2\nu-1)}$	$(c/c^\star)^{-1}$
$R_g/R_g^{0,\theta}$	1	$z^{2\nu-1}$	$(c/c^\star)^{-\frac{1}{2}\frac{2\nu-1}{3\nu-1}}\, z^{2\nu-1}$	1	1
D/D_0^θ	1	$z^{-(2\nu-1)}$	$(c/c^\star)^{-\frac{1-\nu}{3\nu-1}}\, z^{-(2\nu-1)}$	$(c/c^\star)^{-1}\, z^{3(2\nu-1)}$	$(c/c^\star)^{-1}$
η_p/η_s	$c/c^\star$	$c/c^\star$	$(c/c^\star)^{\frac{1}{3\nu-1}}$	$(c/c^\star)^2\, z^{-6(2\nu-1)}$	$(c/c^\star)^2$
$R_g/R_g^\star$	1	1	$(c/c^\star)^{-\frac{1}{2}\frac{2\nu-1}{3\nu-1}}$	$z^{-(2\nu-1)}$	1
$D/D^\star$	1	1	$(c/c^\star)^{-\frac{1-\nu}{3\nu-1}}$	$(c/c^\star)^{-1}\, z^{4(2\nu-1)}$	$(c/c^\star)^{-1}$
$\eta_p/\eta_p^\star$	$c/c^\star$	$c/c^\star$	$(c/c^\star)^{\frac{1}{3\nu-1}}$	$(c/c^\star)^2\, z^{-6(2\nu-1)}$	$(c/c^\star)^2$
$(c/c^\star)^{1/4}\left(R_g/R_g^\star\right)(D/D^\star)^{1/4}$	$(c/c^\star)^{1/4}$	$(c/c^\star)^{1/4}$	1	1	1
$(c/c^\star)^{1/3}\left(R_g/R_g^\star\right)\left(\eta_p/\eta_p^\star\right)^{-1/6}$	$(c/c^\star)^{1/6}$	$(c/c^\star)^{1/6}$	1	1	1

Table 3.2: Various normalized quantities of the polymer system (thermal blob size ξ_T, overlap blob size ξ_c, gyration radius R_g, single-chain diffusion constant D, polymer part of the viscosity η_p) in the regimes indicated in Fig. 3.4, in terms of the scaled concentration $c/c^\star$ and the solvent quality $z = z^\star N_b^{1/2}$. Blob sizes are not indicated in cases where they are irrelevant. Numerical prefactors of order unity have been ignored. The scaling laws are valid in the asymptotic regimes sufficiently far away from the crossover boundaries. $R_g^\star$, $D^\star$, $\eta_p^\star$ denote the values of R_g, D, η_p at $c = c^\star$.

Chapter 4

Governing Equations for the Bead-Spring Chain Model

In this chapter, the polymer model and the basic governing equations of the Brownian dynamics simulation algorithm are discussed in the context of a multi-chain system. As mentioned earlier in Chapter 2, excluded volume and hydrodynamic interactions are two important microscopic phenomena that must be incorporated in order to obtain a realistic model for semidilute solution behavior. We briefly discuss the treatment of these two interactions in the context of BD simulations in this chapter. Finally, formulae to calculate the mean values of various equilibrium and rheological properties are presented.

4.1 The Polymer Model

A linear bead-spring chain model (Bird et al., 1987) is used to represent polymers at the mesoscopic level, with each polymer chain coarse-grained into a sequence of N_b beads, which act as centers of hydrodynamic resistance, connected by $N_b - 1$ massless springs that represent the entropic force between adjacent beads.

A semidilute polymer solution is modeled as an ensemble of such bead-spring chains, immersed in an incompressible Newtonian solvent. A total of N_c chains are initially enclosed in a cubic and periodic cell of edge length L, giving a total of $N = N_b \times N_c$ beads per cell at a bulk monomer concentration of $c = N/V$, where $V = L^3$ is the volume of the simulation cell. Figure 4.1 schematically shows a simple example of the simulation system in 2-D.

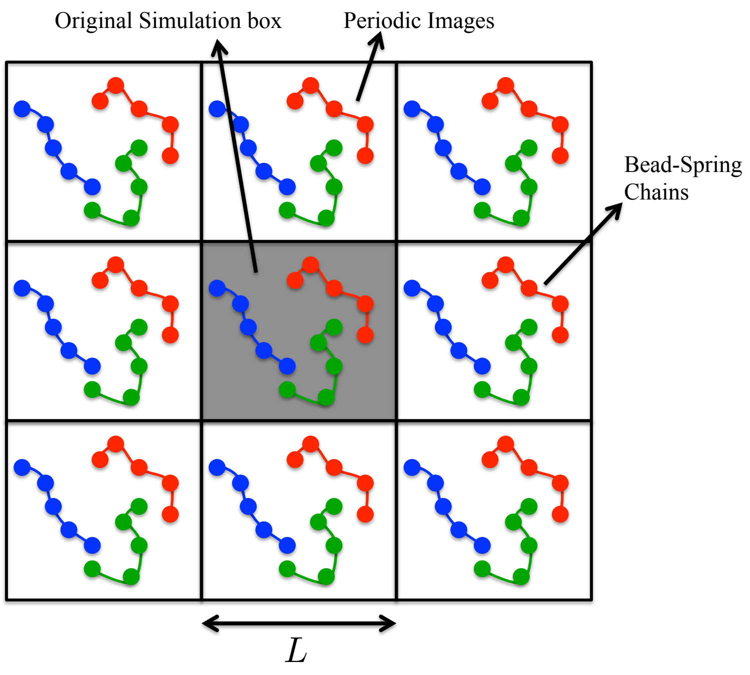


Figure 4.1: A schematic illustration of the simulation system in 2-D, showing an example of a system with box size L, number of chains $N_c = 3$ and number of beads in a chain $N_b = 5$. The grey box indicates the original simulation box while the white boxes are the periodic images.

4.2 The Brownian Dynamics Integrator

Using the length scale $l_H = \sqrt{k_B T/H}$ and time scale $\lambda_H = \zeta/4H$, where k_B is the Boltzmann's constant, T is the temperature, H is the spring constant and ζ is the hydrodynamic friction coefficient associated with a bead, the Euler integration algorithm for the nondimensional Ito stochastic differential equation governing the position vector $\mathbf{r}_\nu(t)$ of bead ν at time t, is (Stoltz et al., 2006)

$$\begin{aligned}\mathbf{r}_\nu(t+\Delta t) = \mathbf{r}_\nu(t) + [\boldsymbol{\kappa}\cdot\mathbf{r}_\nu(t)] + \frac{\Delta t}{4}\sum_{\mu=1}^{N}\left[\mathbf{D}_{\nu\mu}(t)\cdot\mathbf{F}_\mu(t)\right] \\ + \frac{1}{\sqrt{2}}\sum_{\mu=1}^{N}\left[\mathbf{B}_{\nu\mu}(t)\cdot\boldsymbol{\Delta}\mathbf{W}_\mu(t)\right]\end{aligned} \tag{4.1}$$

Here, the 3×3 tensor $\boldsymbol{\kappa}$ is equal to $(\boldsymbol{\nabla}\boldsymbol{v})^T$, with $\boldsymbol{v}$ being the unperturbed solvent velocity. The dimensionless diffusion tensor $\mathbf{D}_{\nu\mu}$ is a 3×3 matrix for a fixed pair of beads μ and ν. It is related to the hydrodynamic interaction tensor, as discussed further subsequently. $\mathbf{F}_\mu$ incorporates all the non-hydrodynamic forces on bead μ due to all the other beads. The non-hydrodynamic forces in the model are comprised of the spring forces $\mathbf{F}_\mu^{\text{spr}}$ and excluded volume interaction forces $\mathbf{F}_\mu^{\text{exv}}$, i.e., $\mathbf{F}_\mu = \mathbf{F}_\mu^{\text{spr}} + \mathbf{F}_\mu^{\text{exv}}$. The components of the Gaussian noise $\boldsymbol{\Delta}\mathbf{W}_\mu$ are obtained from a real-valued Gaussian distribution with zero mean and variance Δt. The quantity $\mathbf{B}_{\nu\mu}$ is a nondimensional tensor whose presence leads to multiplicative noise (Ottinger, 1996). Its evaluation requires the decomposition of the diffusion tensor. Defining the matrices $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{B}}$ as block matrices consisting of $N\times N$ blocks each having dimensions of 3×3, with the (ν,μ)-th block of $\boldsymbol{\mathcal{D}}$ containing the components of the diffusion tensor $\mathbf{D}_{\nu\mu}$, and the corresponding block of $\boldsymbol{\mathcal{B}}$ being equal to $\mathbf{B}_{\nu\mu}$, the decomposition rule for obtaining $\boldsymbol{\mathcal{B}}$ can be expressed as

$$\boldsymbol{\mathcal{B}}\cdot\boldsymbol{\mathcal{B}}^{\mathrm{T}} = \boldsymbol{\mathcal{D}} \tag{4.2}$$

In Eq. (4.1), there are three terms that are challenging to implement efficiently: (i) the flow term $[\boldsymbol{\kappa}\cdot\mathbf{r}_\nu(t)]$, (ii) the drift term $\sum_{\mu=1}^{N}\left[\mathbf{D}_{\nu\mu}(t)\cdot\mathbf{F}_\mu(t)\right]$, and (iii) the diffusion term $\sum_{\mu=1}^{N} [\mathbf{B}_{\nu\mu}(t) \cdot \mathbf{\Delta W}_{\mu}(t)]$. The implementation and optimization of the later two terms are presented in Chapter 5, where as the implementation of the flow term is discussed in Chapter 7. Note that since Chapters 5 and 6 consider only equilibrium systems, the flow term can be ignored for the purpose of the problems discussed in these two chapters.

The specification of the force term in Eq. (4.1) requires the consideration of bonded and non-bonded interactions between beads. These interactions are discussed below.

4.3 Bonded Interactions

Interactions that arise due to the presence of spring forces are labelled here as bonded interactions. In order to model spring forces, a linear Hookean spring potential is used in this thesis for nearly all our studies, except for two cases. The first case involves validation studies where our results are compared with the Lattice Boltzmann method. In this particular study, a finitely extensible nonlinear elastic (FENE) potential has been used to model spring forces. The second case involves the simulation of polymer solutions undergoing either planar elongational flow or planar mixed flow, and FENE springs are used in this case as well. The entropic spring force on bead μ due to adjacent beads can be expressed as $\mathbf{F}_{\mu}^{\text{spr}} = \mathbf{F}^{c}(\mathbf{Q}_{\mu}) - \mathbf{F}^{c}(\mathbf{Q}_{\mu-1})$ where $\mathbf{F}^{c}(\mathbf{Q}_{\mu-1})$ is the force between the beads $\mu - 1$ and μ, acting in the direction of the connector vector between the two beads $\mathbf{Q}_{\mu-1} = \mathbf{r}_{\mu} - \mathbf{r}_{\mu-1}$. The dimensionless Hookean spring force is given by $\mathbf{F}^{c}(\mathbf{Q}_{\mu}) = \mathbf{Q}_{\mu}$, while for FENE springs, $\mathbf{F}^{c}(\mathbf{Q}_{\mu}) = \dfrac{\mathbf{Q}_{\mu}}{1 - |\mathbf{Q}_{\mu}|^{2}/b_{p}}$, where $b_{p} = Hq_{0}^{2}/k_{B}T$ is the dimensionless finite extensibility parameter, and q_{0} is the dimensional maximum stretch of a spring.

4.4 Non-bonded Interactions

In the present instance, where we are considering the dynamics of neutral polymer solutions, the two non-bonded interactions are hydrodynamic and excluded volume interactions, respectively. Hydrodynamic interactions are accounted for through the diffusion tensor (Prabhakar and Prakash, 2004) and as mentioned earlier, excluded volume interactions are modeled using a narrow Gaussian potential (Prakash and Öttinger, 1999).

The nondimensional diffusion tensor $\mathsf{D}_{\nu\mu}$ in Eq. (4.1) is related to the nondimensional hydrodynamic interaction tensor $\boldsymbol{\Omega}$ through

$$\mathsf{D}_{\mu\nu} = \delta_{\mu\nu}\,\boldsymbol{\delta} + (1-\delta_{\mu\nu})\,\boldsymbol{\Omega}(\mathbf{r}_{\mu\nu}) \tag{4.3}$$

where $\boldsymbol{\delta}$ and $\delta_{\mu\nu}$ represent a unit tensor and a Kronecker delta, respectively, while $\boldsymbol{\Omega}$ represents the effect of the motion of a bead μ on another bead ν through the disturbances carried by the surrounding fluid. In above equation, the distance vector $\mathbf{r}_{\mu\nu}$ connecting the beads μ and ν is an abbreviation for $\mathbf{r}_{\nu} - \mathbf{r}_{\mu}$. The hydrodynamic interaction tensor $\boldsymbol{\Omega}$ is assumed to be given by the Rotne-Prager-Yamakawa (RPY) regularization of the Oseen function

$$\boldsymbol{\Omega}(\mathbf{r}) = \Omega_1\,\boldsymbol{\delta} + \Omega_2\frac{\mathbf{rr}}{\mathbf{r}^2} \tag{4.4}$$

where for $r \geq 2a$, the *branch* $\mathcal{A}$ of the RPY functions Ω_1 and Ω_2 is given by, respectively,

$$\Omega_1 = \frac{3a}{4r}\left(1+\frac{2a^2}{3r^2}\right) \quad \text{and} \quad \Omega_2 = \frac{3a}{4r}\left(1-\frac{2a^2}{r^2}\right) \tag{4.5}$$

while for $0 < r \leq 2a$, the *branch* $\mathcal{B}$ of the RPY functions Ω_1 and Ω_2 is given by, respectively,

$$\varOmega_1 = 1 - \frac{9}{32}\frac{r}{a} \quad \text{and} \quad \varOmega_2 = \frac{3}{32}\frac{r}{a} \tag{4.6}$$

We introduce the notation of the two branches $\mathcal{A}$ and $\mathcal{B}$ for facilitating subsequent discussion in Section 5.2.2. The quantity a has been introduced here as the nondimensional radius of the bead as an additional independent parameter. It is related to the conventionally defined (Thurston and Peterlin, 1967; Bird et al., 1987) hydrodynamic interaction parameter h^* by $a = \sqrt{\pi}h^*$. As is well known (Beenakker, 1986), the sum $\sum_\mu \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_\mu$ in Eq. (4.1) converges slowly since $\mathsf{D}_{\mu\nu}$ is long-ranged in nature, scaling as $1/r$. The problem of slow convergence can be resolved through the use of the Ewald sum, as discussed in greater detail in Section 5.2. It is worth noting here that it is sufficient to evaluate $\sum_\mu \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_\mu$ in order to determine the time evolution of $\mathbf{r}_\nu(t)$. It is not necessary to know $\mathsf{D}_{\nu\mu}$ explicitly. Further, as will be seen later in Section 5.5, the evaluation of $\sum_\mu \mathsf{B}_{\nu\mu} \cdot \boldsymbol{\Delta}\mathbf{W}_\mu$ using a Chebyshev polynomial approximation for $\mathsf{B}_{\nu\mu}$, also requires a repeated evaluation of the Ewald sum.

As in the case of dilute solutions, we treat EV interactions in semidilute polymer solutions as well with a narrow Gaussian potential which in terms of non-dimensional variables is given by

$$E\left(\mathbf{r}_{\nu\mu}\right) = z^\star \left(\frac{1}{d^{\star 3}}\right) \exp\left\{-\frac{1}{2}\frac{\mathbf{r}_{\nu\mu}^2}{d^{\star 2}}\right\} \tag{4.7}$$

As discussed in Section 2.1, the dimensionless parameter $z^\star$ is the strength of excluded volume interactions and $d^\star$ is a dimensionless parameter that measures the range of the excluded volume interaction. $z^\star$ is related to the solvent quality parameter through $z^\star = z/\sqrt{N_b}$ (see Section 2.1), and $d^\star$ is related to $z^\star$ through $d^\star = K z^{\star 1/5}$, where K is an arbitrary parameter which becomes irrelevant in the long chain limit (Kumar and Prakash, 2003). In the absence of excluded volume interactions (when the strength of the potential $z^\star$ is zero), polymer chains behave like ideal chains. As $z^\star$ increases, the solvent quality increases and hence polymer chains tend to swell.

4.5 Macroscopic Properties

Static and dynamic properties of semidilute polymer solutions at equilibrium can be calculated once the trajectories of the time evolution of all the beads on all the chains are obtained using Eq. (4.1). For rheological properties, not only are bead configurations, but also the forces on them are required. As the simulation progresses in time, bead configurations and forces are stored simultaneously on the fly. The following sections briefly describe the expressions used to calculate the various physical properties considered in this thesis.

4.5.1 Equilibrium properties

Equilibrium physical properties that are independent of time are called static properties. There are three important static properties for polymer molecules: (i) the end-to-end distance, (ii) the gyration radius and (iii) the static structure factor. The mean dimension or the size of a polymer chain is assessed through the end-to-end distance and the gyration radius (Doi and Edwards, 1986; Rubinstein and Colby, 2003). The end-to-end distance is defined as the mean square distance between the first and the last beads on a chain,

$$\left\langle R_e^2 \right\rangle = \left\langle (\mathbf{r}_{N_b} - \mathbf{r}_0)^2 \right\rangle \tag{4.8}$$

where, $\langle \cdots \rangle$ represents an ensemble average, and $\mathbf{r}_{N_b}$ and $\mathbf{r}_0$ are position vectors of the first and the last bead, respectively. The mean square gyration radius, which is the mean square distance between the beads and the center of mass $\mathbf{r}_{\text{cm}}$ of the chain, written as $\mathbf{r}_{\text{cm}} = \dfrac{1}{N_b+1}\sum_{\mu=0}^{N_b}\mathbf{r}_\mu$, is defined by

$$\left\langle R_g^2 \right\rangle = \frac{1}{N_b+1}\sum_{\mu=0}^{N_b}\left\langle (\mathbf{r}_\mu - \mathbf{r}_{\text{cm}})^2 \right\rangle \tag{4.9}$$

The static structure factor or form factor is a convenient quantity that describes the structure of the chains at all length scales (Binder, 1995). It is important to calculate structure factors not only to quantify the structure of a polymer but also to check with experimental results because polymer structures are mainly determined by experimental means. The static structure factor of a polymer chain is given by (Doi and Edwards, 1986)

$$S(k) = \frac{1}{N_b}\sum_{\mu\nu}\langle \exp\left(i\mathbf{k}\cdot\mathbf{r}_{\mu\nu}\right)\rangle = \frac{1}{N_b}\sum_{\mu\nu}\left\langle \frac{\sin\left(kr_{\mu\nu}\right)}{kr_{\mu\nu}} \right\rangle \tag{4.10}$$

where $\mathbf{k}$ is the scattering vector and k is the magnitude of $\mathbf{k}$, and $r_{\mu\nu}$ is the magnitude of $\mathbf{r}_{\mu\nu}$.

Equations (4.8) to (4.10) refer to the static properties of a single chain at infinite dilution. For a multi-chain system, these static properties are calculated for each individual chain and averaged across all the chains in the system.

The output of a Brownian dynamics simulation is basically the time evolution of the configuration of all the chains in the system. The mean values and the standard error are obtained in the post-processing stage. First, at a given time, the values of the static properties of each chain are calculated, and then these values are averaged across all the chains to obtain a typical average property which we denote as S_{chains}. S_{chains} is then calculated at all the times and the time series $S_{\text{chains}}(t)$ is obtained. In order to estimate the mean values and the standard error, *the block averaging* method is typically used. In this method, blocks of different sizes are chosen from the whole time series and the error analysis is performed with the help of these blocks (see details in Appendix A). However, we found in our work, that the *block averaging method* does not lead to satisfactory results (see the discussion in Appendix A) because of the limitation of running long time simulations. Therefore, we use an alternative approach, which is a special case of *block averaging*, in which only one block is chosen and the size of the block is equal to the size of a trajectory. In particular, many trajectories are run simultaneously and the time average of each trajectory is calculated. Next, an ensemble average of all these mean values is calculated, along with the standard error of the ensemble average obtained in this process. In this approach the length of simulation can be small (of the order of $10 - 70$ relaxation times) but the number of trajectories need to be large in order to obtain highly accurate results.

Properties, such as diffusivity, that depend on time are called dynamic properties. The short-time diffusivity, D_{s}, of an individual polymer chain can be calculated via a center-of-mass definition or by the Kirkwood formula (D_{K}), which are given respectively as

$$D_{\mathrm{s}} = \lim_{\Delta t \to 0} \frac{1}{6} \left\langle \frac{|\mathbf{r}_{\mathrm{cm}}(t + \Delta t) - \mathbf{r}_{\mathrm{cm}}(t)|^2}{\Delta t} \right\rangle \tag{4.11}$$

and

$$D_{\mathrm{K}} = \frac{1}{3N_b^2} \sum_{\nu=1}^{N_b} \sum_{\mu=1}^{N_b} (\mathrm{tr}\langle \mathbf{D}_{\nu\mu} \rangle) \tag{4.12}$$

The long-time diffusivity is calculated by tracking the mean-square displacement of the center-of-mass of each chain,

$$D = \lim_{t \to \infty} \left\langle \frac{|\mathbf{r}_{\mathrm{cm}}(t) - \mathbf{r}_{\mathrm{cm}}(0)|^2}{6t} \right\rangle \tag{4.13}$$

In this thesis, only the long-time diffusivity is calculated. For a system of N_c chains, typically D is calculated for each individual chain and averaged across all the chains in the system, i.e.,

$$D(t) = \lim_{t\to\infty} \frac{1}{N_c} \sum_{i=1}^{N_c} \left\langle \frac{|\mathbf{r}^i{}_{\mathrm{cm}}(t) - \mathbf{r}^i{}_{\mathrm{cm}}(0)|^2}{6t} \right\rangle \tag{4.14}$$

Detailes of the approach used to calculate the mean diffusivity and the standard error are presented in Appendix A. In this approach, the idea is to first calculate the mean-squared-distance (MSD $= \langle |\mathbf{r}_{\mathrm{cm}}(t) - \mathbf{r}_{\mathrm{cm}}(0)|^2 \rangle$) as a function of time. Next, the slope of the MSD vs. time curve is used to find the diffusivity (note that D = slope/6). It is important to smooth out short-term fluctuations and highlight longer-term trends in the time series data of the MSD. Appendix A discusses an efficient method of doing this with the help of *the sliding-average* method.

4.5.2 Rheological properties

The behavior of semidilute polymer solutions, when subjected to an imposed flow such as shear, elongational or mixed flow, is described in terms of various material functions. In this work, we have focused our attention on the prediction of viscosities in these flows. Expressions for viscosities in these various flow system are presented in this section.

For a planar shear flow (PSF), $\boldsymbol{\nabla v}$ is given by

$$(\boldsymbol{\nabla v})_{\mathrm{PSF}} = \begin{pmatrix} 0 & 0 & 0 \\ \dot{\gamma} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

while in planar elongational flow (PEF) it is given by

$$(\boldsymbol{\nabla v})_{\text{PEF}} = \begin{pmatrix} \dot{\epsilon} & 0 & 0 \\ 0 & -\dot{\epsilon} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

where $\dot{\gamma}$ and $\dot{\epsilon}$ are the dimensionless shear rate and elongational rate, respectively. A planar mixed flow (PMF) is a linear combination of planar shear flow and planar elongational flow, and Hunt et al. (2010) propose a velocity gradient tensor for PMF of the following form

$$(\boldsymbol{\nabla v})_{\text{PMF}} = \begin{pmatrix} \dot{\epsilon} & 0 & 0 \\ \dot{\gamma} & -\dot{\epsilon} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.15}$$

This form is referred to as the *canonical form*, in which the expanding direction is along the x axis and the contracting direction is along the y axis, with elongational field strength $\dot{\epsilon}$, while the shear gradient is along the y direction, with shear field strength $\dot{\gamma}$. Therefore, the expansion axis is always parallel to the x-axis, but the contraction axis is along the direction of one of the eigenvectors of the velocity gradient tensor. Hunt et al. (2010) have shown that the noncanonical form of the velocity gradient tensor (in which the elongation and contraction axes are always orthogonal) is equivalent to the canonical form in a rotated frame of reference. For simplicity, we use the canonical form of the velocity gradient tensor in this work.

Using $\boldsymbol{\kappa} = (\boldsymbol{\nabla v})^T_{\text{PMF}}$ in Eq. (4.1), the beads configurations and forces on them can be computed, and hence the stress tensor can be calculated. In the absence of external forces, the nondimensional stress tensor for a single chain is given by Kramer's expression (Bird et al., 1987),

$$\boldsymbol{\sigma} = \sum_{\nu=1}^{N_b} \langle \mathbf{r}_\nu \mathbf{F}_\nu \rangle \tag{4.16}$$

For a multi-chain system, the stress tensor can be shown to be (Stoltz, 2006)

$$\boldsymbol{\sigma} = \frac{1}{N_c} \sum_{\nu=1}^{N} \langle \mathbf{r}_\nu \mathbf{F}_\nu \rangle \tag{4.17}$$

Note that $\boldsymbol{\sigma}$ is a dimensionless stress tensor, which is obtained by nondimensionalizing the stress tensor by $n_p k_B T$, where n_p is number of polymer chains per unit volume i.e. $n_p = N_c/V$.

Once the stress tensor is calculated, the polymer contribution to the viscosity can be estimated. Hounkonnou et al. (1992) proposed the following expression for a generalized viscosity η_H for any arbitrary flow gradient tensor $(\boldsymbol{\nabla v})$

$$\eta_H = \frac{\dot{\boldsymbol{\Gamma}} : \boldsymbol{\sigma}}{\dot{\boldsymbol{\Gamma}} : \dot{\boldsymbol{\Gamma}}} \tag{4.18}$$

where $\dot{\boldsymbol{\Gamma}}$ is the rate of strain tensor, defined by $\dot{\boldsymbol{\Gamma}} = (\boldsymbol{\nabla v}) + (\boldsymbol{\nabla v})^T$. Using Eq. (4.18) and considering $(\boldsymbol{\nabla v}) = (\boldsymbol{\nabla v})_{\text{PMF}}$, Hunt et al. (2010) derived an expression for the planar mixed flow viscosity η_{PMF},

$$\eta_{\text{PMF}} = -\frac{2\dot{\epsilon}(\sigma_{xx} - \sigma_{yy}) + 2\dot{\gamma}\sigma_{xy}}{8\dot{\epsilon}^2 + 2\dot{\gamma}^2} \tag{4.19}$$

In the limit of pure planar shear flow (i.e., $\dot{\epsilon} = 0$), Eq. (4.19) leads to

$$\eta_{\text{PSF}} = -\frac{\sigma_{xy}}{\dot{\gamma}} \tag{4.20}$$

while in the limit of pure planar elongational flow (i.e., $\dot{\gamma} = 0$), Eq. (4.19) simplifies to

$$\eta_{\text{PEF}} = -\frac{\sigma_{xx} - \sigma_{yy}}{4\dot{\epsilon}} \tag{4.21}$$

Clearly, Eq. (4.19) can be rewritten as a linear combination of η_{PSF} and η_{PEF} as,

$$\eta_{\text{PMF}} = \frac{(4\dot{\epsilon}^2\eta_{\text{PEF}} + \dot{\gamma}^2\eta_{\text{PSF}})}{4\dot{\epsilon}^2 + \dot{\gamma}^2} \tag{4.22}$$

Eqs. (4.20) - (4.22) have been used by Hounkonnou et al. (1992); Baranyai and Cummings (1995); Todd and Daivis (1998); Daivis et al. (2003) and Hunt et al. (2010) in their nonequilibrium molecular dynamics (NEMD) simulations for the viscosity of various fluids.

While the form of the velocity gradient tensor given by $(\boldsymbol{\nabla v})_{\text{PMF}}$ [Eq. (4.15)] instinctively separates the shear and elongational flow components, it does not permit one to easily study the variation in material behavior as the flow changes smoothly from pure shear to pure elongation or vice versa. Secondly, the elongational viscosity limit of the definition of viscosity in Eq. (4.18), namely η_{PEF}, differs from the conventional definition of viscosity in PEF, which is (Bird et al., 1987)

$$\bar{\eta}_1 = -\frac{\sigma_{xx} - \sigma_{yy}}{\dot{\epsilon}} \tag{4.23}$$

As can be seen by comparing Eqs. (4.21) and (4.23), $\bar{\eta}_1 = 4\eta_{\text{PEF}}$.

An alternative form for $(\boldsymbol{\nabla v})$ in PMF proposed by Fuller and Leal (1981) resolves the first of these issues

$$(\boldsymbol{\nabla v}) = \begin{pmatrix} 0 & \dot{\Gamma}\chi & 0 \\ \dot{\Gamma} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.24}$$

where, $\dot{\Gamma}$ is the characteristic strain rate and χ ($\in [-1, 1]$) is the *mixedness* parameter which measures the relative strength of rotational and elongational components of a planar mixed flow. It can be shown that this form for $(\boldsymbol{\nabla v})$ reduces to PSF when $\chi \to 0$, while the pure PEF limit is recovered when $\chi \to 1$. Eq. (4.24) is also valid in the limit of $\chi \to -1$, which corresponds to the pure rotational flow limit. In their studies of PMF of dilute polymer solutions, Hoffman and Shaqfeh (2007) showed that Eq. (4.24) was equivalent to

$$(\boldsymbol{\nabla v}) = \begin{pmatrix} \dot{\Gamma}\sqrt{\chi} & 0 & 0 \\ \dot{\Gamma}(1-\chi) & -\dot{\Gamma}\sqrt{\chi} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.25}$$

in a suitably rotated coordinate system, where they confined their attention to elongation-dominated mixed flow, for which $\chi > 0$. Clearly, $(\boldsymbol{\nabla v})$ and $(\boldsymbol{\nabla v})_{\text{PMF}}$ are similar in structure. Comparing Eqs. (4.15) and (4.25), we can express the shear rate $\dot{\gamma}$ and elongational rate $\dot{\epsilon}$ in terms of $\dot{\Gamma}$ and χ as follows

$$\dot{\gamma} = \dot{\Gamma}(1-\chi) \tag{4.26}$$

and

$$\dot{\epsilon} = \dot{\Gamma}\sqrt{\chi} \tag{4.27}$$

Clearly, the velocity gradient defined in this manner approaches the limit of pure planar shear flow when $\chi \to 0$, and pure planar elongation flow when $\chi \to 1$. As a result, it allows us to study the smooth crossover between the pure planar shear and pure planar elongational flow limits by varying χ between 0 and 1.

In an attempt to address the issue of η_{PEF} not being equal to $\bar{\eta}_1$, we propose the following definition of the PMF viscosity

$$\eta = -\frac{\dot{\boldsymbol{\Gamma}}^{\star} : \boldsymbol{\sigma}^{\star}}{\dot{\Gamma}} \tag{4.28}$$

where $\dot{\boldsymbol{\Gamma}}^{\star} = \dot{\boldsymbol{\Gamma}}/\dot{\Gamma}$ and $\boldsymbol{\sigma}^{\star} = \dfrac{1}{2(\sqrt{\chi}+1-\chi)}\boldsymbol{\sigma}$. Using Eq. (4.25), $\dot{\Gamma}$ can be shown to be

$$\dot{\Gamma} = \sqrt{\frac{\dot{\boldsymbol{\Gamma}} : \dot{\boldsymbol{\Gamma}}}{8\chi + 2(1-\chi)^2}} \tag{4.29}$$

while Eq. (4.28) leads to

$$\eta = -\frac{\sqrt{\chi}(\sigma_{xx} - \sigma_{yy}) + (1-\chi)\sigma_{xy}}{\dot{\Gamma}(\sqrt{\chi}+1-\chi)} \tag{4.30}$$

Clearly, Eq. (4.30) reduces to Eq. (4.20) for planar shear flow in the limit $\chi \to 0$, while for $\chi \to 1$, we recover the expression for the planar elongational flow viscosity $\bar{\eta}_1$. η can be related to η_{PMF} through the expression

$$\eta = \left[\frac{4\,\chi + (1-\chi)^2}{\sqrt{\chi} + (1-\chi)}\right] \eta_{\mathrm{PMF}} \tag{4.31}$$

Equation (4.30) not only allows us to study the crossover of PMF viscosity smoothly between the pure PSF and pure PEF limits, but also leads to an expression for the PEF viscosity which is consistent with that reported in the polymer rheology literature. As we shall see subsequently, defining the PMF viscosity in this manner allows us to compare our multi-chain simulation results with the BD simulation results for single polymer chain systems (dilute solutions) obtained earlier by Prakash (2001b, 2002) and Kumar and Prakash (2003), who used the conventional definition of the various viscosities.

Chapter 5

Development of a Brownian Dynamics Simulation Algorithm for Semidilute Polymer Solutions

5.1 Introduction

This chapter describes the implementation, optimization and validation of a Brownian dynamics simulation algorithm for semidilute polymer solutions at equilibrium. This section presents the background related to mesoscopic simulation algorithm development. As discussed in Section 1.2, one of the major challenges in algorithm development is to achieve maximum computational efficiency. It is consequently worth reviewing and comparing the performance of the different mesoscopic simulations algorithms, mentioned in Section 1.2.

To our knowledge, there has been no systematic investigation to compare the performance of the different techniques in terms of their computational efficiency in the semidilute regime. Recently, however, a quantitative comparison of the predictions of the explicit solvent LB/MD method with the predictions of the implicit solvent BD method for the dynamics of a single chain in a solvent, i.e. in the dilute regime, has been carried out with a view to compare their computational efficiencies (Pham et al., 2009; Ladd et al., 2009). It was shown in Pham et al. (2009) that in order to observe the system for the same time span in physical units, significantly less CPU time is required with BD in comparison to LB/MD, for bead-spring chains with $N_b \lesssim 10^6$. The situation, however, is expected to be quite different in the semidilute regime. For the LB/MD method, the CPU cost scales linearly with the number of particles, which implies that the CPU costs grows as L^3 since the solvent particles (the calculation of whose dynamics dominates the CPU cost) are distributed on lattice grid points in a simulation box of size L. In order to prevent a chain from wrapping over itself due to spatial restriction and hence altering its static conformation, it is necessary to ensure that $L \geq 2\sqrt{\langle R_e^2 \rangle}$, where $\langle R_e^2 \rangle$ is the mean-square end-to-end distance of the chain. In the dilute case, this leads to the CPU time scaling as $N_b^{3\nu}$ for the LB/MD method. Using a simple scaling argument based on the blob picture of semidilute solutions, Pham et al. (2009) suggest that the CPU effort is even somewhat decreased in semidilute solutions due to the shrinkage of the chains resulting from the screening of excluded volume interactions (de Gennes, 1979; Doi and Edwards, 1986).

In the case of BD, even though the number of degrees of freedom is significantly reduced by eliminating the solvent, implementation of pairwise hydrodynamic interactions between segments proves to be extremely computationally expensive. For dilute polymer solutions, the computational cost of evaluating intramolecular hydrodynamic interactions arises from the need to carry out a decomposition of the diffusion tensor that appears in Eq. (4.1). A straightforward Cholesky decomposition leads to an algorithm that scales like $O(N_b^3)$. Many current implementations of single chain BD simulations, however, mitigate this large CPU cost by using Fixman's polynomial approximation to this decomposition, which leads to $O(N_b^{2.25})$ scaling (Fixman, 1986; Jendrejack et al., 2000; Kroger et al., 2000; Prabhakar and Prakash, 2004). In the case of semidilute polymer solutions, both intramolecular and intermolecular hydrodynamic interactions must be taken into account. For a typical segment on a polymer chain, the use of periodic boundary conditions to imitate bulk systems necessitates the evaluation of the sum of pairwise hydrodynamic interactions not only between the particular segment and all other segments within the primary simulation box, but also with all the other segments in all the periodic images of the box. Because of the long-range nature of hydrodynamic interactions, which decay only reciprocally with distance, this sum converges very slowly and only conditionally (Hasimoto, 1959; Beenakker, 1986). Inspired by its earlier success in summing electrostatic interactions between charged species (Allen and Tildesley, 1990; Luty et al., 1994; Toukmaji and Board, 1996; Deserno and Holm, 1998) (which are also long-ranged in nature), the problem of slow convergence has been resolved through the use of the Ewald summation technique (Ewald, 1921) — both in the context of BD simulations of colloidal suspensions where hydrodynamic interactions between particles with a finite radius must be taken into account (Smith et al., 1987; Brady et al., 1988; Rinn et al., 1999; Sierou and Brady, 2001; Banchio and Brady, 2003), and in the context of BD simulations of semidilute polymer solutions where the polymer segments are assumed to be point particles (Stoltz et al., 2006).

Rapid convergence is achieved in the Ewald sum by splitting the slowly converging sum into two sums, one of them in real space and the other in reciprocal space, both of which converge exponentially. A straightforward implementation of the Ewald sum, however, is computationally demanding, scaling like $O(N^2)$, where it may be recalled, $N = N_c \times N_b$, is the total number of beads in the primary simulation box with N_c polymer chains. Interestingly, by a suitable choice of a parameter α in the Ewald sum that tunes the relative weights of the real space and reciprocal space contributions (consequently splitting the load of calculating the total sum between the real space and reciprocal space sums), it is possible to make the computational cost of calculating either the real space or the reciprocal space sum scale like $O(N^2)$, while the remaining sum scales as $O(N)$. In their recent simulation of semidilute polymer solutions, Stoltz et al. (2006) have implemented a BD algorithm that leads to the real space sum scaling like $O(N^2)$. In the case of colloidal suspensions, accelerated BD algorithms have been developed by Brady and co-workers with the Ewald sum scaling like $O(N \log N)$ (Sierou and Brady, 2001; Banchio and Brady, 2003). The essential idea is to retain an $O(N)$ scaling for the real space sum, while reducing the complexity of the reciprocal part of the Ewald sum to $O(N \log N)$ with the help of Fast Fourier Transformation. For confined systems which are non-periodic, and in which methods based on Fourier transforms are not applicable, the Wisconsin group have recently successfully introduced a BD simulation technique they term the "general geometry Ewald-like method", which achieves $O(N \log N)$ scaling (Hernández-Ortiz et al., 2007). Analogous to the Ewald method, the technique is based on splitting the solution to Stokes equation into singular short-ranged parts and smooth long-ranged parts. Thus, even though a detailed quantitative comparison of all the currently available mesoscopic simulation techniques is yet to be carried out, they all appear to scale, in their most efficient versions, roughly linearly with system size.

In the context of electrostatic interactions, to date broadly two different classes of schemes have been proposed for the optimization of the Ewald sum over the Coulomb potential (Allen and Tildesley, 1990; Deserno and Holm, 1998; Perram et al., 1988; Fincham, 1994). One of these classes of schemes (on which the accelerated BD schemes are modeled), achieves $O(N \log N)$ scaling by assigning particles to a mesh and then using fast Fourier transform techniques to evaluate the reciprocal space part of the Ewald sum on this mesh (Luty et al., 1994; Deserno and Holm, 1998). The other class of schemes (Perram et al., 1988; Fincham, 1994) achieves $O(N^{1.5})$ scaling by balancing the computational cost of evaluating the real space and reciprocal space sums. To our knowledge, the latter approach has so far not been trialled for summing hydrodynamic interactions.

In the context of hydrodynamic interactions, it is also worth noting that the BD simulation of semidilute polymer solutions carried out by Stoltz et al. (2006) differs from BD simulations of colloidal suspensions (Brady et al., 1988; Banchio and Brady, 2003) in the procedure adopted for the calculation of *far-field* hydrodynamic interactions, even though both are based on the Ewald summation technique. While the latter are based on Hasimoto's solution of the Stokes equations for flow past a periodic array of point forces (Hasimoto, 1959), the former is based on Beenakker's solution (Beenakker, 1986), which generalises Hasimoto's treatment to the Rotne-Prager-Yamakawa (RPY) tensor (Rotne and Prager, 1969; Yamakawa, 1970). The RPY tensor, which is a regularization of the Oseen-Burgers tensor to account for finite particle radius, is commonly used in BD simulations of polymer solutions in order to maintain the positive-definiteness of the diffusion tensor. Essentially, the RPY tensor avoids the singularity of the Oseen-Burgers tensor for vanishing inter-bead distance by having two branches depending on the magnitude of the inter-bead distance relative to the particle diameter. In BD simulations of colloidal suspensions on the other hand, *near-field* hydrodynamic interactions are taken into account through short-range lubrication forces (Brady et al., 1988). Notably, even though Beenakker's periodic hydrodynamic interaction tensor is based on the RPY tensor, the original derivation by Beenakker is only valid for the long-ranged branch of the tensor.

As a result, it is not applicable to situations where bead overlap might occur. In their BD simulations of semidilute solutions, Stoltz *et al* (Stoltz et al., 2006) avoid bead overlap through the use of an excluded volume potential between beads. However, one can anticipate that bead overlap can occur in simulations of semidilute θ-solutions, since θ-solutions are commonly simulated by switching off excluded volume interactions (Prabhakar and Prakash, 2004; Sunthar and Prakash, 2005). An extension of Beenakker's derivation of the periodic RPY tensor is consequently necessary in order for it to be useful for the treatment of hydrodynamic interactions in semidilute θ-solutions.

As discussed earlier in Section 4.2, in the absence of flow, there are two major challenges to integrating the stochastic differential equation (4.1): (i) calculation of the drift term, which can be done using the Ewald summation method and (ii) calculation of the diffusion term, which can be done using the Chebyshev polynomial approximation. Also, since it is required to simulate a large number of beads, optimization of the algorithm is necessary. The plan of the chapter is as follows. In Section 5.2, the evaluation and implementation of the Ewald sum and its modification to handle overlapping beads, are discussed. Sections 5.4, 5.5 and 5.6 consider the optimization of (i) the Ewald sum for hydrodynamic interactions, (ii) the Chebyshev polynomial approximation for the decomposition of the diffusion tensor, and (iii) the execution of a single Euler time step, respectively. The optimized BD algorithm is validated by a variety of different means, under both θ and good solvent conditions, in Section 5.7. In Section 5.8, its computational cost is compared with that of the LB/MD method at a concentration that lies in the semidilute regime.

5.2 Ewald Summation Method for Hydrodynamic Interactions

5.2.1 Evaluation of $\sum_\mu \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_\mu$ as an Ewald sum

Beenakker's (Beenakker, 1986) representation of the sum $\sum_\mu \boldsymbol{\mathsf{D}}_{\nu\mu} \cdot \mathbf{F}_\mu$ as an Ewald sum for infinite periodic systems, using the RPY tensor to represent hydrodynamic interactions, has the form

$$\sum_{\mu=1}^{N} \boldsymbol{\mathsf{D}}_{\nu\mu} \cdot \mathbf{F}_\mu = \left(1 - \frac{6a\alpha}{\sqrt{\pi}} + \frac{40a^3\,\alpha^3}{3\sqrt{\pi}}\right) \mathbf{F}_\nu + {\sum_{\mathbf{n}}}' \sum_{\mu=1}^{N} \boldsymbol{\mathsf{M}}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}}) \cdot \mathbf{F}_\mu$$
$$+ \sum_{\mathbf{k}\neq\mathbf{0}} \boldsymbol{\mathsf{M}}^{(2)}(\mathbf{k}) \cdot \left\{ \cos(\mathbf{k}\cdot\mathbf{r}_\nu) \sum_{\mu=1}^{N} \cos(\mathbf{k}\cdot\mathbf{r}_\mu)\mathbf{F}_\mu - \sin(\mathbf{k}\cdot\mathbf{r}_\nu) \sum_{\mu=1}^{N} \sin(\mathbf{k}\cdot\mathbf{r}_\mu)\mathbf{F}_\mu \right\} \tag{5.1}$$

where the first and the second sums on the right hand side, both of which converge rapidly, are carried out in real and reciprocal space, respectively. The first term on the RHS is the correction due to self-interactions and does not involve any summation. A brief introduction to the Ewald summation method and the detailed derivation of Eq. (5.1) is presented in Appendix B. In Eq. (5.1), the parameter α determines the manner in which the computational burden is split between the two sums. The vector $\mathbf{r}_{\nu\mu,\mathbf{n}}$ is defined by $\mathbf{r}_{\nu\mu,\mathbf{n}} = \mathbf{r}_\nu - \mathbf{r}_\mu + \mathbf{n}L$, where $\mathbf{n} = (n_x, n_y, n_z)$ is the lattice vector with n_x, n_y, n_z being integer numbers (see Fig. 5.1). The first summation on the RHS of Eq. (5.1) is carried out in the original simulation box and over all the neighboring periodic images. The prime on the summation indicates that the lattice vector $\mathbf{n} = \mathbf{0}$ is omitted for $\nu = \mu$. $\boldsymbol{\mathsf{M}}^{(1)}(\mathbf{r})$ is a 3×3 matrix (in real space), which depends on a and α, and $\boldsymbol{\mathsf{M}}^{(2)}(\mathbf{k})$ is also a 3×3 matrix (in reciprocal space), which depends on a, α and the volume of the simulation box V. The expressions for $\mathsf{M}^{(1)}(\mathbf{r})$ and $\mathsf{M}^{(2)}(\mathbf{k})$ are

$$\begin{aligned}\mathsf{M}^{(1)}(\mathbf{r}) = &\Bigg[\operatorname{erfc}(\alpha r)\left(\frac{3a}{4r} + \frac{a^3}{2r^3}\right) + \frac{\exp\left(-\alpha^2 r^2\right)}{\sqrt{\pi}}\Bigg(3a\,\alpha^3 r^2 - \frac{9a\alpha}{2} \\ &+ 4a^3\,\alpha^7 r^4 - 20a^3\,\alpha^5 r^2 + 14a^3\,\alpha^3 + \frac{a^3\alpha}{r^2}\Bigg)\Bigg]\,\boldsymbol{\delta} \\ &+ \Bigg[\operatorname{erfc}(\alpha r)\left(\frac{3a}{4r} - \frac{3a^3}{2r^3}\right) + \frac{\exp\left(-\alpha^2 r^2\right)}{\sqrt{\pi}}\Bigg(\frac{3a\alpha}{2} - 3a\,\alpha^3 r^2 \\ &- 4a^3\,\alpha^7 r^4 + 16a^3\,\alpha^5 r^2 - 2a^3\,\alpha^3 - \frac{3a^3\alpha}{r^2}\Bigg)\Bigg]\,\hat{\mathbf{r}}\hat{\mathbf{r}}\end{aligned} \tag{5.2}$$

with erfc denoting the complementary error function, and

$$\mathsf{M}^{(2)}(\mathbf{k}) = \left(a - \frac{a^3 k^2}{3}\right)\left(1 + \frac{k^2}{4\,\alpha^2} + \frac{k^4}{8\alpha^4}\right)\left(\frac{6\pi}{k^2 V}\right)\exp\left(\frac{-k^2}{4\alpha^2}\right)\left(\boldsymbol{\delta} - \hat{\mathbf{k}}\,\hat{\mathbf{k}}\right) \tag{5.3}$$

The second summation in Eq. (5.1) (denoted here as the reciprocal space sum) is carried out over lattice vectors $\mathbf{k} = 2\pi\mathbf{n}/L$. In Eq. (5.2), r and $\hat{\mathbf{r}}$ are the magnitude and unit vector, respectively, in the direction of $\mathbf{r}$. In Eq. (5.3), k and $\hat{\mathbf{k}}$ are the magnitude and unit vector, respectively, corresponding to $\mathbf{k}$.

5.2.2 Modification of the Ewald sum to account for overlapping beads

As pointed out earlier, the derivation of the Ewald sum by Beenakker (1986) is valid *only* for the branch $\mathcal{A}$ of the RPY functions Ω_1 and Ω_2 (Eq. (4.5)), which forbids its use for the case of overlapping beads ($r < 2a$). The original expression consequently cannot be used for the simulation of θ solvents by neglecting excluded volume interactions, as in this case beads on the same or on different chains are highly prone to overlap with each other. The Ewald sum can be modified to account for such situations (Zhou and Chen, 2006), and we discuss it here

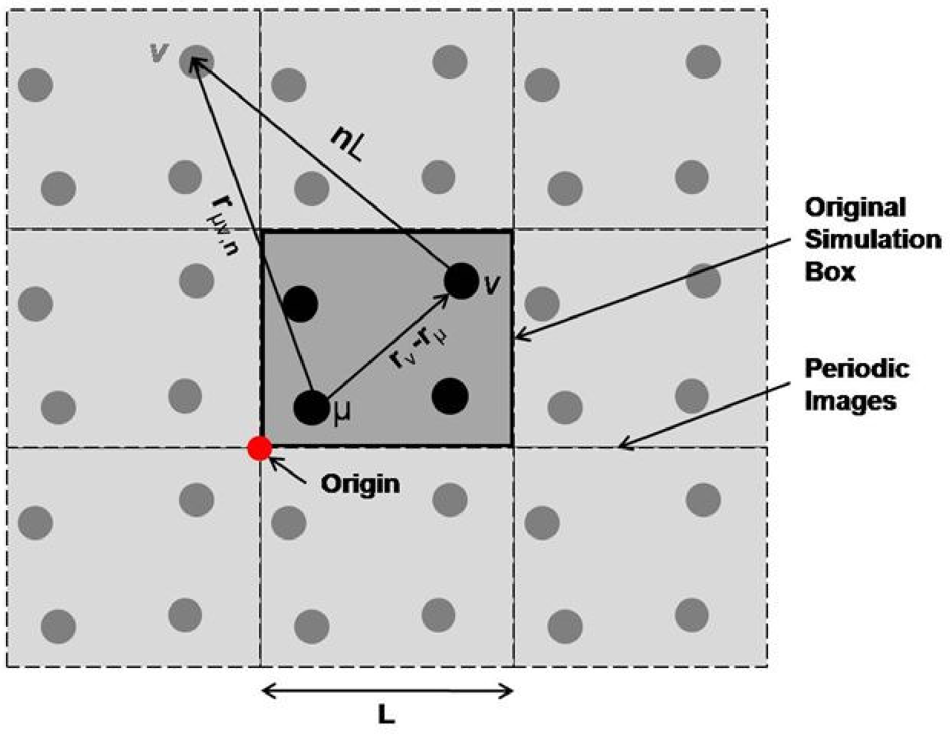


Figure 5.1: Periodic boundary conditions in 2-D: demonstration of the distance vector between two beads

as follows.

Starting from a given bead ν, we consider all those beads that have distances less than $2a$ from it, including bead ν itself. By a proper re-labeling, we can assume that these are the beads $\mu = 1, \ldots, N^*$. The number of non-overlapping particles is thus $N - N^*$. As the correction needs to be carried out only in the real space sum, the first summation on the RHS of Eq. (5.1) $\left(\sum_{\mathbf{n}}' \sum_{\mu=1}^{N} \mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}}) \cdot \mathbf{F}_\mu\right)$ is replaced by

$$\sum_{\substack{\mu=1 \\ \mu\neq\nu}}^{N^*} \mathbf{M}_{\boldsymbol{\mathcal{B}}}{}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}}) \cdot \mathbf{F}_\mu + \sum_{\mathbf{n}\neq\mathbf{0}} \sum_{\mu=1}^{N^*} \mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}}) \cdot \mathbf{F}_\mu + \sum_{\mathbf{n}} \sum_{\mu=N^*+1}^{N} \mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}}) \cdot \mathbf{F}_\mu \tag{5.4}$$

where, ${\mathbf{M}_{\mathcal{B}}}^{(1)}$ is equivalent to the $\mathbf{M}^{(1)}$ matrix, but for branch $\mathcal{B}$ of the RPY tensor given by Eq. (4.6). The summation in the first term is carried out only over the overlapping particles in the original simulation box ($\mathbf{n} = \mathbf{0}$). The second summation is carried out over the periodic images of the overlapping particles (whose distances are more than $2a$), and the third summation is similar to that given in Eq. (5.1) but here it is carried out only over the non-overlapping particles. Note that the second sum is not carried out in the original box. In order to make this sum extend over the original box and periodic images, a term is added and subtracted as follows

$$\begin{aligned}\sum_{\substack{\mu=1\\\mu\neq\nu}}^{N^*} {\mathbf{M}_{\mathcal{B}}}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\cdot\mathbf{F}_\mu &+ \sum_{\mathbf{n}\neq\mathbf{0}}\sum_{\mu=1}^{N^*}\mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})\cdot\mathbf{F}_\mu \\ + \sum_{\substack{\mu=1\\\mu\neq\nu}}^{N^*}\mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\cdot\mathbf{F}_\mu &- \sum_{\substack{\mu=1\\\mu\neq\nu}}^{N^*}\mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\cdot\mathbf{F}_\mu \\ &+\sum_{\mathbf{n}}\sum_{\mu=N^*+1}^{N}\mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})\cdot\mathbf{F}_\mu\end{aligned} \tag{5.5}$$

Note that while the first sum involves branch $\mathcal{B}$ of the RPY tensor, the third sum involves branch $\mathcal{A}$. The second, third and fifth summations of the above equation together represent the original real space sum in Eq. (5.1). Equation (5.5) can consequently be rearranged as

$$\left(\sum_{\mathbf{n}}{}'\sum_{\mu=1}^{N}\mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})\cdot\mathbf{F}_\mu\right) + \sum_{\substack{\mu=1\\\mu\neq\nu}}^{N^*}\left[{\mathbf{M}_{\mathcal{B}}}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}}) - \mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\right]\cdot\mathbf{F}_\mu \tag{5.6}$$

where the second summation is carried over all overlapping particles in the original box. Denoting the second term in Eq. (5.6) by $\mathbf{M}^*$, it is straightforward to show that

$$\mathsf{M}^*(\mathbf{x}) = \boldsymbol{\delta}\left[1 - \frac{1}{2x^3}\left(\frac{3x^2}{4}+1\right)^2\right] + \hat{\mathbf{x}}\,\hat{\mathbf{x}}\left[\frac{1}{2x^3}\left(\frac{3x^2}{4}-1\right)^2\right] \tag{5.7}$$

where $\mathbf{x} = \mathbf{r}_{(\nu\mu,\mathbf{n}=\mathbf{0})}/a$ and $\hat{\mathbf{x}}$ is the unit vector in the direction of $\mathbf{r}_{(\nu\mu,\mathbf{n}=\mathbf{0})}$ (see Appendix B.3.4). The modified form of the Ewald sum that is valid for arbitrary inter-particle distance is consequently

$$\begin{aligned}\sum_{\mu=1}^{N}\mathsf{D}_{\nu\mu}\cdot\mathbf{F}_\mu &= \left(1-\frac{6a\alpha}{\sqrt{\pi}}+\frac{40a^3\,\alpha^3}{3\sqrt{\pi}}\right)\mathbf{F}_\nu + {\sum_{\mathbf{n}}}'\sum_{\mu=1}^{N}\mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})\cdot\mathbf{F}_\mu \\ &+ \sum_{\mathbf{k}\neq 0}\mathsf{M}^{(2)}(\mathbf{k})\cdot\left\{\cos(\mathbf{k}\cdot\mathbf{r}_\nu)\sum_{\mu=1}^{N}\cos(\mathbf{k}\cdot\mathbf{r}_\mu)\mathbf{F}_\mu - \sin(\mathbf{k}\cdot\mathbf{r}_\nu)\sum_{\mu=1}^{N}\sin(\mathbf{k}\cdot\mathbf{r}_\mu)\mathbf{F}_\mu\right\} \\ &+ \sum_{\substack{\mu=1\\ \mu\neq\nu}}^{N^*}\mathsf{M}^*(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\cdot\mathbf{F}_\mu\end{aligned} \tag{5.8}$$

5.2.3 Implementation of the Ewald summation method

As discussed earlier, the real space sum is carried out over all the periodic images while the reciprocal space sum involves particles only in the original simulation box. There are three parameters which control the accuracy of both the real and reciprocal space sums: n_{max}, an integer which defines the range of the real space sum (governed by the number of periodic images, see Fig. 5.1), k_{max}, an integer that defines the summation range in reciprocal space and the Ewald parameter α. These three parameters are related to each other from the point of view of accuracy and speed. A large value of α makes the real space sum converge faster (since a smaller value of n_{max} is required). However, this leads to the reciprocal space sum requiring a larger number of wave-vectors k_{max}. On the other hand, a small value of α implies an expensive real space sum but a cheaper reciprocal space sum. The optimal choice of these parameters has been discussed previously by Fincham (1994) in the context of electrostatic interactions. Here, a similar study is performed for hydrodynamics interactions.

Choice of Ewald parameters

At fixed monomer bulk concentration c, the box size increases as $N^{1/3}$. As can be seen from Eqs. (5.1) and (5.2), the convergence of the real space sum depends on the complementary error function $\text{erfc}(\alpha r)$, where r is the distance between a pair of beads. In practice, the sum is evaluated only for $r \leq r_\text{c}$, where r_c denotes a *cutoff* radius. The value of α is chosen such that $\text{erfc}(\alpha r_\text{c})$ is small. At large values of the argument, $\text{erfc}(\alpha r_\text{c})$ behaves as $\exp(-\alpha^2 {r_\text{c}}^2)$. If we specify M such that $\exp(-M^2)$ is very small, then

$$\alpha^2 {r_\text{c}}^2 = M^2 \quad \text{or} \quad \alpha = M/r_\text{c} \tag{5.9}$$

Similarly the rate of convergence of the reciprocal space sum is controlled by the factor $\exp(-k^2/4\alpha^2)$. If it is required (Fincham, 1994) that the accuracy of the real space sum is roughly equal to that of the reciprocal space sum at the reciprocal space cutoff, K_r, then using Eq. (5.9) we find

$$M^2 = K_r^2/(4\alpha^2) \quad \text{or} \quad K_r = 2\alpha M = 2M^2/r_\text{c} \tag{5.10}$$

These relations allow us to specify α and K_r for given values of M and r_c, while the latter parameters control the accuracy and speed of the algorithm, as discussed subsequently.

The real space and reciprocal space sums

Locating all pairs of beads which are separated by less than the cutoff distance r_c is the first step in evaluating the real space sum. A naive all-pairs neighbor search results in $O(N^2)$ performance and therefore the link-cell method, which is a cell-based neighbor search method, is used to improve the performance (Hockney et al., 1973; Griebel et al., 2007; Grest et al., 1989). The calculation of the infinite real space sum is thus reduced to the calculation of the sum locally over only a small number of neighboring beads. Here, the neighbor search is implemented with cells of side $r_\text{c}/5$. The reciprocal space sum is more straightforward to implement. The major effort is expended in the evaluation of terms of the form $\exp(i\mathbf{k} \cdot \mathbf{r}_\mu)$. The method adopted here precomputes the components of these factors by recursion and stores them (Fincham, 1994). This avoids calling the complex exponential function repeatedly. However, it involves a substantial amount of computer memory.

5.3 Validation of $\sum_\mu \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_\mu$ Calculations

Three different approaches have been used to validate the calculations of the Ewald sum:

1. A single-chain code (developed to simulate a dilute polymer solution) (Prabhakar, 2005; Prabhakar and Prakash, 2004) is used to validate the Ewald sum in the dilute regime

2. Hasimoto's solution (Hasimoto, 1959) of the Stokes equation of motion for viscous fluid flow past a periodic array of spherical obstacles is used to validate the implementation of periodic boundary conditions, again in the dilute limit.

3. The pairwise summation of hydrodynamic interactions is carried out explicitly i.e. without using the Ewald summation and this sum is then compared with the Ewald sum. This approach enables us to validate the Ewald sum

for finite concentration systems.

Each of these approaches are discussed below in detail.

5.3.1 Comparison with single-chain simulations

In this approach, firstly the values of $\sum_{\mu} [\mathbf{D}_{\nu\mu} \cdot \mathbf{F}_{\mu}]$ from a single-chain code, written for dilute polymer solutions (Prabhakar, 2005; Prabhakar and Prakash, 2004), are computed. Thereafter, data for $\sum_{\mu} [\mathbf{D}_{\nu\mu} \cdot \mathbf{F}_{\mu}]$ for the same set of parameters ($a = 0.1$, $N_c = 1$, $N_b = 20, 100$) are generated using the current code for various values of box size L. The element by element difference between the vector $\mathbf{D} \cdot \mathbf{F}$ of the single-chain code and the current code is calculated to get the sum of the square of the error (SSE). As can be seen in Fig. 5.2, the SSE approaches to $\sim 10^{-8}$ as $1/L \to 0$, which indicates that at lower concentrations, the current code agrees with the single-chain code. Note that a tolerance criteria of 10^{-8} is used in the current code for the convergence of the Ewald summation.

5.3.2 Verification of the prediction of the Madelung constant

In this approach, the Madelung constant for the motion of a viscous fluid flowing past a periodic array of spherical particles is verified by the current code. Madelung's constant was first derived by Hasimoto (1959) by solving the Stokes equation for the system mentioned above. Note that the Hasimoto solution assumes that the suspension is dilute. By considering a single particle at the centre of a box and periodic images of that particle, Hasimoto derived an expression

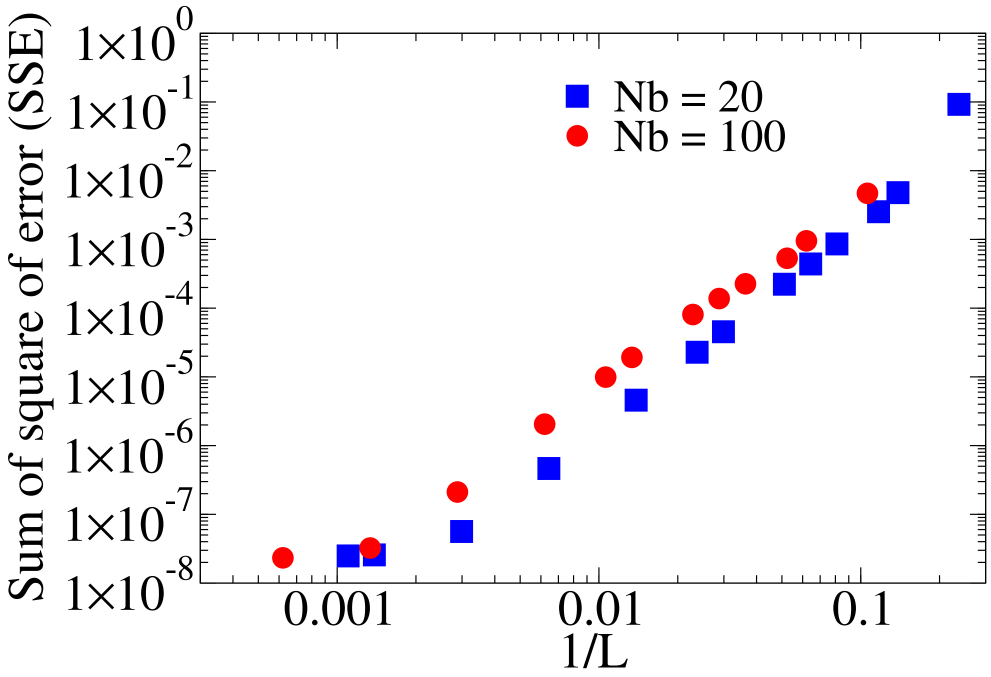


Figure 5.2: Validation of the Ewald sum with the single-chain code

for the velocity field at the centre of a box,

$$\mathbf{v} = \left(1 - M_d \frac{a}{L}\right) \mathbf{F} + O(a^3) \tag{5.11}$$

where a is the radius of the particle, $\mathbf{F}$ is the force on the particle, L is the box size and M_d is the *Madelung constant* whose value is reported by Hasimoto (Hasimoto, 1959) to be 2.8373. Derivation of Eq. (5.11) is presented in Hasimoto (1959). In order to calculate the Madelung constant, firstly the diffusion tensor D is calculated for a single bead located at the center of the box with an infinite number of periodic images. Since there is only one particle, D is diagonal, and it can be shown that the value of the diagonal element, for small values of a/L should equal $\left(1 - M_d \frac{a}{L}\right)$. As expected, Fig. 5.3 shows that the computed value of M_d approaches 2.8373 for small values of a/L. This approach ensures that the

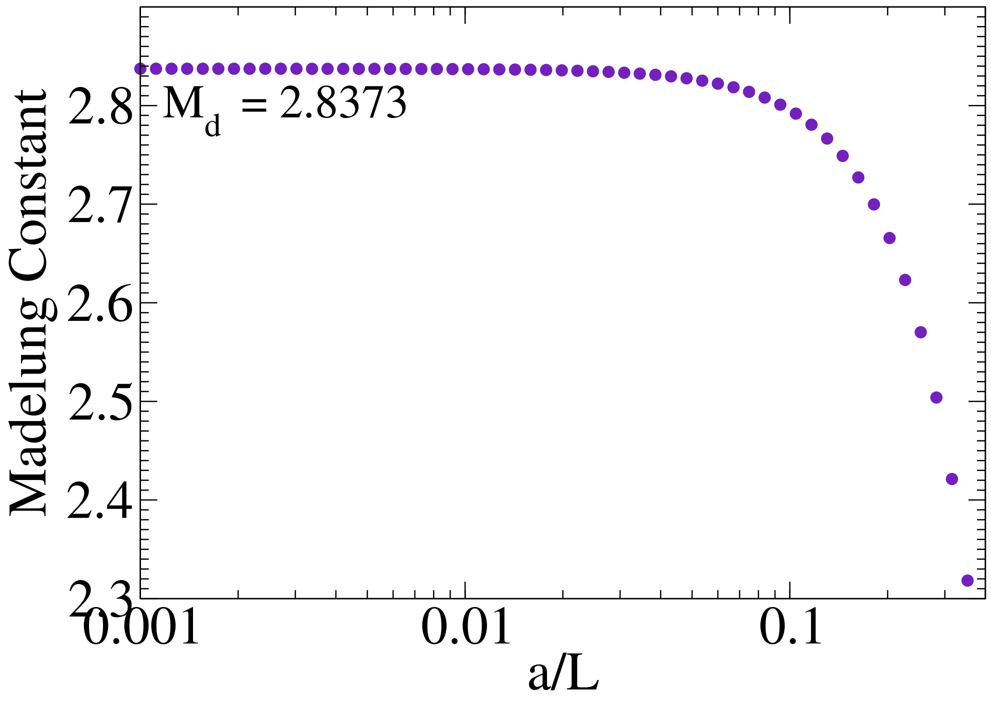


Figure 5.3: Validation of the Ewald sum with Hasimoto's solution for the Madelung constant

periodic boundary conditions are implemented correctly in the algorithm.

5.3.3 Explicit vs Ewald sum

The velocity of the solvent, $\mathbf{v}_{\nu}$, at the location of a bead ν, due to forces on all the other beads in the system, is given by,

$$\mathbf{v}_{\nu} = \sum_{\mathbf{n}} \sum_{\mu=1}^{N} [\mathbf{D}_{\nu\mu,\mathbf{n}} \cdot \mathbf{F}_{\mu}] \tag{5.12}$$

As mentioned earlier, in general, it is not possible to evaluate this sum explicitly because of the slow and conditional nature of the sum. For a periodic system, however, as we have seen, the sum can be evaluated using an Ewald summation technique. It would seem, consequently, that we cannot validate the result of

Ewald summation by comparison with the explicit sum, since the latter cannot in general be evaluated.

We have found, however, that by constructing a special kind of periodic system, we can in fact compare the Ewald and explicit sums, and as a result obtain a validation of our implementation of the Ewald sum for systems at finite concentrations.

Before we discuss the special periodic arrangement, it is worth making a few remarks on our observations for a general periodic arrangement of particles, where they are placed at random in the original simulation box. A converged value for the velocity $\mathbf{v}_{\nu}$ at an arbitrary location $\mathbf{r}_{\nu}$ in the original simulation box, for a periodic but random system of particles, was obtained when it was evaluated using the explicit sum, provided the sum of forces on all the beads was zero, i.e., $\sum_{\mu=1} \mathbf{F}_{\mu} = \mathbf{0}$. The converged value was independent of the order in which summation was carried out, i.e., whether all the particles in the simulation box were first summed, and then all the sums over the images carried out, or the sum for each particle over all the images was followed by a sum over all particles in the original simulation box. However, this converged value of velocity did not agree with that obtained using the Ewald sum for the same arrangement of particles.

We are not entirely sure of the reason for this discrepancy. A possible reason is that while the Ewald sum is strictly valid only for a periodic system the explicit sum has to in some sense *discover* the periodicity, and is unable to do so for a random periodic arrangement of particles. In order to check this hypothesis, we evaluated the solvent velocity with the explicit sum at locations, one of which was the periodic image of the other. The only points in the simulation box that also have a periodic image in the box are points on the walls of the box. As a result, we placed a particle on one of the walls, and evaluated the velocity $\mathbf{v}_{\nu}$ at

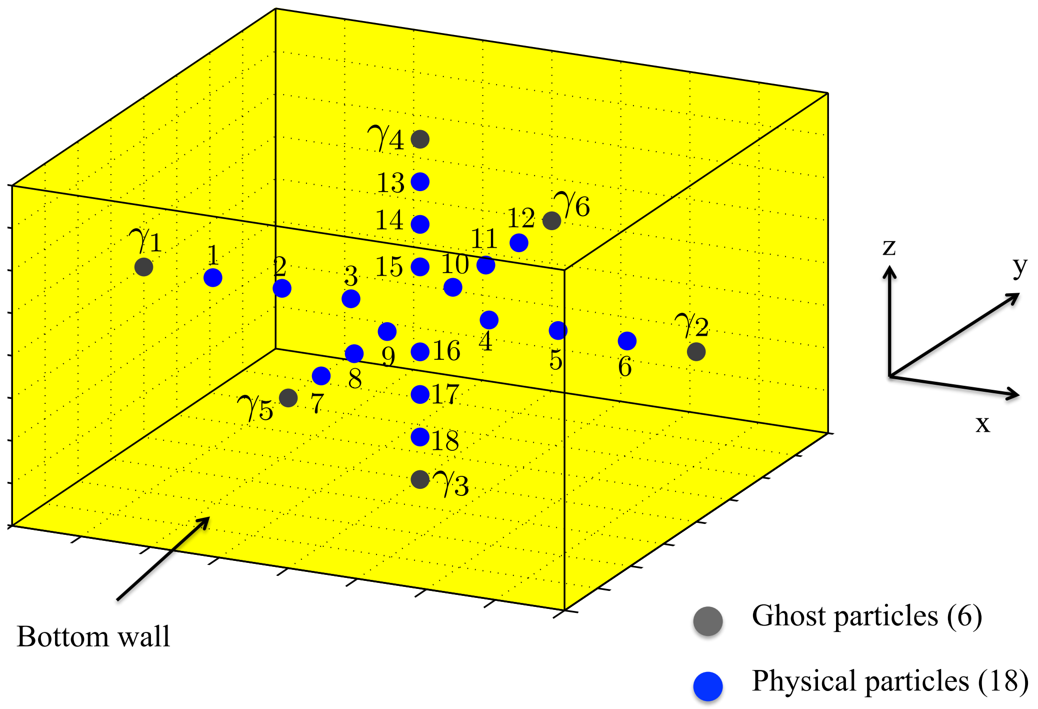


Figure 5.4: An example to illustrate the placement of particles in the simulation box in a highly symmetric manner

this particle and at it's image on the opposite wall. Note that each particle is simultaneously an original particle and the image of the particle on the opposite wall. In addition to placing the particles on the walls, we also had to set the force on the particles to zero to ensure they do not appear in the explicit sum. The reason for considering such *ghost* particles is the following. if γ_1 and γ_2 are the two particles on the opposite walls and $\mathbf{r}_{\gamma_1}$ and $\mathbf{r}_{\gamma_2}$ are their locations, then $\mathbf{r}_{\gamma_1\gamma_2,\mathbf{n}} = \mathbf{0}$ for $\mathbf{n}$ corresponding to the two image boxes that lie adjacent to these particles, and this leads to problems in the calculation of the explicit sum (see Fig. 5.4). By following this procedure, we found that while the Ewald sum leads to a perfectly periodic solvent velocity at the wall particles, the explicit sum leads to differing values.

An alternative special periodic system however leads to a prediction of a periodic solvent velocity by the explicit sum, which also agrees with the value obtained with the Ewald sum. In this special arrangement, rather than choosing a random distribution of particles, the particles are placed in a highly symmetric manner in the original simulation box. A typical example is shown in Fig. 5.4. Further, the particles are subjected to forces that are also symmetric. Thus, for instance, particles 1 and 6 in Fig. 5.4 have equal and opposite forces acting on them. The constraint $\sum_{\mu=1} \mathbf{F}_{\mu} = 0$ is also imposed. Under these conditions, we find that for a range of values of concentration c, bead radii a, and number of beads N, the velocities of the solvent at all the particle locations are identical in the explicit and Ewald sum, within the prescribed tolerance for the convergence of the sums. This completes the validation of the implementation of the Ewald sum under a variety of different circumstances.

5.4 Optimization of the Evaluation of $\sum_{\mu} \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_{\mu}$

The Ewald parameter α, which splits the computational burden between the real space sum and the reciprocal space sum, is related to the real space cutoff $r_{\rm c}$ by Eq. (5.9). The aim of optimization is to minimize the total execution time (which is the sum of the real space execution time, T_R and the reciprocal space execution time, T_F), with respect to the real space cutoff $r_{\rm c}$. Following Fincham (1994), the execution time T_R is calculated as follows. A sphere of cutoff radius $r_{\rm c}$ contains on average $N_{\rm r_c} = \dfrac{4\pi}{3} r_{\rm c}^3 \, c$ beads. Each bead interacts with the $N_{\rm r_c}$ beads that surround it. Since for symmetry reasons, each pair interaction needs to be considered only once, the execution time is

$$T_R = \frac{1}{2} N \frac{4\pi}{3} r_{\rm c}^3 \, c \, t_r \tag{5.13}$$

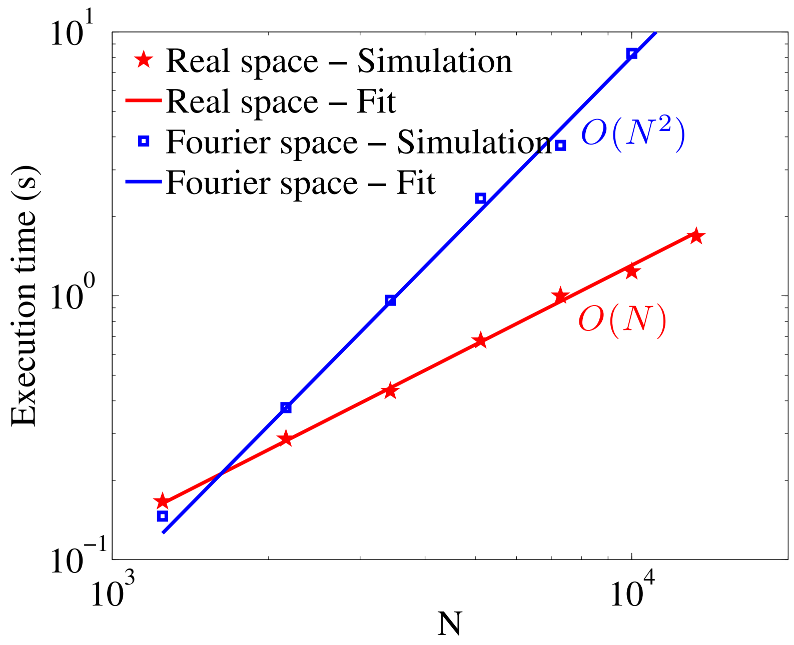
where t_r is a constant parameter with a unit of time, and it depends on the code architecture and on the execution time to evaluate one interaction. Equation (5.13) is fit to data obtained by running BD simulations for a range of parameters on a 156 SGI Altix XE 320 cluster, and the fitted parameter t_r is then found to be $0.75\,\mu s$. The execution time T_F is evaluated as follows. Within the cutoff K_r, the volume of the reciprocal space is $\dfrac{4\pi}{3}K_r^3 = \dfrac{4\pi}{3}\dfrac{8M^6}{r_\mathrm{c}^3}$ (the latter follows from Eq. (5.10)). The reciprocal space points are defined by $k = \dfrac{2\pi}{L}(l, m, n)$ where l, m, n are integers and L is the simulation box size. The volume of reciprocal space per point is, thus, $(2\pi/L)^3$, and $\dfrac{4\pi}{3}\dfrac{8M^6}{r_\mathrm{c}^3}\dfrac{L^3}{8\pi^3}$ is the number of points in the cutoff sphere. Using $L^3 = N/c$ to highlight the N dependence at fixed concentration c, the number of reciprocal space points in the cutoff sphere becomes $\dfrac{4\pi}{3}\dfrac{M^6}{\pi^3}\dfrac{N}{cr_\mathrm{c}^3}$. It is worth pointing out that for fixed cutoff radius, the number of k-space points increases as N, because the concentration of points in reciprocal space increases with system size. Further, inversion symmetry of reciprocal space halves the number of reciprocal space points mentioned above. A sum over the N beads must be performed for each k-space point, so the execution time is

$$T_F = \frac{1}{2}\,\frac{4\pi}{3}\,\frac{M^6}{\pi^3}\,\frac{N^2}{cr_\mathrm{c}^3}\,t_f \tag{5.14}$$

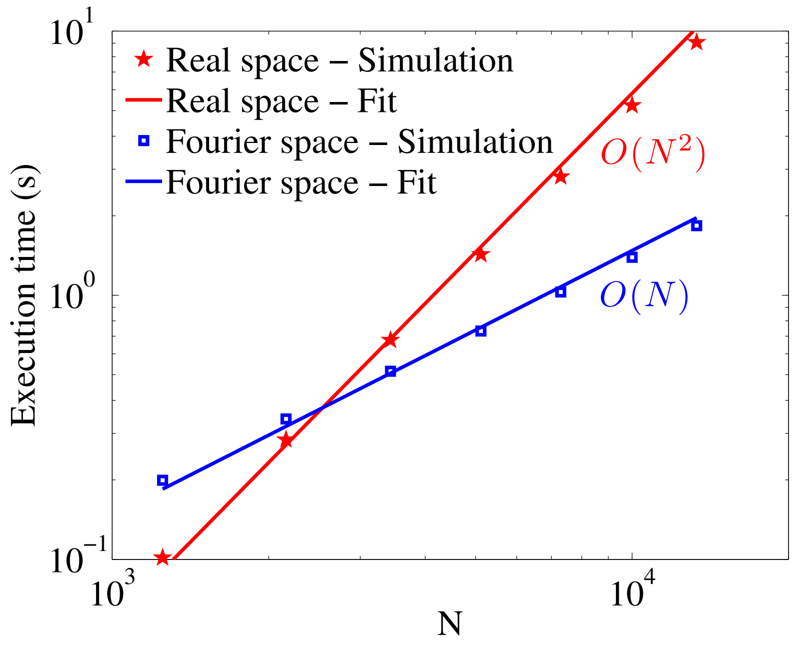
where t_f is a constant parameter with a unit of time, and it depends on the code architecture and on the execution time to evaluate one term in the sum. As in the real space instance, Eq. (5.14) is fitted to simulation data to obtain $t_f = 0.063\,\mu s$. The total execution time is consequently

$$T = T_R + T_F = \frac{1}{2}\,\frac{4\pi}{3}\left[N\,r_\mathrm{c}^3\,c\,t_r + \frac{M^6}{\pi^3}\,\frac{N^2}{cr_\mathrm{c}^3}\,t_f\right] \tag{5.15}$$

Equation (5.15) shows that, for fixed M and r_c, T_R varies as N, but T_F varies as N^2, because of the increasing concentration of points in reciprocal space.



(a)



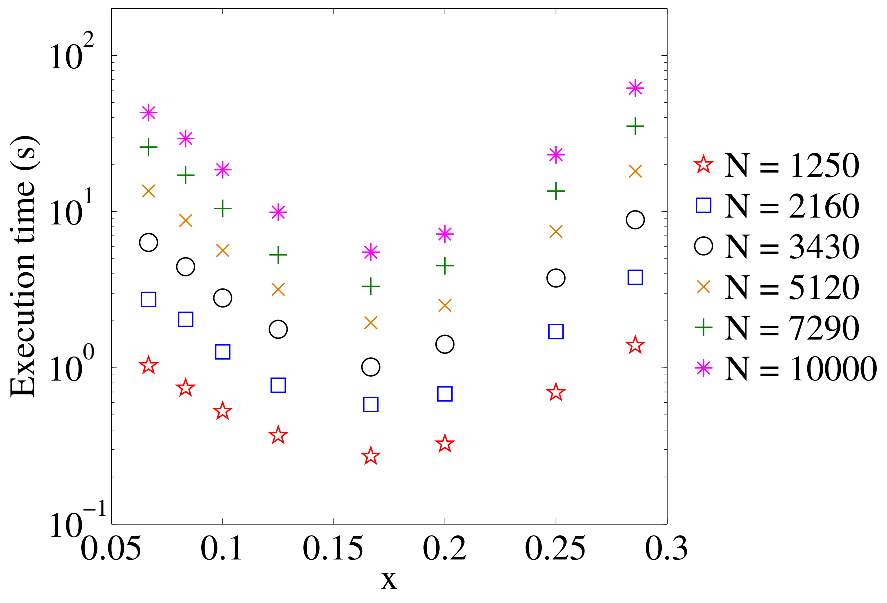
(b)

Figure 5.5: Execution time scaling for the real space and the Fourier space sums for: (a) Constant r_c (b) Constant L/r_c

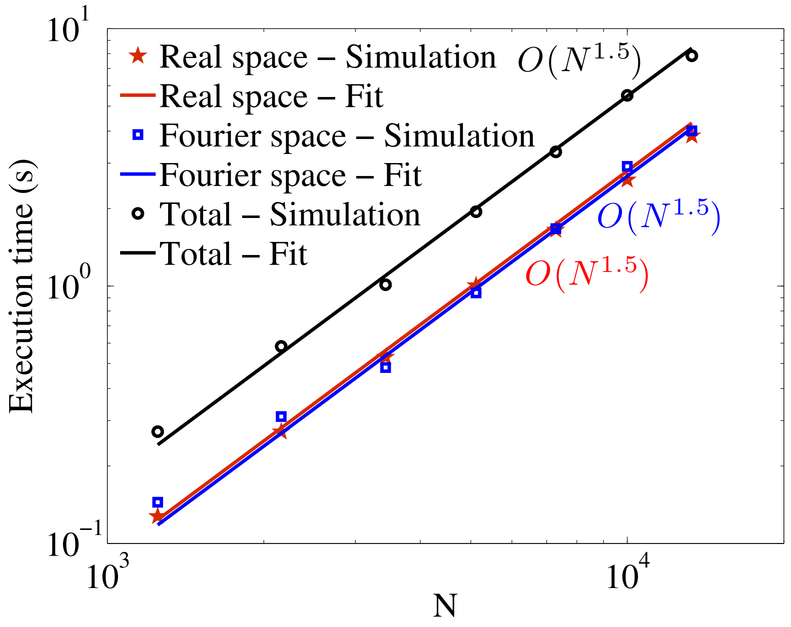
This behavior is demonstrated for $c = 4.44\,c^\star$ and $r_\mathrm{c} = 10$ in Fig. 5.5 (a) for the simulation data (symbols), which agrees with the expressions given in Eqs. (5.13) and (5.14) (solid lines). Here $c^\star$ is the overlap concentration defined by $N_b/\left[\frac{4\pi}{3}(R_g^0)^3\right]$, where R_g^0 is the gyration radius of a polymer chain in the dilute limit. To increase the value of N, we fix the value of beads per chain at $N_b = 10$ and increase the number of chains N_c. Conversely, if r_c is increased as the system size increases in such a way that r_c/L is constant (as in the approach adopted by Stoltz et al. (2006)), then since $c = N/L^3$, T_R varies as N^2 but T_F varies as N. Fig. 5.5 (b) displays this behavior for $N_b = 10$, $c = 4.44\,c^\star$, $M = 3.3$ and $L/r_\mathrm{c} = 3$. Once again the simulation data is seen to match the expressions given in Eqs. (5.13) and (5.14). This suggests that by appropriate choice of parameters it may be possible to achieve better than N^2 behavior in the total time T. For a given accuracy, the only free parameter is r_c, since this determines α and hence K by Eqs. (5.9) and (5.10). To find the value of r_c which minimizes the total execution time, we set $dT/dr_\mathrm{c} = 0$. This leads to

$$\left(r_\mathrm{c}^\mathrm{E}\right)_\mathrm{opt} = \frac{M}{\sqrt{\pi}} \left(\frac{t_f}{t_r}\right)^{1/6} \frac{N^{1/6}}{c^{1/3}} \tag{5.16}$$

Thus the optimal choice of the cutoff radius $\left(r_\mathrm{c}^\mathrm{E}\right)_\mathrm{opt}$ increases slowly (1/6th power) with N. The validity of Eq. (5.16) has been verified by carrying out simulations. Assuming that the quantity $N^{1/6}$ in Eq. (5.16) is replaced by N^x, various values of the exponent x are selected in place of the exponent $1/6$, and the total execution time for a given value of N is estimated. Figure 5.6 (a) shows the total execution time as a function of the exponent x and it is clear that the minimum execution time is achieved when $x = 1/6$ as given by Eq. (5.16), for all N. Substituting



(a)



(b)

Figure 5.6: (a) Total execution time vs. exponent x for various N (b) Power law scaling for T_R, T_F and T_{opt} at $\left(r_c^{\text{E}}\right)_{\text{opt}}$

$\left(r_{\mathrm{c}}^{\mathrm{E}}\right)_{\mathrm{opt}}$ in Eq. (5.15) we find for the optimal time

$$T_{\mathrm{opt}} = 2T_R = 2T_F = \frac{4\pi}{3}\, N^{1.5}\, \frac{M^3}{\pi^{1.5}}\, \sqrt{t_r\, t_f} \tag{5.17}$$

Thus, when the total time is optimized, it is equally divided between the real and reciprocal space parts of the calculation. This is verified in Fig. 5.6 (b), which displays plots of T_R, T_F and T as a function of N, at $x = 1/6$ and for $N_b = 10$, $c = 4.44c^{\star}$ and $M = 3.3$. Symbols indicate simulation data and solid lines correspond to Eq. (5.17) with the appropriate values for the various parameters. Equation (5.17) also indicates that the real space, reciprocal space and total time scale as $N^{1.5}$. Simulation results shown in Fig. 5.6 (b) substantiate this prediction. These results are similar to those obtained by Fincham (1994) in the context of electrostatic interactions.

5.5 Decomposition of the Diffusion Tensor

In component form, the decomposition rule (Eq. (4.2)) for obtaining the block matrix $\boldsymbol{\mathcal{B}}$ can be expressed as follows,

$$\sum_{\beta=1}^{N} \sum_{q=1}^{3} \mathcal{B}_{\nu\beta}^{rq}\, \mathcal{B}_{\mu\beta}^{sq} = \mathcal{D}_{\nu\mu}^{rs} \tag{5.18}$$

where $\{\nu, \beta, \mu = 1, \ldots, N\}$, $\{r, q, s = 1, 2, 3\}$, and $\mathcal{D}_{\nu\mu}^{rs}$ is the 'rs'th Cartesian component of the tensor $\mathbf{D}_{\nu\mu}$. The matrix $\boldsymbol{\mathcal{B}}$ is not unique. Assuming that $\boldsymbol{\mathcal{B}}$ is a lower (or upper) triangular matrix leads to its calculation using a Cholesky decomposition of $\boldsymbol{\mathcal{D}}$, which as mentioned earlier in Section 5.1, requires $O(N^3)$ operations. Fixman's (Fixman, 1986) approach achieves an attenuation of this CPU intensity by recognizing that (i) it is sufficient to find $\boldsymbol{\mathcal{B}}$ approximately, and

(ii) the individual columns of the matrix $\boldsymbol{\mathcal{B}}$ are in themselves not of much interest, only the vector $d\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{B}} \cdot \Delta\boldsymbol{\mathcal{W}}$ is required, where $\Delta\boldsymbol{\mathcal{W}}$ is a vector consisting of the $3N$ Gaussian noise coordinates ΔW_μ^s, with $\mu = 1, \ldots, N$, and $s = 1, 2, 3$. By assuming that $\boldsymbol{\mathcal{B}} = \sqrt{\boldsymbol{\mathcal{D}}}$, and using a Chebyshev polynomial approximation for the square root function, Fixman showed that (Fixman, 1986; Jendrejack et al., 2000; Kroger et al., 2000; Prabhakar, 2005)

$$d\boldsymbol{\mathcal{S}} = \sqrt{\boldsymbol{\mathcal{D}}} \cdot \Delta\boldsymbol{\mathcal{W}} \approx \sum_{p=0}^{N_{\mathrm{Ch}}-1} c_p \, \boldsymbol{\mathcal{V}}_p - \frac{c_0}{2} \Delta\boldsymbol{\mathcal{W}} \tag{5.19}$$

where N_{Ch} is the number of terms used in the Chebyshev polynomial approximation, c_p are the Chebyshev coefficients, and the vectors $\boldsymbol{\mathcal{V}}_p$ are defined by the recurrence relations

$$\boldsymbol{\mathcal{V}}_p = 2\boldsymbol{\mathcal{Y}} \cdot \boldsymbol{\mathcal{V}}_{p-1} - \boldsymbol{\mathcal{V}}_{p-2} \, ; \quad p \geq 2 \tag{5.20}$$

with $\boldsymbol{\mathcal{V}}_0 = \Delta\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{V}}_1 = \boldsymbol{\mathcal{Y}} \cdot \boldsymbol{\mathcal{V}}_0$. The linear mapping

$$\boldsymbol{\mathcal{Y}} = \left(\frac{2}{d_{\mathrm{max}} - d_{\mathrm{min}}} \right) \boldsymbol{\mathcal{D}} - \left(\frac{d_{\mathrm{max}} + d_{\mathrm{min}}}{d_{\mathrm{max}} - d_{\mathrm{min}}} \right) \boldsymbol{\mathcal{I}} \tag{5.21}$$

(where $\boldsymbol{\mathcal{I}}$ denotes the $3N \times 3N$-dimensional identity matrix), ensures that the eigenvalues of $\boldsymbol{\mathcal{Y}}$ lie in the domain $[-1, 1]$ when the eigenvalues of $\boldsymbol{\mathcal{D}}$ are within $[d_{\mathrm{min}}, d_{\mathrm{max}}]$. This is essential for the validity of the Chebyshev approximation.

It is clear that for a given N_{Ch}, the cost of the direct calculation of the $3N$-dimensional vector $d\boldsymbol{\mathcal{S}}$, without the intermediate calculation of $\boldsymbol{\mathcal{B}}$, is proportional to $N_{\mathrm{Ch}} \times$ [the cost of evaluating $\boldsymbol{\mathcal{V}}_p$]. The number of terms N_{Ch} that are required is determined by the desired accuracy in the estimation of the square root. The choice of N_{Ch} is also affected by the necessity of ensuring that the bounds d_{max} and d_{min} satisfy the following constraints relative to the maximum (λ_{max}) and minimum (λ_{min}) eigenvalues of $\boldsymbol{\mathcal{D}}$, namely, $d_{\text{max}} \geq \lambda_{\text{max}}$ and $d_{\text{min}} \leq \lambda_{\text{min}}$.

The CPU cost involved in adopting Fixman's procedure for dilute polymer solutions has been discussed in depth in Jendrejack et al. (2000); Kroger et al. (2000) and Prabhakar (2005). Here, we briefly summarize the main conclusions: (i) The cost of evaluating $\boldsymbol{\mathcal{V}}_p$ is simply $O(N^2)$. (ii) The number of terms in the Chebyshev approximation is determined using the expression (Kroger et al., 2000; Prabhakar and Prakash, 2004)

$$N_{\text{Ch}} = \text{nint}\left[\left(\frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}\right)^{\frac{1}{2}}\right] + 1 \tag{5.22}$$

where nint is the nearest integer function. The use of Eq. (5.22) is motivated by the finding (Fixman, 1986; Jendrejack et al., 2000) that the value of N_{Ch} required to keep the error in the estimation of the square root below a fixed tolerance, scales as $(\lambda_{\text{max}}/\lambda_{\text{min}})^{\frac{1}{2}}$. (iii) The limiting eigenvalues λ_{max} and λ_{min} can be calculated exactly in $O(N^2)$ operations using standard software packages—this procedure was adopted in Jendrejack et al. (2000) with the package ARPACK. On the other hand, Kroger et al. (2000) and Prabhakar and Prakash (2004) avoid explicit evaluation of the eigenvalues, but instead obtain approximate estimates for λ_{max} and λ_{min}. In particular, Prabhakar and Prakash (2004) use the following expressions based on a suggestion by Fixman (1986)

$$\lambda_{\text{max}}^{\text{Fixman}} = \frac{1}{3N}\left(\boldsymbol{\mathcal{U}}^{+} \cdot \boldsymbol{\mathcal{D}} \cdot \boldsymbol{\mathcal{U}}^{+}\right) \quad \text{and} \quad \lambda_{\text{min}}^{\text{Fixman}} = \frac{1}{3N}\left(\boldsymbol{\mathcal{U}}^{-} \cdot \boldsymbol{\mathcal{D}} \cdot \boldsymbol{\mathcal{U}}^{-}\right) \tag{5.23}$$

where $\boldsymbol{\mathcal{U}}^{+}$ is a $3N$-dimensional vector, all of whose elements are equal to 1 and $\boldsymbol{\mathcal{U}}^{-}$ is a $3N$-dimensional vector with alternating 1's and -1's. Further, the bounds $d_{\text{max}} = 2\lambda_{\text{max}}^{\text{Fixman}}$ and $d_{\text{min}} = 0.5\lambda_{\text{min}}^{\text{Fixman}}$ were chosen to satisfy the conditions on the magnitudes of d_{max} and d_{min} relative to the maximum and minimum eigenvalues. Since for dilute polymer solutions, $(\lambda_{\text{max}}/\lambda_{\text{min}}) \sim N^{0.5}$ and consequently $N_{\text{Ch}} \sim$

$N^{1/4}$, the CPU time requirement for the calculation of $d\boldsymbol{\mathcal{S}}$ in Fixman's method scales as $N_{\text{Ch}}N^2 \sim N^{9/4}$.

In the case of semidilute polymer solutions, since determining $\boldsymbol{\mathcal{V}}_p$ requires the recursive evaluation of the product of a linear transformation of the diffusion tensor with various $3N$-dimensional vectors (see Eq. (5.20)), the Ewald sum can be used for its evaluation, with the force vector $\mathbf{F}_\mu$ in Eq. (5.1) replaced by the relevant vector in the Chebyshev recursive relationship Eq. (5.20). Thus the cost of evaluating $\boldsymbol{\mathcal{V}}_p$ is identical to the cost of carrying out the Ewald sum. With the optimization introduced here, this would imply a cost that scales as $O(N^{1.5})$. The issues of determining the number of terms N_{Ch}, and the maximum and minimum eigenvalues of $\boldsymbol{\mathcal{D}}$, must also be addressed before the total cost of Fixman's procedure in the context of semidilute solutions can be estimated.

As pointed out earlier, it is not necessary to know the diffusion matrix $\boldsymbol{\mathcal{D}}$ explicitly in order to describe the conformational evolution of polymer molecules in a semidilute solution. However, since Beenakker (1986) provides an expression for the periodic diffusion tensor $\mathbf{D}_{\nu\mu}$ in his original derivation, it can used to determine the exact values of the maximum and minimum eigenvalues, denoted here by $\lambda_{\text{max}}^{\text{exact}}$ and $\lambda_{\text{min}}^{\text{exact}}$. By comparing the values given by Eq. (5.23) with the exact values (obtained with the gsl_eigen_symm subroutine of the GNU Scientific Library), we find that the behavior for our semidilute system is quite different from what is known for single-chain simulations: While in the latter case, $\lambda_{\text{max}}^{\text{Fixman}}$ is a reasonable approximation to $\lambda_{\text{max}}^{\text{exact}}$ (meaning that it scales in the same way with the number of beads, with a constant ratio of order unity), we here find that $\lambda_{\text{max}}^{\text{Fixman}}$ is essentially independent of N, while $\lambda_{\text{max}}^{\text{exact}}$ increases with N, roughly like $N^{0.6}$. One such example for these scalings are illustrated in Fig. 5.7 (a). In other words, Eq. (5.23) provides only a poor approximation. The reason why the behavior is so different for dilute and semidilute systems is not clear to us;

we speculate that it might have to do with the different density distributions of segments. Nevertheless, we can still use $\lambda_{\max}^{\text{Fixman}}$ for estimating the maximum eigenvalue, since we empirically find, for a range of values of $c/c^\star$, a, N_b and N_c, and for a variety of polymer conformations, the relation

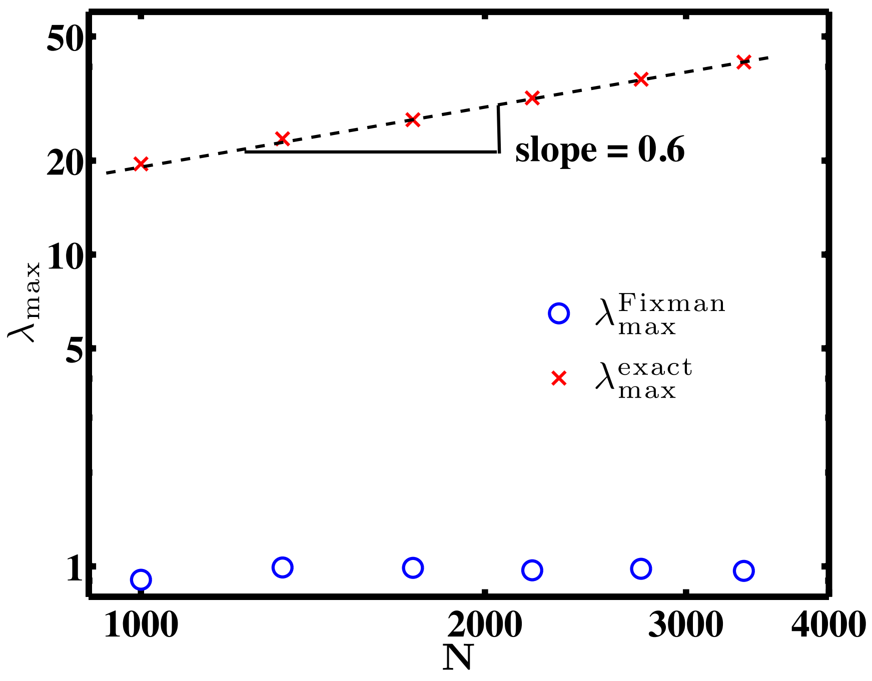
$$\lambda_{\max}^{\text{exact}} = 0.35\, N^{0.6}\, \lambda_{\max}^{\text{Fixman}} \tag{5.24}$$

which is therefore used to estimate $\lambda_{\max}$, assuming that it is valid throughout. An example showing this scaling is plotted in Fig. 5.7 (b). Similarly, we find empirically that the lowest exact eigenvalue is essentially independent of the number of segments, i.e.

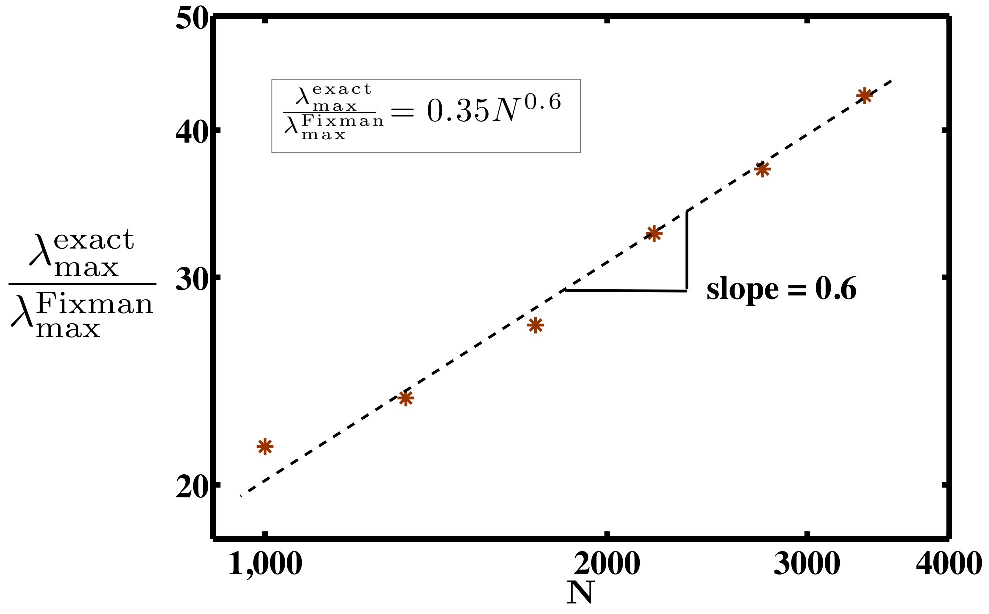
$$\frac{\lambda_{\max}^{\text{exact}}}{\lambda_{\min}^{\text{exact}}} = C\, N^{0.6} \tag{5.25}$$

where the prefactor C depends on the values of $c/c^\star$, a and N_b, increasing with an increase in a and N_b and decreasing with an increase in $c/c^\star$. For instance, for $c = 4.44c^\star$, $a = 0.5$ and $N_b = 10$, we find $C = 8$. The scaling of $\lambda_{\max}^{\text{exact}}/\lambda_{\min}^{\text{exact}}$ is shown in Fig. 5.8. It follows that in the course of simulations a fairly accurate estimate of the minimum eigenvalue can be obtained, once $\lambda_{\max}$ is determined, by using the expression $\lambda_{\min} = \lambda_{\max}/(C\, N^{0.6})$. In general, the value of C is obtained by trial and error. Once $\lambda_{\max}$ and $\lambda_{\min}$ are determined by this procedure, we find that it is adequate to use the bounds $d_{\max} = \lambda_{\max}$ and $d_{\min} = \lambda_{\min}$ to ensure a robust implementation of the Chebyshev polynomial approximation.

With regard to the number of Chebyshev terms, we find that for semidilute solutions, as in the case of dilute solutions, the value of N_{Ch} required to keep the error in the estimation of the square root below a fixed tolerance, scales as $(\lambda_{\max}/\lambda_{\min})^{\frac{1}{2}}$. This immediately suggests from Eq. (5.25) that the CPU time requirement for the calculation of $d\boldsymbol{\mathcal{S}}$ for semidilute solutions using Fixman's method scales as $N_{\text{Ch}}N^{1.5} \sim N^{1.8}$. Equation (5.22) is used here to provide an



(a)



(b)

Figure 5.7: Scaling of maximum eigen values with N for $c = 4.44c^{\star}$, $a = 0.5$ and $N_b = 10$ (a) Actual and Fixman maximum eigenvalues (b) Ratio of actual to Fixman maximum eigenvalues as a function of N

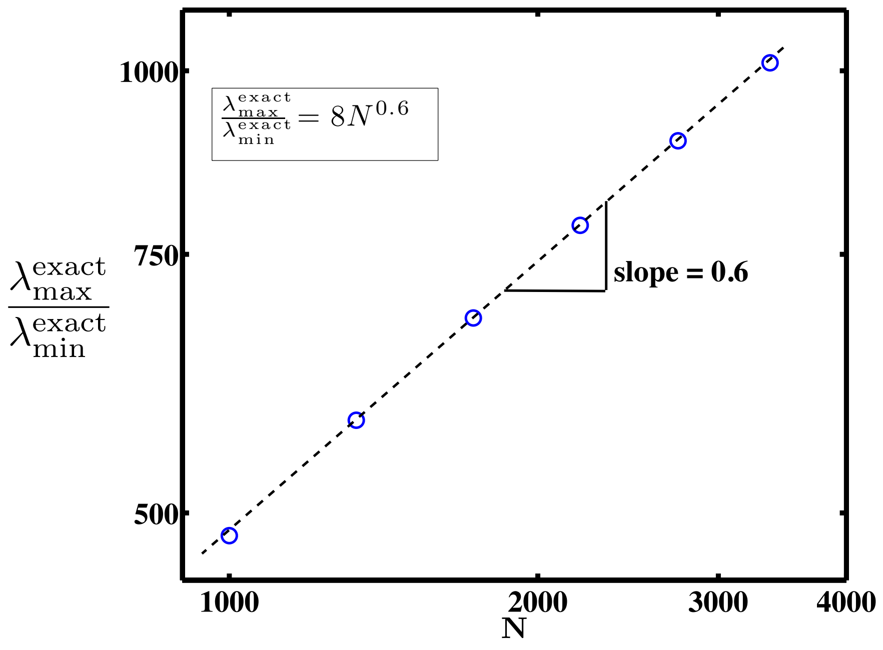


Figure 5.8: Ratio of actual maximum to minimum eigenvalues of the diffusion tensor

initial guess for N_{Ch}, which is then incrementally increased until the relative error E_f in the estimation of the square root function, given by the following expression suggested in Jendrejack et al. (2000),

$$E_f = \left(\frac{|\,(\boldsymbol{\mathcal{B}} \cdot \Delta\boldsymbol{\mathcal{W}}) \cdot (\boldsymbol{\mathcal{B}} \cdot \Delta\boldsymbol{\mathcal{W}}) - \Delta\boldsymbol{\mathcal{W}} \cdot \boldsymbol{\mathcal{D}} \cdot \Delta\boldsymbol{\mathcal{W}}\,|}{\Delta\boldsymbol{\mathcal{W}} \cdot \boldsymbol{\mathcal{D}} \cdot \Delta\boldsymbol{\mathcal{W}}} \right)^{1/2} \tag{5.26}$$

is less than a specified tolerance. In practice we find that the choice of C affects the efficiency with which the final value of N_{Ch} is obtained.

5.6 Optimization of Each Euler Time Step

The implementation of the Euler algorithm used here to determine the configurational evolution of the system requires the estimation of the drift term $\sum_{\mu} \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_{\mu}$

and the diffusion term $\sum_{\mu} \mathsf{B}_{\nu\mu} \cdot \mathbf{\Delta W}_{\mu}$ *at each time step*, since the algorithm proceeds by evaluating the right hand side of Eq. (4.1) for each bead ν in the original simulation box. As mentioned earlier, determining the latter sum involves the repeated invocation of the Ewald sum. Since the spatial configuration of the system is frozen in a single time step, all terms in the Ewald sum that are either (i) constant, (ii) only dependent on the reciprocal space vector $\mathbf{k}$, or (iii) only dependent on the spatial configuration, do not have to be repeatedly evaluated. As a result, it becomes necessary not only to discuss the optimal evaluation and scaling with system size of the drift and diffusion terms individually (as we have in Sections 5.4 and 5.5), but also to consider the overall optimization of each Euler time step.

It turns out that there are two ways in which this optimization can be carried out. In order to give a flavor of the issues involved, we only discuss here the treatment of the term involving $\mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})$ in the Ewald sum (see Eq. (5.8)). The remaining terms are either treated similarly, or entail a more straightforward treatment. Clearly, the term involving $\mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})$ is first evaluated when the drift term $\sum_{\mu} \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_{\mu}$ is evaluated. Subsequently, it is required that each time the term $\boldsymbol{\mathcal{D}} \cdot \boldsymbol{\mathcal{V}}_p$; $p = 0, \ldots, N_{\text{Ch}} - 1$ is evaluated in the Chebyshev polynomial approximation (see Eqs. (5.20) and (5.21)). For ease of discussion, we denote by $\mathcal{V}^s_{\mu}$ the $3N$ components of a typical vector $\boldsymbol{\mathcal{V}}_p$. Then the term involving $\mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})$ in the implementation of the Chebyshev polynomial approximation can be written as $\sum'_{\mathbf{n}} \sum_{\mu=1}^{N} \sum_{s=1}^{3} M^{rs}_{\nu\mu,\mathbf{n}} \mathcal{V}^s_{\mu}$, where $M^{rs}_{\nu\mu,\mathbf{n}}$ represents the 'rs'th Cartesian component of the tensor $\mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})$. Before discussing the two methods of optimization used here, it is worth noting the following point that is common to both methods. For each bead ν, in any periodic image $\mathbf{n}$, the sum over the index μ is carried out only over the nearest neighbors of bead ν, i.e., over the N_{r_c} particles that lie within a sphere centered at bead ν with cutoff radius r_{c}. The choice of r_{c}, however, is different in the two schemes, as detailed below.

In the first method of optimization, which we denote here as HMA (for "High Memory Algorithm"), the $3N \times 3N$ matrix $S^{rs}_{\nu\mu} = \sum'_{\mathbf{n}} M^{rs}_{\nu\mu,\mathbf{n}}$ is calculated once and for all and stored in the course of evaluating the drift term $\sum_{\mu} \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_{\mu}$. Note that the cost of evaluating $S^{rs}_{\nu\mu}$ scales as $O(N \times N_{\mathrm{r_c}})$ since in each periodic image $\mathbf{n}$, only the beads μ whose distance from bead ν is less than a cutoff radius $\left(r^{\mathrm{HMA}}_{\mathrm{c}}\right)_{\mathrm{opt}}$ are considered in the sum over all periodic images. The nature of $\left(r^{\mathrm{HMA}}_{\mathrm{c}}\right)_{\mathrm{opt}}$ and the value of $N_{\mathrm{r_c}}$ in this context, is discussed in more detail below. It should be noted that the matrix $S^{rs}_{\nu\mu}$ becomes increasingly sparse when the system size is increased. While it is therefore possible to save memory by sparse-matrix techniques (meaning in practice the construction of a Verlet table (Grest et al., 1989) and making use of indirect addressing), this was not attempted here, i.e. we stored the matrix with a simple $O(N^2)$ implementation. Subsequently, each time the term $\boldsymbol{\mathcal{D}} \cdot \boldsymbol{\mathcal{V}}_p\,;\, p = 0, \ldots, N_{\mathrm{Ch}} - 1$ is calculated in the Chebyshev polynomial approximation, the $O(N^2)$ matrix multiplication $\sum_{\mu=1}^{N} \sum_{s=1}^{3} S^{rs}_{\nu\mu} \mathcal{V}^{s}_{\mu}$ is carried out. Again, a sparse matrix implementation might be able to reduce this computational complexity. Ultimately this term dominates and the total CPU cost of this scheme scales as $O(N_{\mathrm{Ch}} \times N^2)$. For systems that are not sufficiently large, the CPU cost might lie in the crossover region between $O(N \times N_{\mathrm{r_c}})$ (the cost for the deterministic drift) and $O(N_{\mathrm{Ch}} \times N^2)$.

The reason that $\left(r^{\mathrm{HMA}}_{\mathrm{c}}\right)_{\mathrm{opt}}$ is different from the cutoff radius $\left(r^{\mathrm{E}}_{\mathrm{c}}\right)_{\mathrm{opt}}$ (calculated earlier for just the evaluation of the Ewald sum) is because in the HMA algorithm a different procedure is used in the repeated evaluation of the Ewald sum, with certain quantities being calculated once and for all and stored. By repeating the procedure adopted earlier for optimizing the bare Ewald sum, namely, by estimating the total time required to evaluate the various quantities in the real space and reciprocal space sums, we find for the CPU time for one step in nanoseconds

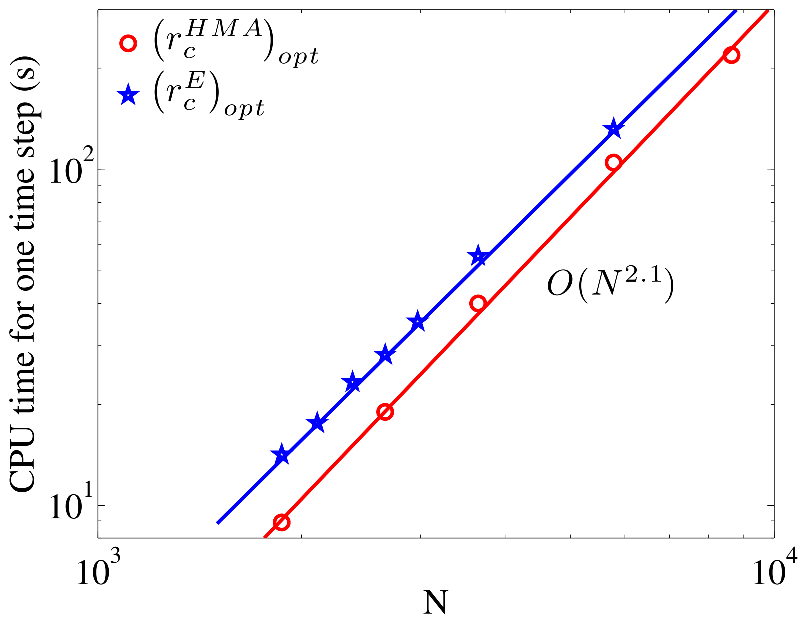
$$\begin{aligned} T^{\rm HMA}/ns &= 30N(N-1) + 1500N\, r_{\rm c}^3 c \\ &+ 23N^2(2+N_{\rm Ch}) + 0.67\,\frac{M^6}{r_{\rm c}^3 c} N^2\,(9.2 + 4.2N_{\rm Ch}) \end{aligned} \tag{5.27}$$

where the constants reflect the various execution times for individual terms on a 156 SGI Altix XE 320 cluster. Minimizing this with respect to the cutoff radius leads to

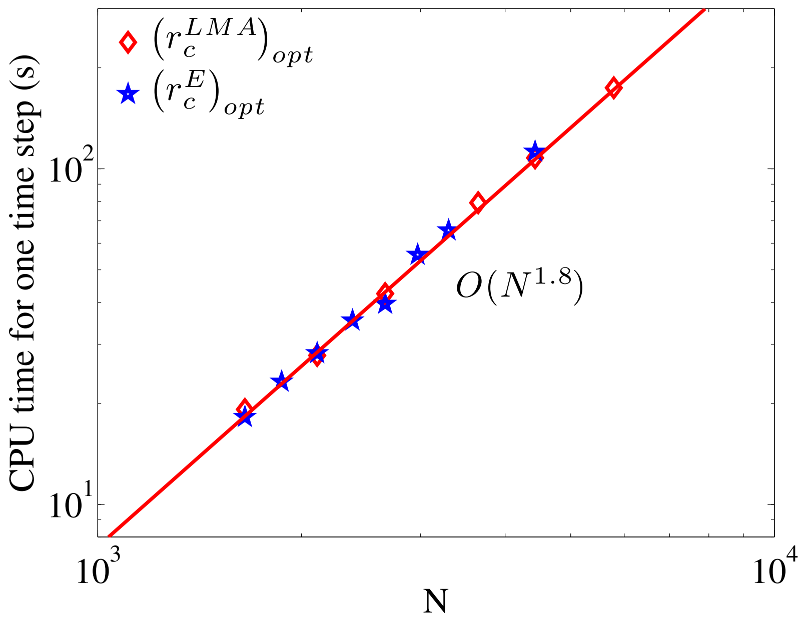
$$\left(r_{\rm c}^{\rm HMA}\right)_{\rm opt} = \frac{M}{2c^{1/3}}\,[0.25N \;+\; 0.12NN_{\rm Ch}]^{1/6} \tag{5.28}$$

It turns out that $\left(r_{\rm c}^{\rm HMA}\right)_{\rm opt} > \left(r_{\rm c}^{\rm E}\right)_{\rm opt}$. This is because a major part of the real space calculation of $\sum_{\mathbf{n}}' \mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})$ and $\mathsf{M}^{*}(\mathbf{r}_{\nu\mu,\mathbf{n}\,=\,\mathbf{0}})$ is not repeated $N_{\rm Ch}$ times in the HMA, leading to a cheaper real space implementation. As a result, the optimization procedure allows the HMA to attribute a greater computational load to the real space sum relative to the reciprocal space sum by having a larger cutoff radius than $\left(r_{\rm c}^{\rm E}\right)_{\rm opt}$. In contrast to the bare Ewald sum, where $N_{r_{\rm c}} \sim N^{0.5}$, we find empirically that in the HMA, $N_{r_{\rm c}} \sim N^{0.7}$. It is clear from Fig. 5.9 (a) that the CPU time for HMA scales as $O(N^{2.1})$ when the simulation is run at a cutoff radius of $\left(r_{\rm c}^{\rm HMA}\right)_{\rm opt}$, with the empirically estimated exponent 2.1 lying in the crossover regime discussed earlier. Fig. 5.9 (a) also indicates that the CPU cost is greater when $\left(r_{\rm c}^{\rm E}\right)_{\rm opt}$ is used in the HMA, confirming the necessity to optimize the total procedure for evaluating a single Euler time step rather than using the cutoff radius obtained from the Ewald sum optimization.

The major difference from the HMA, in the alternative method of optimization used here (denoted by LMA for "Low Memory Algorithm"), is the treatment of the sum $\sum_{\mathbf{n}}' \sum_{\mu=1}^{N} \sum_{s=1}^{3} M_{\nu\mu,\mathbf{n}}^{rs} \mathcal{V}_{\mu}^{s}$. While many quantities, such as those that are constant, or only functions of the reciprocal space vector $\mathbf{k}$, are still calculated and stored once and for all when the drift term $\sum_{\mu} \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_{\mu}$ is evaluated,



(a)



(b)

Figure 5.9: CPU time scalings for (a) HMA (b) LMA. Symbols represent simulations results, and the lines are analytical estimates for the total time.

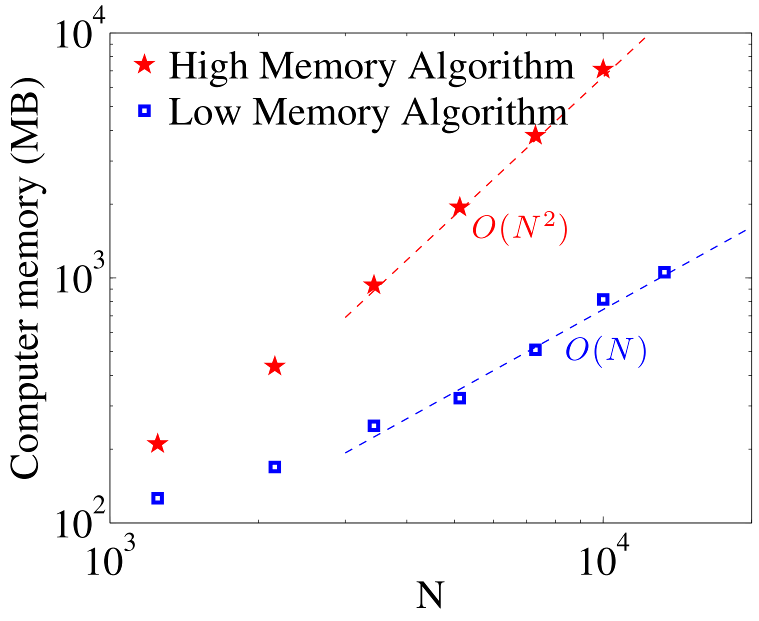
the $3N \times 3N$ matrix $S^{rs}_{\nu\mu}$ is not stored. Instead, the following two steps are carried out: (i) For a given bead index ν and periodic image $\mathbf{n}$, the quantity $T^{r}_{\nu,\mathbf{n}} = \sum_{\mu=1}^{N} \sum_{s=1}^{3} M^{rs}_{\nu\mu,\mathbf{n}} \mathcal{V}^{s}_{\mu}$ is evaluated, ensuring that only those beads μ that lie within a cutoff radius $\left(r_{\rm c}^{\rm LMA}\right)_{\rm opt}$ of bead ν are considered in the sum over μ. Note that for each bead μ, the sum over s involves a simple $(3\times3)\times(3\times1)$ matrix multiplication. (ii) The sum $\sum_{\mathbf{n}}' T^{r}_{\nu,\mathbf{n}}$ over periodic images $\mathbf{n}$ is then performed to obtain the required quantity in the Ewald sum.

Since, even in the LMA, some quantities are stored during the evaluation of the drift term $\sum_{\mu} \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_{\mu}$, we can optimize the entire process involved in executing one time step in the Euler algorithm by choosing the appropriate cutoff radius. Adopting the procedure described earlier, we find for the CPU time per step in nanoseconds

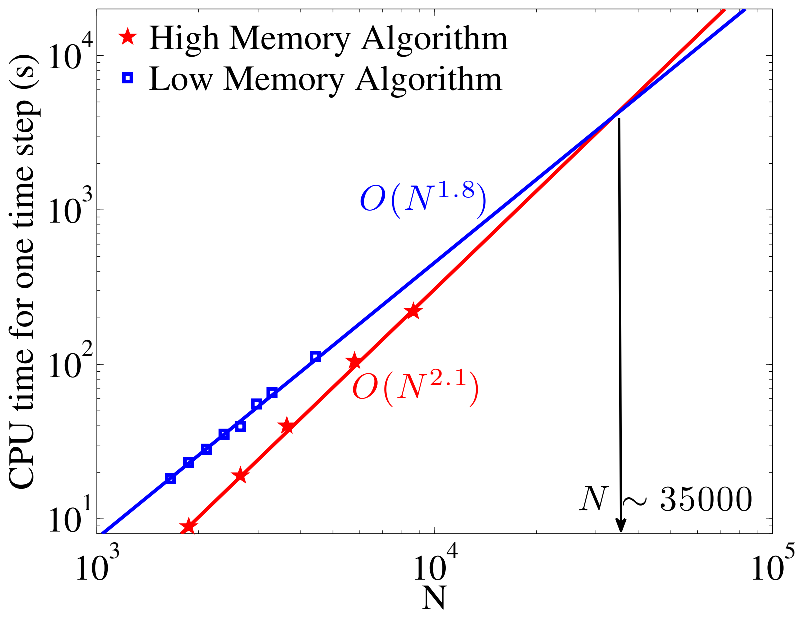
$$T^{\rm LMA}/ns = 1200\, r_{\rm c}^{3} c\, N (1 + N_{\rm Ch}) + 2\, \frac{M^{6}}{r_{\rm c}^{3} c} N^{2} \,(1.8 + 1.5 N_{\rm Ch}) \tag{5.29}$$

$$\left(r_{\rm c}^{\rm LMA}\right)_{\rm opt} = \frac{M}{2c^{1/3}} \left[\frac{0.18N \,+\, 0.15 N N_{\rm Ch}}{1 + N_{\rm Ch}} \right]^{1/6} \tag{5.30}$$

Fig. 5.9 (b) compares the CPU cost involved when either the cutoff radius $\left(r_{\rm c}^{\rm LMA}\right)_{\rm opt}$ or $\left(r_{\rm c}^{\rm E}\right)_{\rm opt}$ is used in the LMA. The reason that $\left(r_{\rm c}^{\rm LMA}\right)_{\rm opt}$ and $\left(r_{\rm c}^{\rm E}\right)_{\rm opt}$ are nearly equal to each other is because practically all the time consuming parts of the Ewald sum are calculated repeatedly $N_{\rm Ch}$ times in the LMA. As a result, in contrast to the HMA, the saving achieved by storing some quantities does not make a significant difference to the choice of cutoff radius. Unlike the HMA, there is no large storage requirement in the LMA, as shown in Fig. 5.10 (a), where it is seen to scale as $O(N)$ for sufficiently large N. Further, the CPU cost scales as $O(N_{\rm Ch} \times N \times N_{r_{\rm c}})$, which is identical to the scaling for the basic Ewald sum, namely, $O(N^{1.8})$ as can be seen in Fig. 5.10 (b). On the other hand, since $S^{rs}_{\nu\mu}$ is not stored, the components $M^{rs}_{\nu\mu,\mathbf{n}}$ are repeatedly evaluated in each



(a)



(b)

Figure 5.10: Comparison between HMA and LMA for (a) Computer memory requirement (b) CPU time required for a single time step computation

of the recursive Chebyshev calculations. This extra calculation leads to a large pre-factor in the scaling of the CPU with N. However, at $N \sim 35000$ a crossover in the CPU cost can be seen to occur, suggesting that it is advisable to use HMA below a system size of roughly 35000, while the LMA would be cheaper for larger systems.

5.7 Testing and Verification of the Algorithm

The optimized BD algorithm developed here is validated by testing and verification under both θ solvent and good solvent conditions. In the former case, we first check to see if static equilibrium properties, namely, the gyration radius and the end-to-end vector, agree with known analytical results. Secondly, the current implementation of the Ewald sum for hydrodynamic interactions (which enables its use even in simulations that do not incorporate excluded volume interactions) is tested by comparing the prediction of the infinite dilution equilibrium self-diffusion coefficient, which is a dynamic property, with the results of a BD simulation of single chain dynamics. As mentioned in Section 5.1, Pham et al. (2009) have recently quantitatively compared the predictions of the explicit solvent LB/MD method with the predictions of the implicit solvent BD method for the dynamics of a single chain under good solvent conditions in the dilute limit. A natural follow up of the development of the current BD algorithm is to compare the two methods at finite concentrations under good solvent conditions. Here we carry out our study by comparing the predictions of the radius of gyration, the end-to-end vector, and the self-diffusion coefficient. This serves both to verify the predictions of the current algorithm in a regime where there are no analytical predictions, and to obtain an estimate of the relative computational costs of the two mesoscopic simulation methods in the semidilute regime.

The mean-square end-to-end distance is given by

$$\langle R_e^2 \rangle = \langle (\mathbf{r}_{N_b} - \mathbf{r}_1)^2 \rangle \tag{5.31}$$

while the mean-square radius of gyration is given by

$$\langle R_g^2 \rangle = \frac{1}{2N_b^2} \sum_{\mu=1}^{N_b} \sum_{\nu=1}^{N_b} \langle r_{\mu\nu}^2 \rangle \tag{5.32}$$

with $r_{\mu\nu} = |\mathbf{r}_\nu - \mathbf{r}_\mu|$ being the inter-particle distance. The long-time self-diffusion coefficient is calculated by tracking the mean-square displacement of the centre of mass $\mathbf{r}_c$ of each chain

$$D = \lim_{t\to\infty} \left\langle \frac{|\mathbf{r}_c(t) - \mathbf{r}_c(0)|^2}{6t} \right\rangle \tag{5.33}$$

The predictions of the gyration radius, the end-to-end vector, and the self-diffusion coefficient by the current algorithm under θ solvent and good solvent conditions, and their verification by various means, are discussed in turn below.

5.7.1 θ solvents

The mean-square end-to-end distance and the mean-square radius of gyration at equilibrium were obtained by carrying out simulations of bead-spring chains with Hookean springs, using $N_b = 20$ and 40 and a fixed number of chains $N_c = 20$. The nondimensional bead radius a was chosen to be 0.5, and a time step $\Delta t = 0.01$ was used to carry out the Euler integration. A range of concentrations from $3 \times 10^{-4}\, c^\star$ to $3\, c^\star$ were considered, with the concentration being varied by changing the size of the simulation box L. Since the chains are free to cross each other for θ solvents, static properties such as the end-to-end distance and the gyration radius are independent of concentration. Further, as is well known, their dependence on N_b can be shown analytically to be Bird et al. (1987)

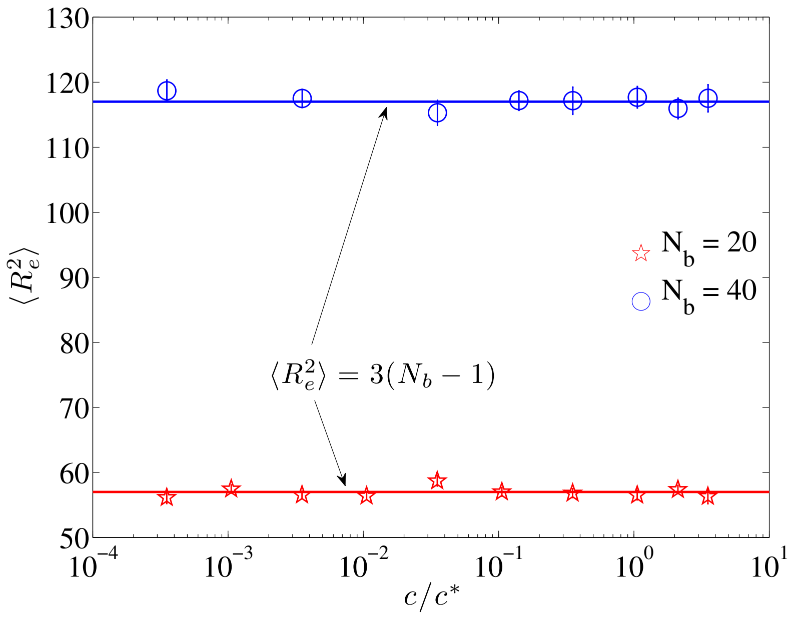
$$\langle R_e^2 \rangle = 3(N_b - 1) \tag{5.34}$$

and

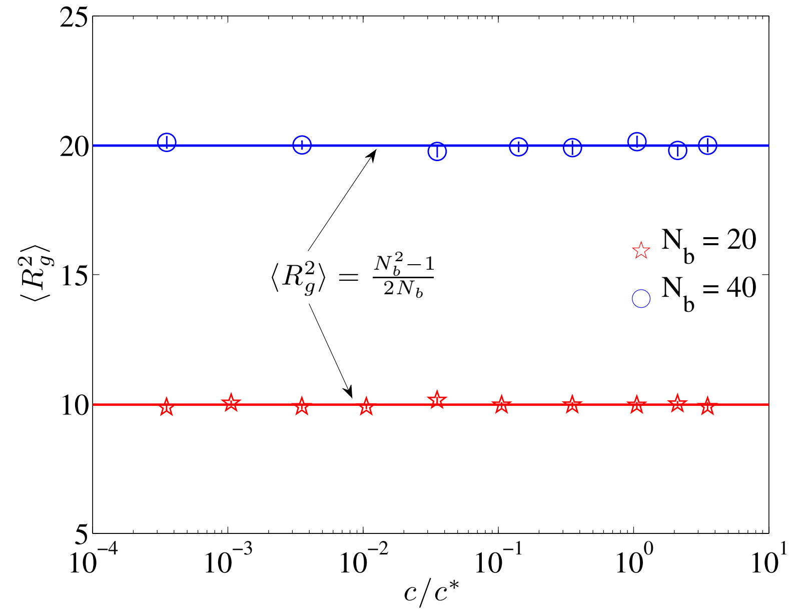
$$\langle R_g^2 \rangle = \frac{N_b^2 - 1}{2N_b} \tag{5.35}$$

Note that $c^\star$ can be determined once a choice for N_b is made. Figs. 5.11 (a) and (b) display the results for $\langle R_e^2 \rangle$ and $\langle R_g^2 \rangle$, respectively, for the range of concentrations considered here. Symbols indicate simulation data, while solid lines represent the analytical results given by Eqs. (5.34) and (5.35). Clearly, the simulated static properties are in good agreement with analytical predictions.

The single chain diffusion coefficient in a dilute solution under θ solvent conditions is used here as the benchmark for verifying the current implementation of the Ewald sum. The value of the diffusion coefficient for $N_b = 20$ and $a = 0.5$ is displayed as the solid line in Fig. 5.12, obtained here by a conventional BD simulation algorithm that uses a semi-implicit predictor corrector scheme developed in our group for simulating a single chain that is not confined in a box (Prabhakar and Prakash, 2004). For the same set of input simulation parameters, the long-time diffusivity is obtained from the current multi-chain BD algorithm for a range of concentrations. It is clear from Fig. 5.12 that the simulated data (symbols) for the diffusivity D approaches the single-chain result in the limit of zero concentration. The value of D at $c/c^\star = 0$ was obtained by fitting the values at $c/c^\star = 0.001, 0.003$ and 0.01 with a second order polynomial and extrapolating to zero concentration.



(a)



(b)

Figure 5.11: Validation of static properties under θ-conditions: (a) Mean-square end-to-end distance $\langle R_e^2 \rangle$ (b) Mean-square gyration radius $\langle R_g^2 \rangle$. Symbols indicate simulation data, while solid lines represent the analytical results given by Eqs. (5.34) and (5.35).

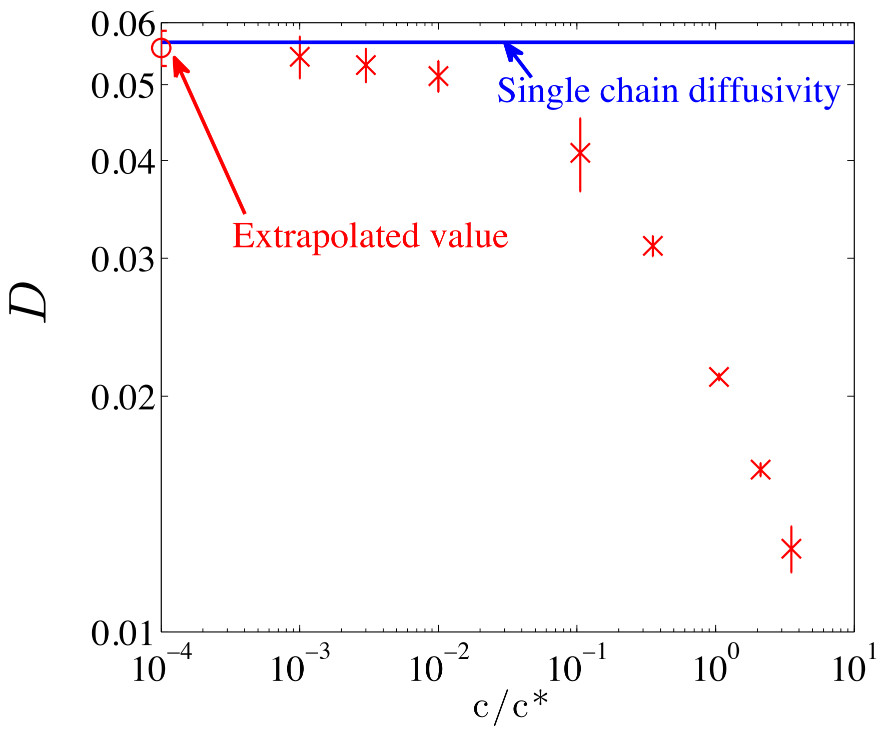


Figure 5.12: Long-time self-diffusion coefficient under θ solvent conditions. Symbols indicate simulation data obtained with the current multi-chain algorithm, while the solid line represents the value obtained by simulating the dynamics of a single chain in a dilute solution. The circle symbol on the y-axis is the value obtained by extrapolating the finite concentration results to the limit of zero concentration.

5.7.2 Good solvents

In order to carry out a quantitative comparison between the LB/MD and BD methods, it is necessary to ensure that the underlying polymer model is identical for both the methods, and to map the input parameters of the hybrid model onto the input values of the BD model. A detailed discussion of how this can be achieved in the context of dilute solutions has been given in Pham et al. (2009). Exactly the same procedure has been adopted here as discussed in Appendix E. In particular, Appendix E discusses (i) the mapping of different simulation parameters between BD and LB/MD, and (ii) the procedure to run LB/MD simulations using an open source software ESPResSo (Extensible Simulation Package for Research on Soft matter). Essentially, a bead-spring chain with FENE springs is used, with a Weeks-Chandler-Andersen potential, which acts between all monomers, employed to model the excluded volume (EV) effect. While in the LB/MD simulation approach, the Weeks-Chandler-Andersen parameters are used to define the units of energy, length, and time, the corresponding units in the BD simulations have been discussed earlier in Section 4.2. Appendix E discusses the details of the length and time unit conversions between the two methods. The comparison of the two methods proceeds by first picking the simulation parameters for the LB/MD model, using these for the LB/MD simulations, then converting them to BD units using the procedure outlined in Pham et al. (2009) and Appendix E, and finally running the equivalent BD model obtained in this manner. In other words, the two units systems are maintained in the respective methods, and a comparison of predicted quantities carried out *a posteriori*.

The results of carrying out this procedure for $\langle R_e^2 \rangle$, $\langle R_g^2 \rangle$ and D are shown in Table 5.1 for $N_b = 10$ at three different concentrations. It is worth noting that, since EV interactions are short-ranged, we have implemented a neighbor-list in the current BD algorithm for computing the pairwise summation of EV interactions, with a cutoff radius equal to the range of the Weeks-Chandler-Andersen potential. All values are reported in BD units, unless specified otherwise.

N_c	Method	L	c	$c/c^\star$	$\langle R_e^2 \rangle$	$\langle R_g^2 \rangle$	D
20	BD	24.152	0.0142	0.546±0.001	111.37±0.47	18.36±0.04	$0.0272\pm6\times10^{-4}$
	LB	10	0.2	0.543±0.005	112.93±0.27	18.29±0.02	$0.0268\pm8\times10^{-4}$
32	BD	21.737	0.0311	1.199±0.002	98.35±0.43	16.6±0.04	$0.0162\pm6\times10^{-4}$
	LB	9	0.439	1.192±0.011	99.11±0.36	16.59±0.03	$0.0151\pm4\times10^{-4}$
70	BD	21.737	0.068	2.623±0.004	76.07±0.53	13.33±0.05	$0.0024\pm1\ \times10^{-4}$
	LB	9	0.96	2.607±0.025	77.29±0.35	13.42±0.04	$0.00245\pm5\times10^{-5}$

Table 5.1: Comparison of predictions of the radius of gyration, the end-to-end vector, and the self-diffusion coefficient by the explicit solvent LB/MD method with the predictions of the implicit solvent BD method, for a bead-spring chain with $N_b = 10$ at three different concentrations, in a good solvent. Note that all properties are given in BD units, except the box size L, and concentration c, which are given in LB units when reported for the LB/MD simulations. Both L and c are identical in both methods when reported in the same unit system. Note that the highest concentration corresponds to melt-like conditions.

We find it convenient to maintain the same absolute concentration in the two methods rather than the same $c/c^\star$, as this would entail an interpolation procedure. In the BD method, $c^\star$ is determined from $\langle R_g^2 \rangle$ in the dilute limit, by carrying out a single-chain simulation for parameter values that are identical to those in the multi-chain BD simulation. In the LB/MD method, simulations are carried out for three box sizes, $L = 12$, 17 and 21, with the number of monomers held fixed (we use $N_c = 20$ and $N_b = 10$). As a result, the monomer concentration decreases with increasing box size. The values of $\langle R_g^2 \rangle$ obtained for these three box sizes are extrapolated to infinite box size in order to determine $\langle R_g^2 \rangle$ (and consequently $c^\star$) in the dilute limit.

It is clear from Table 5.1, both in the dilute limit, with regard to values of $c/c^\star$, and at all three finite concentrations, with regard to values of $\langle R_e^2 \rangle$, $\langle R_g^2 \rangle$ and D, that there is excellent agreement between the two mesoscopic simulation methods, since all properties agree with each other within error bars. This validates the current algorithm in a regime where there are no analytical solutions. Further, it demonstrates the robustness of the parameter mapping technique developed by Pham et al. (2009) for comparing the two simulation methods.

5.8 Comparison of Computational Cost with LB/MD

Recent comparison by Pham et al. (2009) of the predictions of the explicit solvent LB/MD method with the predictions of the implicit solvent BD method for the dynamics of a single chain indicated that in the dilute limit, BD is the method of choice as it is significantly more efficient than LB/MD. However, Fig. 5.13 suggests that for our current implementation the situation is quite the reverse at the finite concentration, $c/c^\star = 1.2$, at which the simulation data in the figure were obtained. The comparison of the two mesoscopic simulation methods displayed in Fig. 5.13 was carried out using the identical procedure developed earlier by Pham et al. (2009). Essentially, the LB/MD method was run for a total of 100

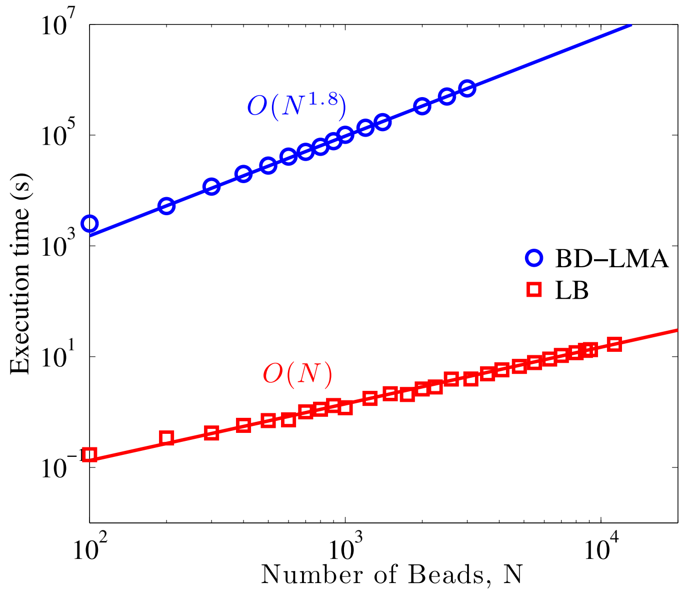


Figure 5.13: Comparison of the CPU time required by the LB and BD systems for a wide range of system sizes N, at concentration $c/c^{\star} = 1.2$, for the equivalent of one LB time unit.

MD time steps (with a step size of 0.01 in LB units, or 0.018 in BD units). This amounts to a total simulation time of one time unit in terms of LB units. The BD algorithm was then run for the same length of physical time, by converting one LB time unit to BD time units. The BD algorithm required a significantly smaller time step of 10^{-4} in BD units. The reason for this choice is because the current implementation uses a simple Euler integration scheme, with a rejection algorithm that ensures that none of the springs in any of the bead-spring chains exceeds the upper limit of the FENE spring length $\sqrt{b_p}$. In contrast, the earlier comparison of the two methods in the dilute limit was based on a BD code that uses a semi-implicit predictor-corrector method, enabling the use of a much larger step size of 5×10^{-3} BD units. The dependence of CPU time on system size was examined here by increasing the number of chains N_c, while keeping the number of beads in a chain fixed at $N_b = 10$. The concentration was maintained constant at $c/c^\star = 1.2$ (or $c = 0.031$ in BD units) by increasing the box size L suitably. Since the difference between the HMA and LMA BD algorithms is insignificant on the scale of the difference between LB/MD and BD, only results for the LMA are shown in Fig. 5.13.

The CPU time scaling of the LMA algorithm has been established in Section 5.6 to be $N^{1.8}$. From Eqs. (5.29) and (5.30) one immediately sees that after optimization the CPU time depends only on the particle number N, but is independent of the concentration c (or the system volume V):

$$T^{\mathrm{LMA}}(N, V) = \gamma^{\mathrm{LMA}} N^{1.8} \tag{5.36}$$

with some proportionality constant γ^{LMA}. Conversely, the LB/MD method is dominated by the CPU effort of the solvent, i.e.

$$T^{\mathrm{LB}}(N, V) = \gamma^{\mathrm{LB}} V = \gamma^{\mathrm{LB}} \frac{N}{c} \tag{5.37}$$

with another constant γ^{LB}. Hence

$$\frac{T^{\mathrm{LMA}}}{T^{\mathrm{LB}}} = \frac{\gamma^{\mathrm{LMA}}}{\gamma^{\mathrm{LB}}} c N^{0.8} \tag{5.38}$$

From our CPU timings we find a value $\gamma^{\mathrm{LMA}}/\gamma^{\mathrm{LB}} = 1.3 \times 10^4$ in BD units, i.e. our current implementation of Ewald BD becomes competitive with LB/MD only if the concentration is below the very small value $7.8 \times 10^{-5} \times N^{-0.8}$.

However, it should be noted that the present version is by far not the fastest conceivable BD code. Firstly, we expect that by implementing an implicit integrator the time step may be increased by nearly two orders of magnitude. Secondly, the evaluation of the real space HI should be substantially faster (both in the LMA and HMA versions) by making use of Verlet tables. Thirdly, the HMA algorithm could then take advantage of sparse-matrix techniques (see also the discussion in Section 5.6). Finally, the evaluation of the Fourier part can be speeded up by making use of Fast Fourier Transformation, which, as shown previously, gives rise to a complexity of the total algorithm of $O(N^{1.3} \log N)$ (Sierou and Brady, 2001; Banchio and Brady, 2003). All together, achieving accelerations by up to three orders of magnitude does not seem unrealistic.

5.9 Conclusions

A range of issues related to the development of an optimized BD algorithm for simulating the dynamics of semidilute solutions in unbounded domains has been considered here. In particular:

1. It is possible to develop an optimized Ewald method for hydrodynamic interactions that splits the cost of evaluating the real space and reciprocal space sums evenly, leading to a CPU cost that scales as $N^{1.5}$, rather than the N^2 scaling that would result from a straightforward implementation.

2. While Beenakker's original implementation of the Ewald sum is only valid for systems without bead overlap, it can be modified to account for bead overlap, such that θ-solutions can be simulated by switching off excluded volume interactions.

3. As in the case of dilute solutions, the number of Chebyshev terms required to maintain a desired accuracy scales as $(\lambda_{\max}/\lambda_{\min})^{\frac{1}{2}}$, where $\lambda_{\max}$ and $\lambda_{\min}$

are the maximum and minimum eigenvalues of the diffusion tensor $\boldsymbol{\mathcal{D}}$. It is shown that this leads to an additional computational load that scales as $N^{0.3}$.

4. It is necessary to consider the optimization of the overall time required to perform one Euler time step, in addition to the individual optimizations of the Ewald sum and Chebyshev polynomial approximation. In this context, two different schemes for optimization have been proposed in the form of the "high memory" (HMA), and the "low memory" (LMA) algorithms. While the LMA leads to an overall CPU time scaling of $N^{1.8}$, which appears better than the $N^{2.1}$ scaling of the HMA, the large prefactor in the former makes it preferable only for large systems with more than roughly 35,000 particles.

5. The optimized BD algorithm gives accurate predictions under both θ and good solvent conditions. In the latter case, BD predictions are compared with those of the LB/MD method. The parameter mapping scheme developed by Pham et al. (2009) for dilute solutions is found to be valid and useful even at a finite concentration in the semidilute regime.

6. In contrast to dilute solutions, where BD was shown to be significantly more computationally efficient than LB/MD (Pham et al., 2009), exactly the opposite is true for semidilute solutions. The CPU cost of the BD method scales as $N^{1.8}$, while the cost of the LB/MD method scales linearly with system size. The necessity of carrying out an Ewald sum renders the BD method developed here significantly more computationally expensive than LB/MD. Nevertheless, it should be noted that the BD method can be further refined and dramatically speeded up, as discussed at the end of Section 5.8.

Chapter 6

Dynamic Crossover Scaling in Semidilute Polymer Solutions

6.1 Introduction

This chapter aims to verify the scaling arguments developed in Chapter 3, using the Brownian dynamics simulation algorithm discussed in Chapter 5. Broadly, this chapter attempts to validate the two important conclusions of the scaling analysis with the help of computer simulations:

1. The existence of universal crossover scaling functions in the solvent quality driven crossover region.

2. The suggestion that there is only a single crossover scaling function from which others can be inferred.

These conclusions can be validated by carrying out simulations for the different crossover scaling functions such as ϕ_R and ϕ_D for a range of $c/c^\star$ and z. In order to perform simulations for these crossover scaling functions for semidilute polymer solutions and verify their universality, it is necessary to consider sufficiently long chains. However, due to the limitation in the computational speed of the current algorithm, it is not possible to perform simulations for very long chains. A discussion of the CPU costs involved in such simulations is given in Appendix C. Nonetheless, simulations for smaller chains can be carried out and the results can be extrapolated to the infinite chain limit ($N_b \to \infty$).

The structure of the rest of this chapter is as follows: In Section 6.2, the extrapolation procedure used here is described and the means by which we decide universality is discussed. Section 6.3 discusses the results of the double crossover studies, and establishes the scaling laws in the crossover region. Section 6.4 addresses the issue of the existence of a single crossover scaling function from which others can be inferred. Finally, the findings of this chapter are summarized in Section 6.5. The work presented in this chapter has been published in Jain et al. (2012).

6.2 Extrapolation Procedure and the Single Chain Diffusivity

In order to observe the behavior of semidilute polymer solutions, it is necessary to simulate long enough polymer chains. In this work, results in the infinite chain limit are obtained by an extrapolation procedure. The example of the ratio $D/D_{\rm Z}$ (which is the ratio of the single chain diffusion constant at a finite concentration to its value in the dilute limit) is used here to illustrate the procedure. The asymptotic result for $D/D_{\rm Z}$ at any value of z and $c/c^\star$ is obtained by accumulating finite chain data for $D/D_{\rm Z}$ from BD simulations, and subsequently extrapolating to the limit $N_b \to \infty$. For example, the red circles in Fig. 6.1 represent $D/D_{\rm Z}$ data for $z = 0.7$, $c/c^\star = 3$ and hydrodynamic interaction parameter $h^\star = 0.28$ for various values of N_b. Mean values and error bars of D for all values of N_b are estimated using the procedure discussed in Appendix A. The reason for choosing $1/\sqrt{N_b}$ as the x axis is justified shortly. The finite chain data are extrapolated to $1/\sqrt{N_b} \to 0$ limit with the help of a "Least-Squares Fitting" numerical routine, which is discussed in Appendix D. This routine, which accounts for the error bars at each finite size data point, also reports the error bar on the extrapolated data.

Kroger et al. (2000); Prakash (2001a); Kumar and Prakash (2003); Sunthar and Prakash (2005, 2006) found, for a single chain, that the extrapolation of finite chain data leads not only to asymptotic predictions at constant z, but also to values independent of the choice of h^*. This independence from the choice of a microscopic simulation parameter was taken to indicate the existence of universality. In their extrapolation analysis, they considered the x axis as $1/\sqrt{N_b}$, which was motivated by the fact that the leading order corrections to the infinite chain length limit, of various material properties, is of order $1/\sqrt{N_b}$.

A similar behavior appears to hold true for semidilute polymer solutions as well, by considering the x axis as $1/\sqrt{N_b}$. The red asterisks in Fig. 6.1 represent data for the same values of $z = 0.7$ and $c/c^\star = 3$ but for a different value of $h^\star = 0.15$. As can be seen in Fig. 6.1, the error bars on the extrapolated values are large because of a conservative estimation of the error (see Appendix D). However, as the extrapolated data for $h^\star = 0.28$ and 0.15 are very close to each other, we consider the prediction of D/D_Z in the infinite chain limit to be independent of $h^\star$. Fig. 6.1 shows another data set for $z = 1.7$ and $c/c^\star = 2$ using blue symbols, and here also the results in the infinite chain limit are independent of $h^\star$. This clearly indicates that the extrapolated value is a universal value independent of the choice of the simulation parameter $h^\star$.

Apart from the hydrodynamic interaction parameter $h^\star$, there is another arbitrary parameter which arises from the use of the narrow Gaussian potential

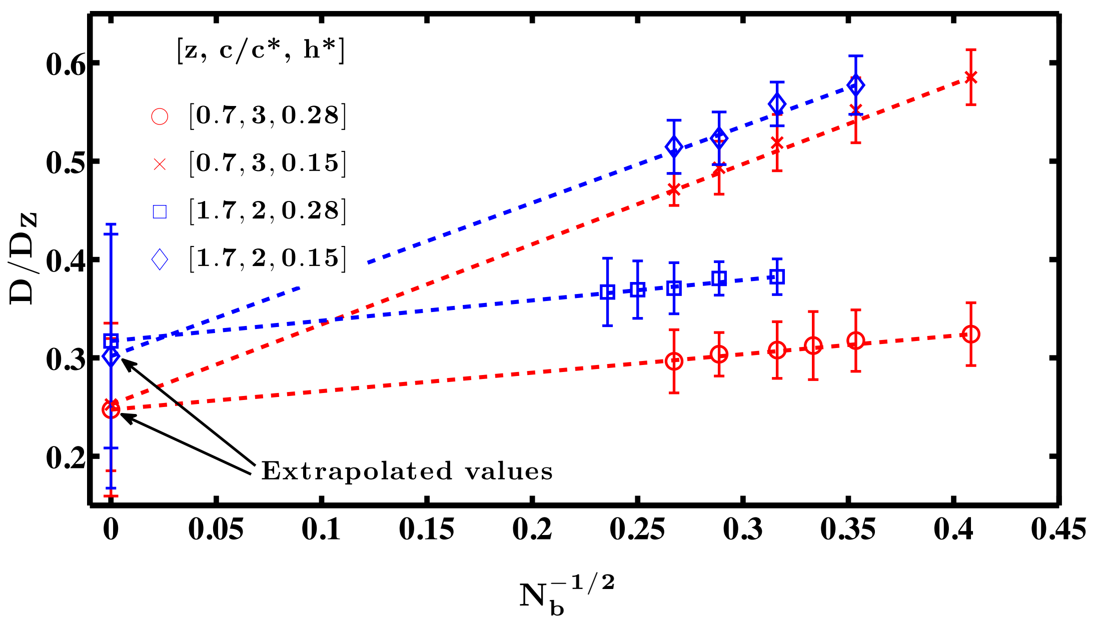


Figure 6.1: Universal predictions of the ratio of the single chain diffusion constant D (at a finite concentration $c/c^\star$), to its value in the dilute limit D_Z, at two values of the solvent quality $z = 0.7$ and $z = 1.7$. Data accumulated for finite values of chain length N_b (symbols) for two different values of $h^\star$ extrapolate to a unique value in the limit $N_b \to \infty$.

for excluded volume interactions. For a given value of solvent quality z, the parameters used in the narrow Gaussian potential are (i) $z^\star = z/\sqrt{N_b}$ and (ii) $d^\star$. Note that $z^\star$ is determined once a choice for z and N_b is made. The parameter $d^\star$ can be chosen arbitrarily. For reasons elaborated in Kumar and Prakash (2003), instead of $d^\star$, we choose to pick a value for the constant K, and then determine $d^\star$ from the relation $d^\star = K z^{\star 1/5}$. It has been shown previously by Kumar and Prakash (2003) that, for a given z, the results extrapolated in the limit $N_b \to \infty$ are independent of the particular choice of K. This finding turns out to be true for semidilute solutions as well. We carried out simulations for fixed values of $z = 1.7$ and $c/c^\star = 2.5$, and varied $d^\star$ according to $d^\star = K z^{\star 1/5}$, with three different values of K. It is clear from Fig. 6.2 that the extrapolated results are essentially independent of K. Therefore, it can be concluded that

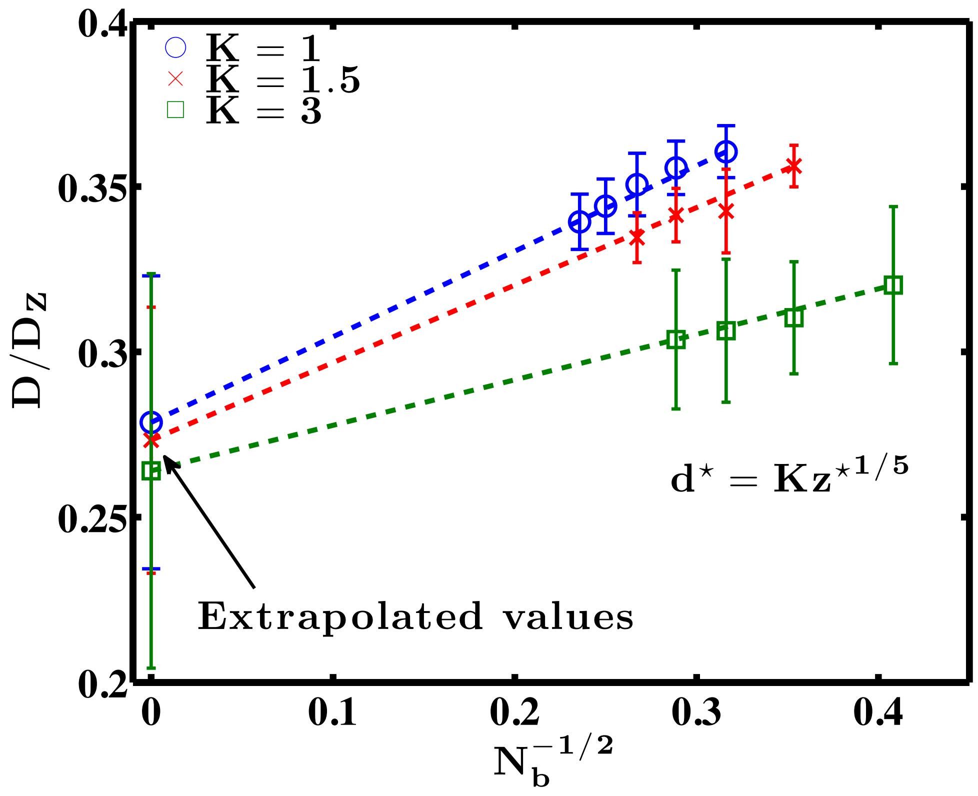


Figure 6.2: Extrapolated values of D/D_Z are independent of K for $z = 1.7$ and $c/c^\star = 2.5$.

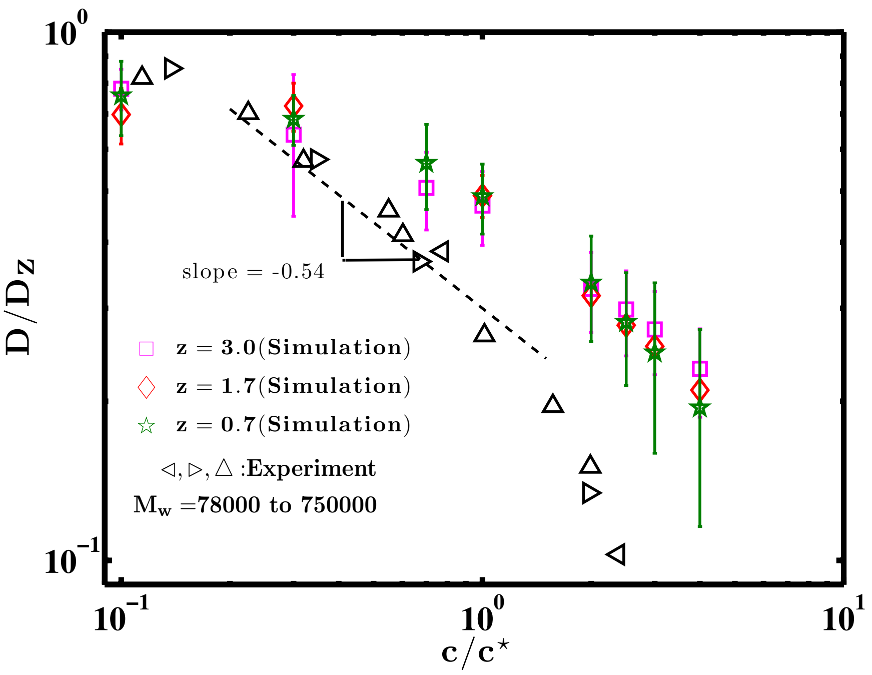
the extrapolation procedure not only provides the result for infinite chains but also leads to parameter free predictions that can be used to directly compare with experimental data. In this work, data accumulated for chains ranging from lengths $N_b = 6$ to $N_b = 20$ have been extrapolated to the limit $N_b \to \infty$.

6.2.1 Comparison with experimental data

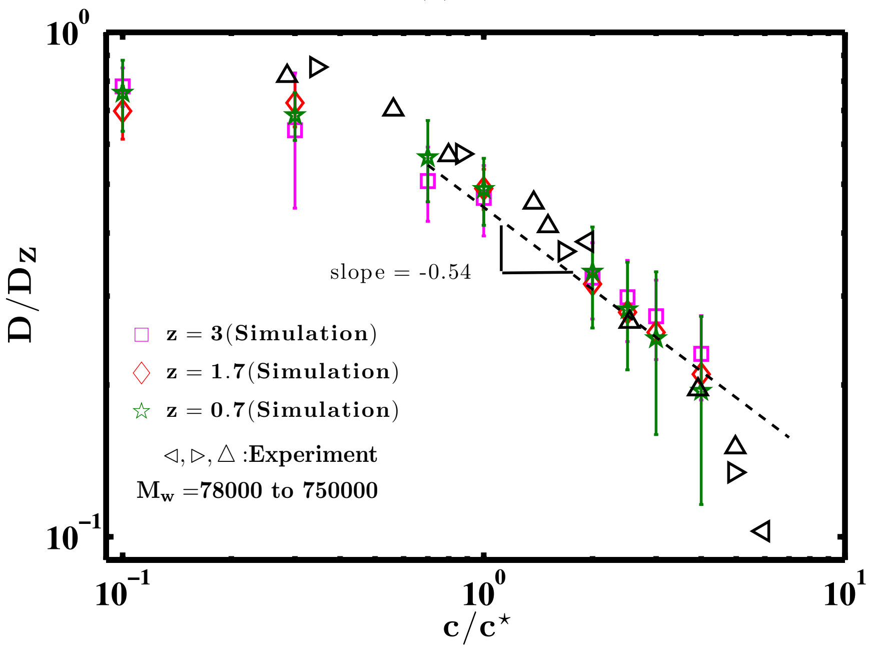
The main advantage of obtaining parameter free predictions in the long chain limit is that the extrapolated values of D/D_Z can be directly compared with experimental data. The self diffusion coefficient D was first measured by Marmonier and Leger (1985) for polystyrene in Benzene (good solvent) for different concentrations and molecular weights using forced Rayleigh light scattering. They also measured D at very low concentrations, which corresponds to D_{Z}. As a result, the ratio D/D_{Z} obtained from BD simulations can be compared with experimental D/D_{Z} data. Fig. 6.3 (a) shows the simulation results for various values of z, along with the experimental data for a good solvent system for a range of molecular weights (ranging from $78000 - 750000$). Colored symbols indicate the simulation results and black symbols represent experimental data. We conjecture that the disagreement between simulation results and experimental observations may be due to two reasons. Firstly, the overlap concentration $c^{\star}$ used by Marmonier and Leger (1985) is based on the following definition: $c^{\star} = \dfrac{N_m\, m\, v_s}{\mathcal{N}_A\, (R_g^0)^3}$, where N_m is the number of monomers in a chain, m is the molar weight of a monomer, v_s is the partial specific volume and $\mathcal{N}_A$ is the Avogadro number. Marmonier and Leger (1985) deduced the values of v_s from Pouyet et al. (1976), for polystyrene in Benzene for a range of molecular weights. This definition of $c^{\star}$ does not have the factor of $4\pi/3$ in the denominator which is included in the definition used here for $c^{\star}$.

Secondly, Marmonier and Leger (1985) do not report the values of R_g^0 used in their paper but rather provide reference to an unpublished PhD thesis. We are consequently unable to compare with simulation results for R_g^0. For these reasons, the value of $c/c^{\star}$ reported by Marmonier and Leger (1985) could be different from simulation values at the same absolute concentration c.

In addition, there is another noteworthy aspect to the experimental data obtained by Marmonier and Leger (1985). In general, the semidilute regime is considered to span the range $1 \leq c/c^{\star} < 10$ in neutral linear polymer solutions. According to scaling theory, the diffusivity in the semidilute regime scales as



(a)



(b)

Figure 6.3: Comparison of experimentally found D/D_Z data with BD simulation results: (a) Experimental data disagree with simulation results (b) Shifting the experimental x axis by a factor of 2.5 leads to agreement between the experimental data and simulation results

$(c/c^{\star})^{-0.54}$ in good solvents. The experimental data of Marmonier and Leger (1985) however suggests that the onset of this scaling occurs at a much lower value of $c/c^{\star}$.

We can bring experimental data and similar results much closer to each other, as shown in Fig. 6.3 (b) by applying a horizontal shift factor of 2.5 to the experimental data. Given the uncertainty in the experimental value of $c/c^{\star}$, this is perhaps not completely unreasonable. It would be worthwhile revisiting the problem of measuring the diffusivity in semidilute solutions and carrying out detailed comparisons in the entire crossover regime.

6.3 Universal Crossover Scaling Functions

In this section, the question of the form of the universal scaling function in the crossover regime is addressed with the help of simulation results for universal crossover scaling functions $\phi_R(c/c^{\star}, z)$ and $\phi_D(c/c^{\star}, z)$. These results are obtained by carrying out a careful Brownian dynamics simulations of short chains and then extrapolating the short chain results to infinite size chains as discussed in Section 6.2. Simulations to explore the solvent quality crossover have been carried out by keeping the value of z constant at four different values: $\{0, 0.7, 1.7, 3\}$, with $z = 0$ corresponding to θ-solutions. At each fixed value of z, a range of $c/c^{\star}$ from 0.1 to 4 have been used to sample both the dilute and semidilute regimes. The overlap concentration $c^{\star}$ is calculated from $c^{\star} = N_b/\left[(4\pi/3){R_g^0}^3\right]$, where R_g^0 is computed *apriori* by running *single-chain* BD simulations. The box size L is selected to ensure $L \geq 2R_e$, in order to prevent a chain from wrapping over itself. For this purpose, the value of R_e at any value of $c/c^{\star}$ is estimated from the blob scaling law $R_e = R_e^0\,(c/c^{\star})^{(2\nu-1)/(2-6\nu)}$ where R_e^0 is the end-to-end distance of a chain computed in the dilute limit. Typical simulations consist of an equilibration run for approximately one relaxation time followed by a production run whose length varies from 10 to 60 relaxation times; here, the relaxation time τ_1 was estimated via $\tau_1 = R_g^2/(6D)$. Moreover, data were obtained by averaging over $30 - 70$ independent runs. Some examples of the simulation parameters used in this study are shown in Table 6.1. The resulting crossover scaling

z	$c/c^\star$	N_b	L	N_c	Simulation length
	0.1	10	23.4	28	68.2 τ_1
0	1	20	15.1	27	12.9 τ_1
	4	16	13.4	103	10.2 τ_1
	0.1	8	24.3	32	44.7 τ_1
0.7	1	10	14.9	51	28.7 τ_1
	4	14	12.2	64	9.8 τ_1
	0.1	10	30.5	34	20.5 τ_1
1.7	1	18	20.4	38	12.8 τ_1
	4	14	13.5	67	10 τ_1
	0.1	10	30.1	27	28.3 τ_1
3	1	6	10	25	26.7 τ_1
	4	16	15.8	69	12.7 τ_1

Table 6.1: Some examples of simulation parameters used in obtaining universal crossover scaling functions.

functions $\phi_R(c/c^\star, z)$ and $\phi_D(c/c^\star, z)$ are shown in Figs. 6.4 and 6.5, where the mean polymer size R is R_g. The data are limited to the crossover region where in principle none of the power laws of Table 3.2 hold. Since the observation regime is small, it is nevertheless possible to fit a power law to the data (without deeper theoretical justification). This gives rise to a convenient description in terms of an *effective* empirical exponent ν_{eff} (see Table 6.2 and Fig. 6.4 for the values of ν_{eff}), which is obtained by adjusting the value of ν in the scaling laws of regime C of Table 3.2 to the observed slope for ϕ_R. Since the crossover scaling functions are related to each other (see below), each crossover yields the same value for ν_{eff}. For instance, in Fig. 6.5, lines are drawn, on top of ϕ_D data, based on the

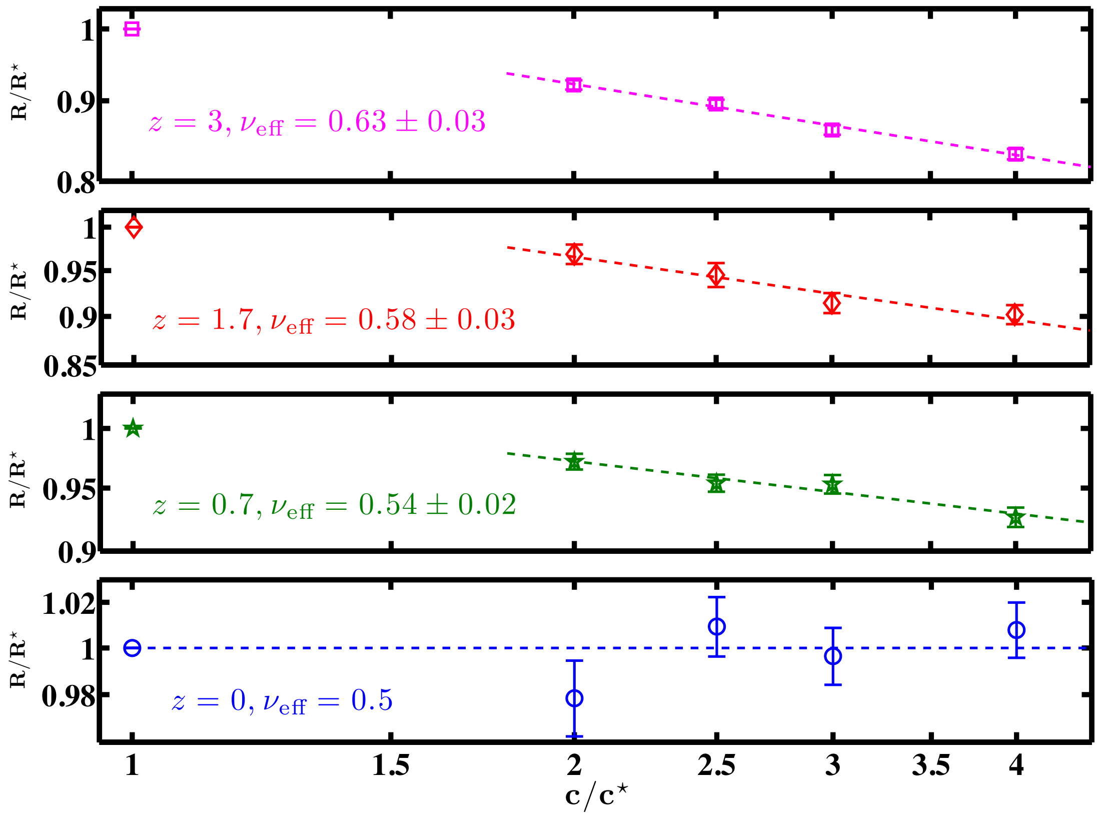


Figure 6.4: Concentration dependence of the crossover scaling function ϕ_R in the semidilute regime, for different values of solvent quality z, obtained by Brownian dynamics simulations. The effective exponents ν_{eff} have been determined by fitting power laws to the data $\phi_R \propto (c/c^\star)^{-p}$ and requiring $p = (2\nu_{\mathrm{eff}} - 1)/(6\nu_{\mathrm{eff}} - 2)$, according to Table 3.2 in regime C.

z	ν_{eff}
0	0.5
0.7	0.54 ± 0.02
1.7	0.58 ± 0.03
3	0.63 ±0.03

Table 6.2: Values of ν_{eff} for various z, found by adjusting the value of ν in the scaling laws of regime C of Table 3.2 to the observed slope for ϕ_R

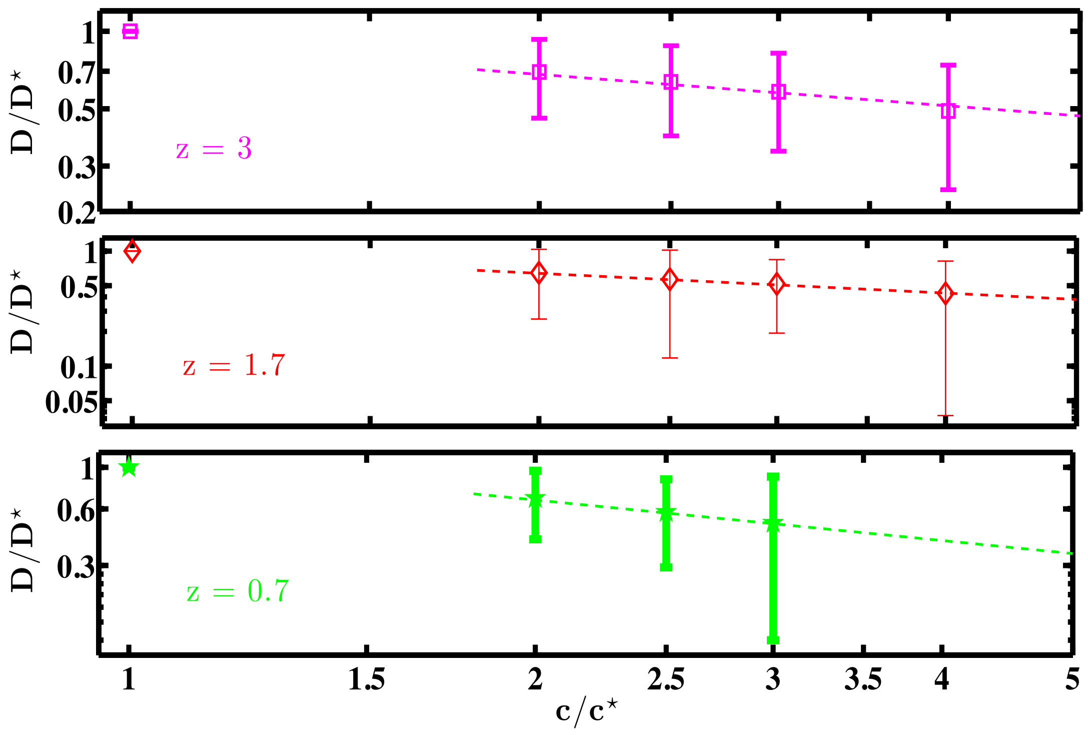


Figure 6.5: Concentration dependence of the crossover scaling functions ϕ_D in the semidilute regime, for different values of solvent quality z, obtained by Brownian dynamics simulations. Using the values of effective exponents for ϕ_R, lines are drawn according to $\phi_D \propto (c/c^\star)^{-q}$ with $q = (1 - \nu_{\text{eff}})/(3\nu_{\text{eff}} - 1)$. ϕ_D data for $z = 0$ are omitted because of large statistical errors.

slopes (or effective exponents) obtained for ϕ_R. Moreover, consistency with the asymptotic laws requires that ν_{eff} varies between 0.5 and 0.6.

A similar crossover scaling behavior is observed by Pan, Nguyen, Sunthar, Sridhar & Prakash (Pan et al.) for $\phi_\eta(c/c^\star, z)$ from measurements of zero-shear rate viscosity η_p for semidilute DNA solutions (see Fig. 6.6). In their experimental studies, they performed static and dynamic light scattering measurements on double-stranded DNA molecules, ranging in length from 3 to 300 kilobase pairs, to obtain the values of z and the theta temperature. Further, the parameter $c/c^\star$

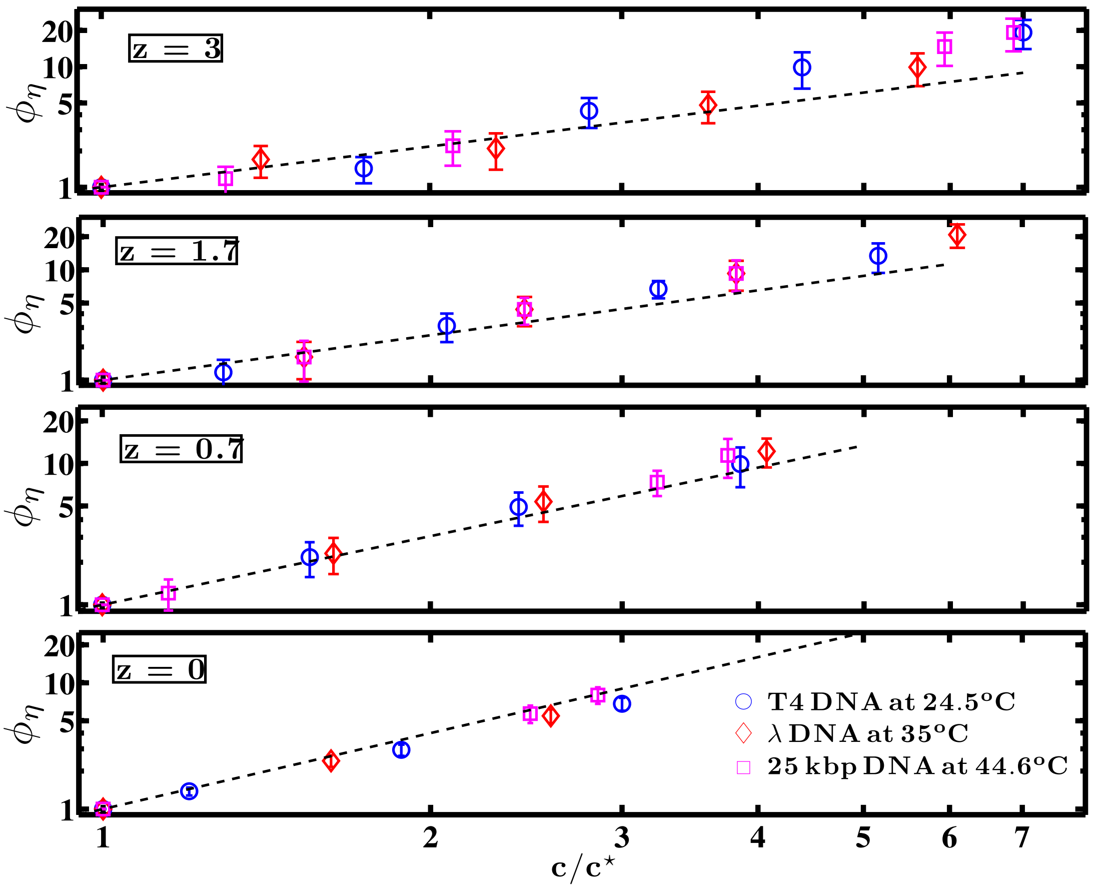


Figure 6.6: Concentration dependence of the crossover scaling functions ϕ_η in the semidilute regime, for different values of solvent quality z, obtained from rheological measurements on DNA solutions performed by Pan, Nguyen, Sunthar, Sridhar & Prakash (Pan et al.). Using the values of effective exponents for ϕ_R, lines are drawn according to $\phi_\eta \propto (c/c^\star)^r$ with $r = 1/(3\nu_{\text{eff}} - 1)$. This experimental data is reproduced with permission from Pan, Nguyen, Sunthar, Sridhar & Prakash (Pan et al.).

defined in their study is identical to the definition used in our simulations. This allows a decent comparison with simulation results for various values of $c/c^\star$ and z.

While we have not obtained predictions of η_p due to computational limitations, we can still compare our prediction of the scaling in the double crossover regime with experimental measurements. The lines shown in Fig. 6.6 are based on ν_{eff} obtained from simulations for ϕ_R at the corresponding values of z. Clearly, the predicted scaling at each value of z agrees remarkably well with experimental observations. As we shall see in the next section, this corroborates the expectation that there is only one scaling function.

6.4 The Existence of a Single Universal Crossover Scaling Function

As discussed in Chapter 3, the combination of crossover scaling functions together in certain specific ways leads to very simple functional dependences in the $(c/c^\star, z)$ phase diagram, as displayed in Table 3.2. In this section we validate these predictions of scaling theory with BD simulations.

Figure 6.7 examines the validity of the scaling prediction that the combination $(c/c^\star)^{1/4}\, \phi_R\, \phi_D^{1/4} = 1$ in the entire semidilute regime, while it is proportional to $(c/c^\star)^{1/4}$ in the dilute limit. It is clear from the Fig. 6.7 that indeed the functions ϕ_R and ϕ_D, which have been obtained by BD simulations in the long-chain limit, when combined as suggested by scaling theory lead to the predicted functional dependence in the two regimes, respectively.

Figure 6.8 examines the validity of the scaling prediction that the combination $(c/c^\star)^{1/3}\, \phi_R\, \phi_\eta^{-1/6} = 1$ in the semidilute regime, while it scales as $(c/c^\star)^{1/6}$ in the dilute regime. We have obtained ϕ_η from the experimental measurements reported in Pan, Nguyen, Sunthar, Sridhar & Prakash (Pan et al.), and ϕ_R from BD simulations as discussed previously. Clearly in this case as well, the scaling predictions are well substantiated. The departure of data from the $(c/c^\star)^{1/4}$ and $(c/c^\star)^{1/6}$ curves for values of $c/c^\star \approx 1$ in Figs. 6.7 and 6.8 can be attributed to

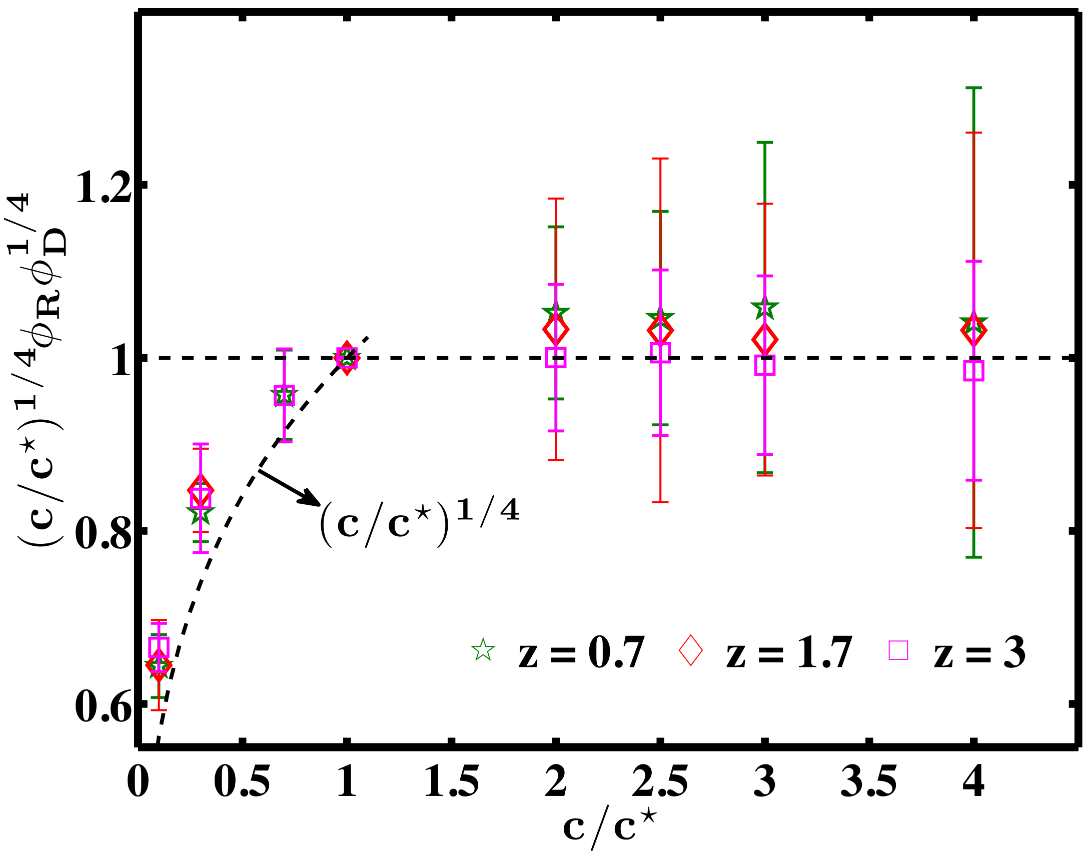


Figure 6.7: Demonstration that the combination $(c/c^\star)^{1/4}\,\phi_R\,\phi_D^{1/4} = 1$ in the semidilute regime and scales as $(c/c^\star)^{1/4}$ in the dilute regime. Here, $z = 0$ data are omitted because of large statistical errors.

the fact that these data are in the crossover region between dilute and semidilute, while the expression is strictly valid in the dilute limit.

Clearly, the knowledge of a single two-argument scaling function is sufficient to describe the entire double crossover regime in polymer solutions, since the other scaling functions can be obtained from the scaling combinations. Due to the universality of our results, these arguments should be broadly applicable to a large class of experimental systems.

Apart from polymer size and diffusivity, we have also studied the intra-chain

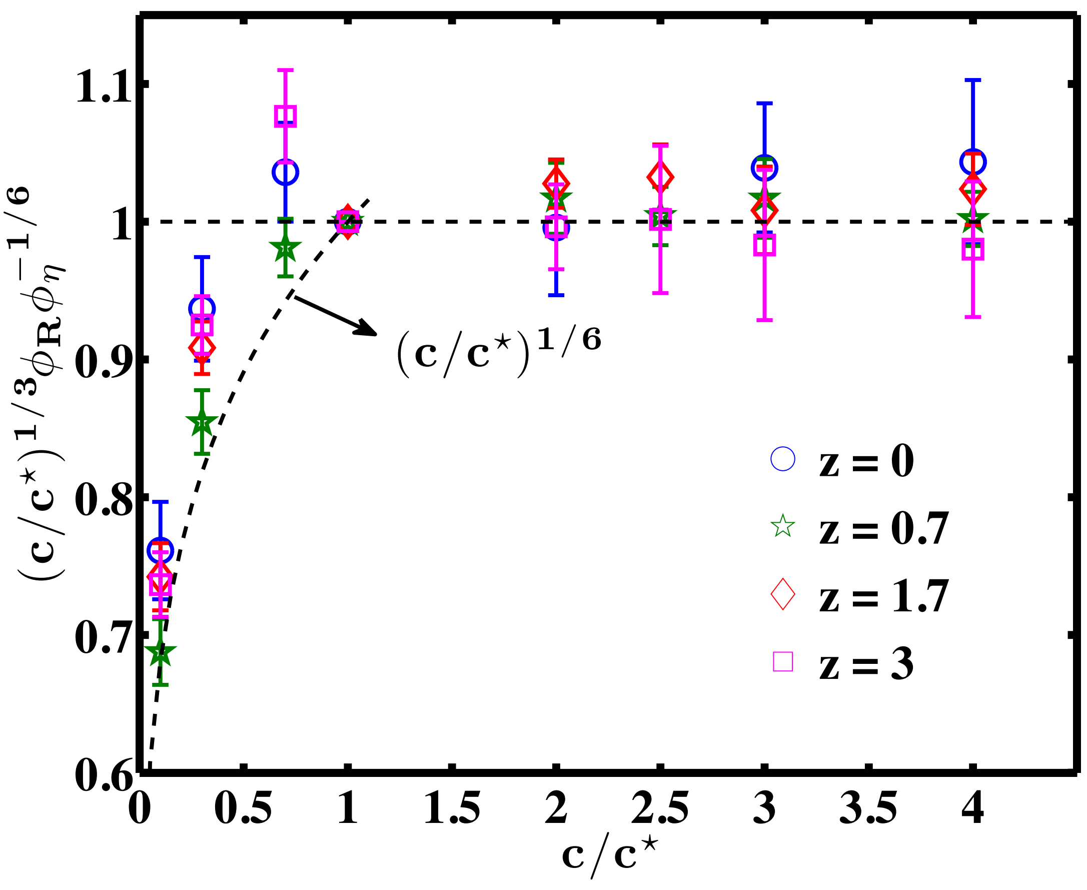


Figure 6.8: Demonstration that the combination $(c/c^\star)^{1/3}\,\phi_R\,\phi_\eta^{-1/6} = 1$ in the semidilute regime and scales as $(c/c^\star)^{1/6}$ in the dilute regime.

static structure factor of semidilute polymer solutions for a range of $c/c^\star$ and z, which is discussed in Appendix F.

6.5 Conclusions

The double crossover behavior for various physical properties was examined in semidilute polymer solutions with the help of Brownian dynamics simulations. Following are the key findings of this study:

6.5. Conclusions

1. Because of the high CPU cost, the current algorithm is restricted in simulating chains of up to 20 beads. However, an extrapolation technique can be used to circumvent the problem.

2. The extrapolation procedure was used to elucidate universal behavior. It was shown that, in the limit of infinite size chain, parameter free predictions are obtained that can be directly compared with experimental data.

3. Universal crossover scaling functions ϕ_R, ϕ_D and ϕ_η defined in terms of $c/c^\star$ and z were obtained. In particular, the blob scaling relationships, that are valid only for very good solvents and θ solvents, were shown to be also applicable in the solvent quality driven crossover regime, with an effective exponent in place of the Flory exponent.

4. The prediction of scaling theory that there exists only a single universal crossover scaling function from which others can be inferred, has been verified by simulations.

Chapter 7

Multi-Chain Brownian Dynamics Simulation Algorithm for Planar Shear, Elongational and Mixed Flows

7.1 Introduction

We have achieved a fairly complete description of static and dynamic properties of semidilute polymer solutions near equilibrium as shown in Chapters 3 and 6. The next challenge is to extend this understanding to the case where the solution is undergoing flow. Because of time constraints, we have only done the preliminary work of extending the multi-chain BD algorithm to an algorithm which is capable of simulating planar shear, elongational and mixed flows. This chapter discusses the issues pertaining to the implementation of these flows, and summarizes some preliminary results that we have obtained.

In simulations, periodic boundary conditions (PBC's) are often used to mimic real systems, enabling us to compute bulk properties by simulating only a small number of particles. This chapter discusses the implementation of suitable PBC's for planar shear flow, planar elongational flow and planar mixed flow in the multi-chain Brownian dynamics simulation algorithm for semidilute polymer solutions.

The term $[\boldsymbol{\kappa} \cdot \mathbf{r}_\nu(t)]$ in Eq. (4.1) (in Chapter 4) accounts for the presence of a homogeneous flow field, where $\boldsymbol{\kappa}$ is a 3×3 tensor and is equal to $(\boldsymbol{\nabla v})^T$, with $\boldsymbol{v}$ being the unperturbed solvent velocity. While the implementation of the term $[\boldsymbol{\kappa} \cdot \mathbf{r}_\nu(t)]$ is straightforward, the major challenge is the implementation of PBC's. Details of the implementation procedure are discussed in Section 7.2. Another difficulty that arises in the implementation of various flows is in treating the diffusion term in Eq. (4.1), which accounts for pairwise hydrodynamic interactions. As discussed in Section 5.2, the Ewald summation method is used to compute pairwise summation of hydrodynamic interactions. However, for a solution undergoing flow, some additional considerations must be taken into account while calculating the Ewald summation because the simulation cell may not be cubical. A few important points that need to be considered in calculating the Ewald sum (Wheeler et al., 1997) are discussed in Section 7.3. Validation studies for the algorithm in different flow geometries are presented in Section 7.4. Some preliminary results for a semidilute polymer solution of FENE dumbbells at finite concentration undergoing planar mixed flow are presented in Section 7.5. Finally, the conclusions of this chapter are given in Section 7.6.

7.2 Implementation of Periodic Boundary Conditions

It is a prerequisite condition for flow simulations with PBCs that the shape of the simulation box changes with time in accordance with the flow; this way Eq. (4.1) is in conjunction with suitably compatible PBCs. As the simulation box deforms with respect to time, there comes a time when the box has deformed to such an extent that the minimum spacing between any two sides of the box becomes less than twice the inter-particle interaction range. At that point in time, particles start to interact with themselves and the simulation needs to be stopped. There might also be cases, such as in shear flows, where after some time, one of the sides of the box becomes very large resulting in numerical problems. In other words, the deformation of the simulation box in such a manner restricts the simulation from proceeding for long times. In fact, this issue becomes even more serious for polymer molecules, since in this case, relaxation times in general are quite long, and it is very important to simulate them for sufficiently long time in order to capture their dynamics accurately.

PBCs for planar shear flows were developed by Lees and Edwards (1972) such that the simulation could be carried out for sufficiently long times. Unfortunately, it is not possible to generalize the Lees-Edwards PBCs to other flow geometries. In order for PBCs to be applicable to any flow, it is required that the simulation box should deform in accordance to the streamlines of that particular flow. However, as the simulation box cannot keep deforming for a very long time, it is required to perform a mapping of the box such that the initial box configuration is periodically recovered. Remapping of the box configuration requires two conditions to be met: (i) *Compatibility*, which means that the minimum lattice spacing should never be less than twice the range of inter-particle interactions, (ii) *Reproducibility*, which means that the lattice points of a lattice should overlap onto the lattice points of the same lattice at a different time. Remapping of lattice in nonequilibrium molecular dynamics (NEMD) simulations of planar shear flow was first carried out by Bhupathiraju et al. (1996) who modified the original sliding-brick algorithm of Lees and Edwards (1972) to a deforming-box algorithm. Satisfying the two conditions of compatibility and reproducibility, Kraynik and Reinelt (1992) developed PBCs capable of being remapped, for planar elongational flows. Kraynik-Reinelt PBCs were first implemented by Todd and Daivis (1998); Baranyai and Cummings (1999) in their planar elongational NEMD simulation algorithm. In these PBCs, basically the lattice (or simulation box) is deformed for a certain period of time τ_p, and then mapped back to its original state. This process of deforming the lattice till τ_p and mapping back to its original state is repeated for as many times as needed. This way, an extended simulation can be performed.

These special PBCs are derived and discussed in this section. In Section 7.2.1, PBCs are discussed for planar shear flows. Section 7.2.2 then presents PBCs for planar elongational flows and finally PBCs for combined planar shear and planar elongational flows or *planar mixed flows* are described in Section 7.2.3.

7.2.1 Planar shear flow

The simplest type of flow to be studied is the planar shear flow (PSF) or planar Couette flow. Because of the simplicity of studying PSF, there have been many studies that have attempted to compare experimental data with simulation predictions.

In a planar shear flow, the fluid flows only in a single direction, and the velocity has a gradient in the direction perpendicular to the flow, which is indicated by the shear rate $\dot{\gamma}$. The velocity gradient tensor for PSF is written as,

$$(\boldsymbol{\nabla v})_{\text{PSF}} = \begin{pmatrix} 0 & 0 & 0 \\ \dot{\gamma} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{7.1}$$

Though we have mentioned in Section 7.1 that the simulation box should deform with the flow, it is worth justifying this argument here by a visual picture. Originally, the PBC for planar shear flow was developed by Lees and Edwards (1972), in which they did not allow the lattice to deform but did allow it to move in a specific manner. This is illustrated in Fig. 7.1, where red marks indicate particles, the box in grey is the original simulation box, and other boxes are periodic images of the original box. First, we examine the case in which the lattice is not allowed to move (as in equilibrium simulations). The top panel in Fig. 7.1 shows this case. It is clear that after being subjected to shear flow, there is no periodicity in the lattice and hence this case is inconsistent with PBC. However, as can be seen in the bottom panel, moving or sliding the lattice with the flow in a specific manner maintains the periodicity in the lattice. This method is also called the *sliding brick method.* In this method, the image box above the simulation box is displaced by $L_{2y}\dot{\gamma}\Delta t$, and the image box below the simulation box is displaced by $-L_{2y}\dot{\gamma}\Delta t$, where L_{2y} is the height of the simulation box and Δt is the time step size. The advantage of this method is that the shape of each box does not change and hence the lattice vectors are always orthonormal. However, the major drawback of this method is the presence of non-aligned boundaries. When these boxes or cells slide past each other, their cell neighbors change and it is a burden in any algorithm to keep a track of all the neighbors of a cell with time.

Hansen and Evans (1994) and Bhupathiraju et al. (1996) found it inefficient to use the sliding brick method in a parallel computing environment. Therefore, they implemented an efficient algorithm by modifying the sliding brick method to the *deforming brick method.* In this method, instead of moving the cells, they deformed the cell with time as shown in Fig. 7.2. The periodic boundary condition in this method works in the following two ways: (i) if particles move out of

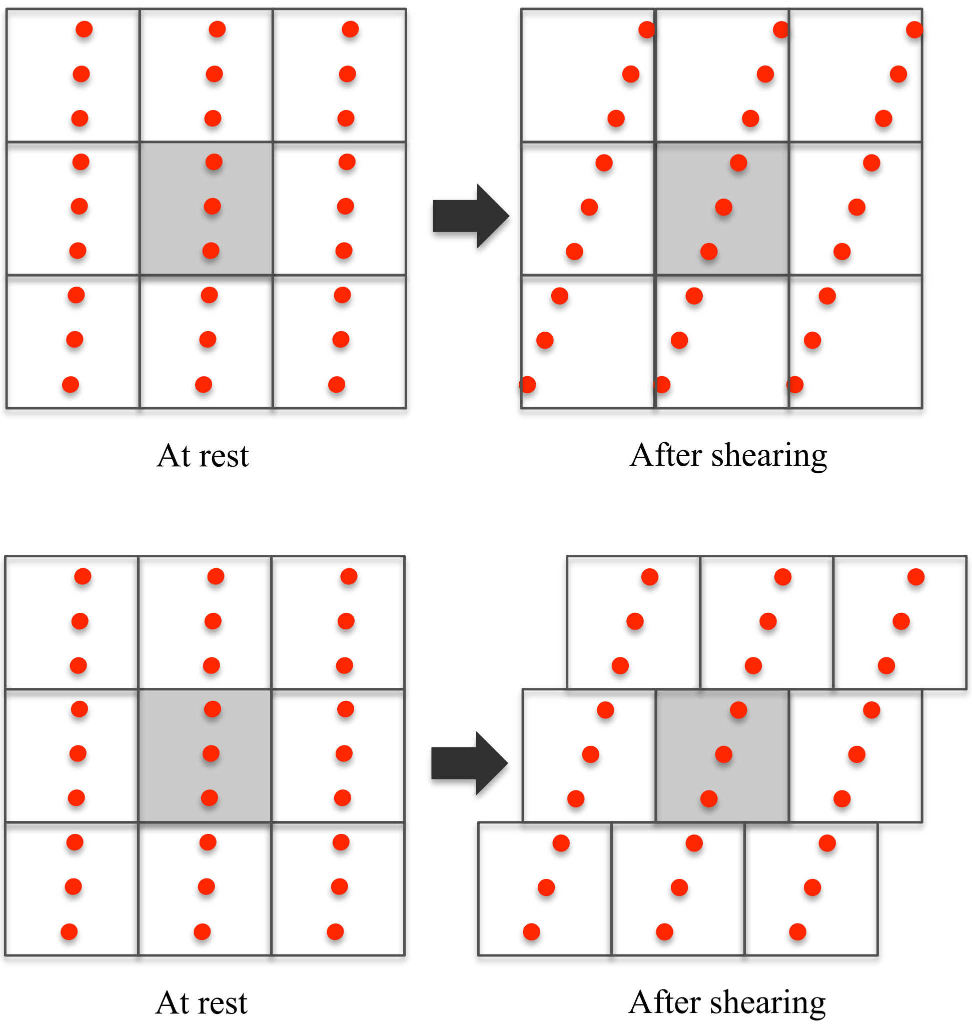


Figure 7.1: Demonstration of the necessity for sliding the lattice in planar shear flow.

the left/right side of the box, they are moved back in the box through right/left side of the box. (ii) if particles leave the box from the top/bottom side, they come back through bottom/top sides, but with a displacement of $\mp\Delta x$ in the x direction as shown in Fig. 7.2. The expression for this displacement can be shown to be $\Delta x = L_{2y}\dot{\gamma}\Delta t$, as will be discussed shortly.

We next address the question of how these boxes deform, or evolve with re-

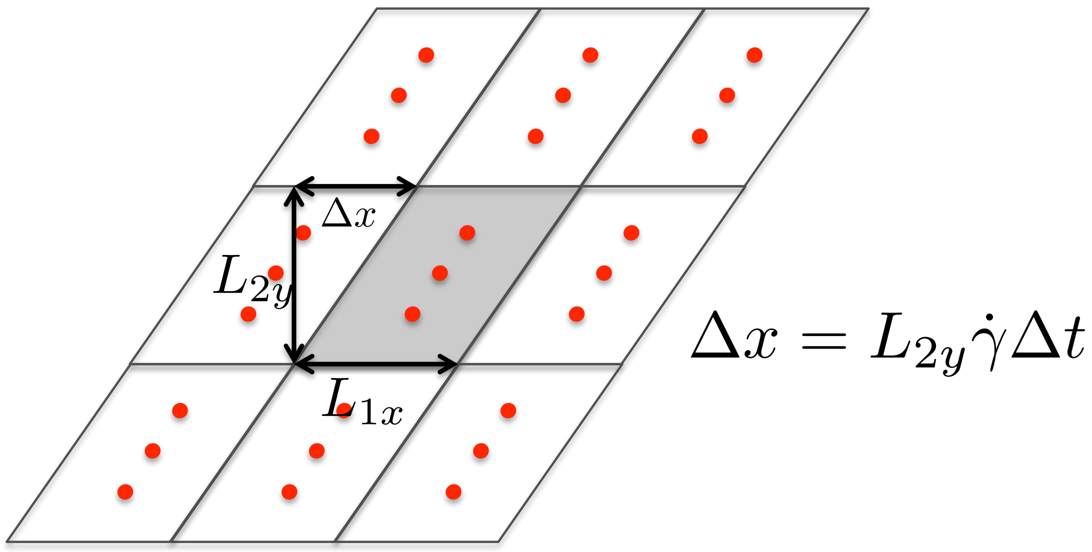


Figure 7.2: Demonstration of deforming brick method

spect to time. If $\mathbf{L_i}(t)$ is a lattice vector (or box vector) at time t, then the evolution equation for $\mathbf{L_i}(t)$ can be written for an arbitrary velocity gradient $\boldsymbol{\nabla v}$ as,

$$\frac{d\mathbf{L_i}}{dt} = \mathbf{L_i(t)} \cdot \boldsymbol{\nabla v} \quad (\text{for } i = 1, 2, 3) \tag{7.2}$$

For a planar shear flow, $\boldsymbol{\nabla v}$ in Eq. (7.2) can be replaced by $(\boldsymbol{\nabla v})_{\text{PSF}}$. Equation (7.2) can then be written for the x, y and z components as given below,

$$\frac{dL_{ix}}{dt} = L_{iy}\dot{\gamma} \tag{7.3}$$

$$\frac{dL_{iy}}{dt} = 0 \tag{7.4}$$

$$\frac{dL_{iz}}{dt} = 0 \tag{7.5}$$

It can be inferred from Eqs. (7.3) - (7.5) that the y and z components of any lattice vectors are constant, and only the x component evolves with respect to time. For a planar flow, L_{3x} is also constant and the only remaining variables are L_{1x} and L_{2x} that may change with time. Writing Eq. (7.3) for $i = 1$,

$$\frac{dL_{1x}}{dt} = L_{1y}\dot{\gamma} \tag{7.6}$$

Here, L_{1y} can be considered to be 0 for convenience; this way the lattice will always be aligned with the x-axis. As a result, L_{1x} is constant, implying that the top and bottom sides of the box are always constant and equal to L_{1x}, as depicted in Fig. 7.2. When Eq. (7.3) is written for $i = 2$,

$$\frac{dL_{2x}}{dt} = L_{2y}\dot{\gamma} \tag{7.7}$$

Recall that L_{2y} is constant. Upon integrating Eq. (7.7), and considering an initial condition that at time $t = 0$, $\mathbf{L_1} \perp \mathbf{L_2}$, i.e., $L_{2x}(t = 0) = 0$, we obtain,

$$L_{2x} = L_{2y}\dot{\gamma}t \tag{7.8}$$

Based on these lattice vectors, the evolution of the lattice system is shown in Fig. 7.3 for greater clarity. The simulation box is initially considered to be a square in 2-D (or cube in 3-D), hence the angle between $\mathbf{L_1}$ and $\mathbf{L_2}$ is $\pi/2$ at $t = 0$. This angle, based on simple geometry, can be expressed in terms of time as $\theta_s(t) = \tan^{-1}[L_{2y}/L_{2x(t)}] = \tan^{-1}[1/\dot{\gamma}t]$, where L_{2y} is constant and L_{2x} increases with time as given by Eq. (7.8). It is obvious that as $t \to \infty$, $\theta_s \to 0$ or $L_{2x} \to \infty$. This situation is computationally very inefficient as it requires dealing with very large numbers. Therefore, the simulation is not carried out for a very long time but rather until the time at which $\theta_s = \pi/4$, at which time, the box is transformed back to its original square lattice shape ($\theta_s = \pi/2$). In this way, L_{2x} never gets too large. The reason for choosing $\pi/4$ comes from the fact that

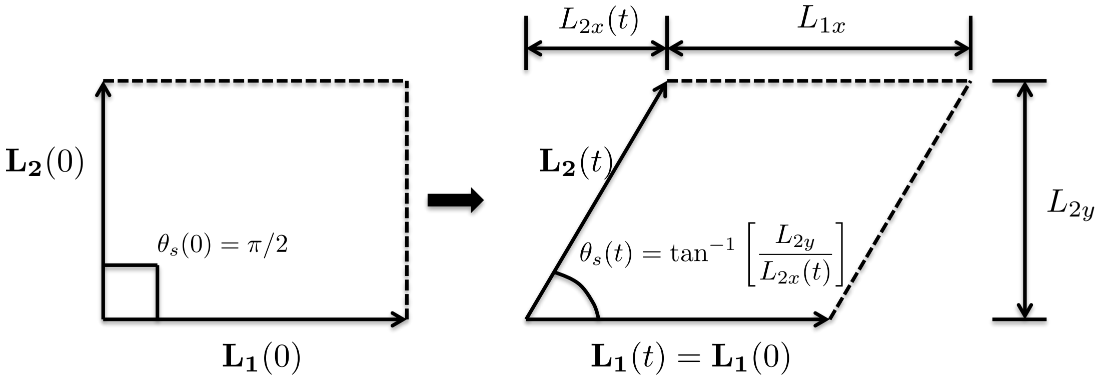


Figure 7.3: Lattice evolution in planar shear flow (Reproduced from Todd and Daivis (2007))

at this angle all the deformed lattice points can be overlapped with the original square lattice, as indicated by the black circles in Fig. 7.4. In this figure, the initial lattice is depicted in grey, while the lattice at $\theta_s = 45^0$ is indicated in red. This also means that the lattice is *reproducible* at $\theta_s = \pi/4$, and the time to reach there is simply $\tau_p = 1/\dot{\gamma}$. A slight modification can be made to this process to double the speed of the algorithm. In the modified form (Bhupathiraju et al., 1996), the initial box is considered to have an angle of $-\pi/4$ and is allowed to deform till $\theta_s = \pi/4$. In this way, the transformations are carried out less frequently. For this case, the evolution equation for L_{2x} has a different form due to a different initial condition (i.e., at $t = 0$, $L_{2x} = -L_{2y}$),

$$L_{2x} = L_{2y}\dot{\gamma}t - L_{2y} \tag{7.9}$$

The modified process is shown in Fig. 7.5. Here, the square lattice is indicated in grey, while the initial lattice, which is oriented at -45^0, is shown in blue. Upon imposing the shear flow, the blue lattice deforms and passes through a stage, at time τ_p , where it overlaps with the square lattice. Continuing the

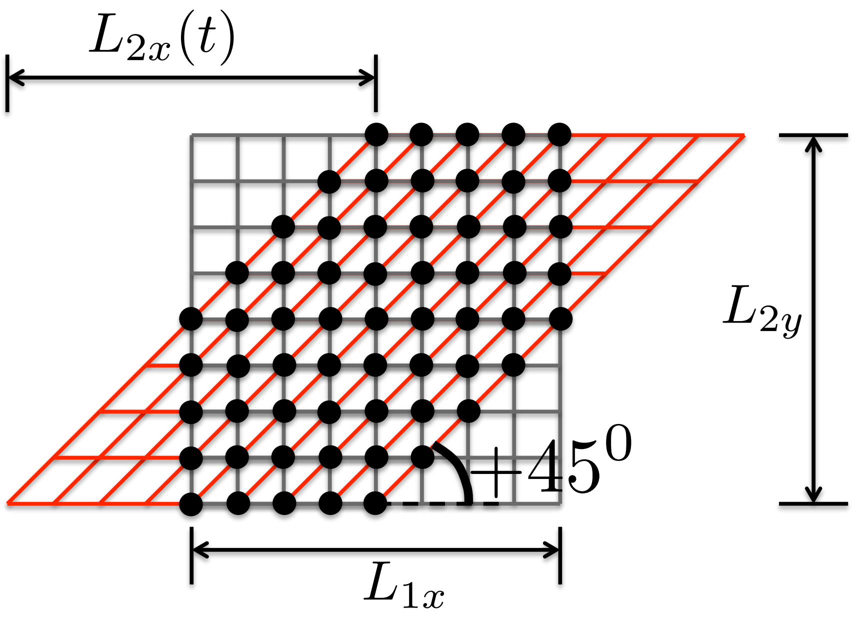


Figure 7.4: Lattice evolution in planar shear flow: 90^0 to 45^0 (Todd and Daivis, 2007)

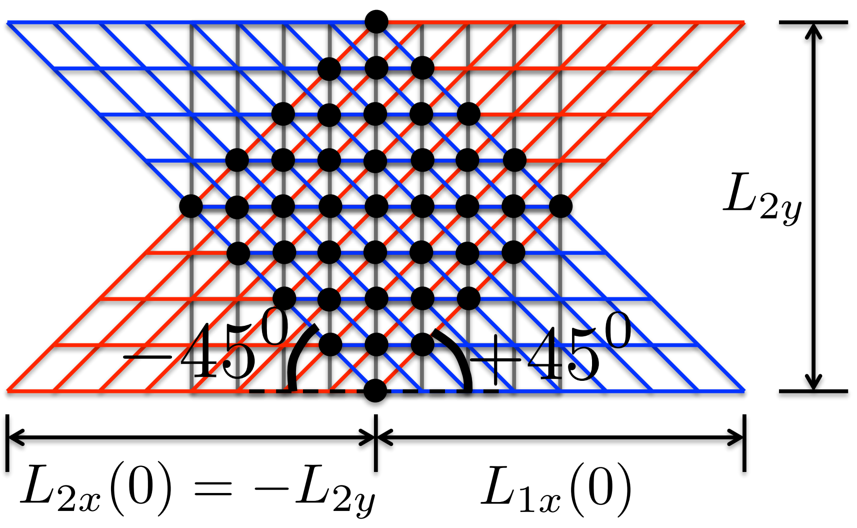


Figure 7.5: Lattice reproducibility in planar shear flow: Blue indicates initial lattice, grey indicates lattice after time τ_p and red indicates lattice after time $2\tau_p$

deformation, at time $2\tau_p$ the angle θ_s becomes $+45^0$ (red lattice) and then, as discussed above, the red lattice is transformed back to the blue lattice. The reason why this transformation is feasible is clear from Fig. 7.5; this is simply because the lattice at $+45^0$ is reproducible to the lattice at -45^0. In other words, the lattice points in these two stages ($\theta_s = -45^0$ and $+ 45^0$) overlap on top of each other, as indicated by the black symbols in Fig. 7.5. This confirms the reproducibility of lattices in planar shear flow, and also completes our discussion on the PBC's for planar shear flow.

7.2.2 Planar elongational flow

Elongational flow occurs in many industrial processes. It is generally difficult to study this flow using computer simulations and experimental techniques. In elongational flow, the fluid element is stretched exponentially with time, when subjected to a constant elongational rate. Therefore, a sample of fluid quickly becomes very thin and long as soon as elongational flow is imposed, and there is a very short span of time in which to observe the phenomena of stretching. Moreover, as discussed in Section 7.2.1, the simulation box needs to be deformed to follow the motion of the fluid, resulting in an exponential increase or decrease in the size of the simulation box. This situation is challenging because once a side of the stretched box becomes less than twice the inter-particle interaction range, the simulation has to cease. For carrying out indefinitely long simulations, special periodic boundary conditions need to be developed for elongational flows. A general velocity gradient for elongational, shear free and volume preserving flows can be written as (Bird et al., 1987)

$$(\boldsymbol{\nabla v}) = \begin{pmatrix} \dot{\epsilon} & 0 & 0 \\ 0 & -\dot{\epsilon}(1 + b_{\mathrm{f}})/2 & 0 \\ 0 & 0 & -\dot{\epsilon}(1 - b_{\mathrm{f}})/2 \end{pmatrix} \tag{7.10}$$

where $\dot{\epsilon}$ is the rate of elongation and b_{f} determines the type of elongational flow. For instance:

1. Uniaxial elongational flow occurs when $b_{\mathrm{f}} = 0$ and $\dot{\epsilon} > 0$
2. Biaxial elongational flow occurs when $b_{\mathrm{f}} = 0$ and $\dot{\epsilon} < 0$
3. Planar elongational flow occurs when $b_{\mathrm{f}} = 1$ and $\dot{\epsilon} \neq 0$

Kraynik and Reinelt (1992) showed that it is not possible to derive PBCs for uniaxial and biaxial elongational flows that can be used for indefinite simulation. However, they derived PBCs for planar elongational flows that satisfies this criteria. This section discusses the periodic boundary conditions for planar elongational flow proposed by Kraynik and Reinelt (1992).

In planar elongational flow (PEF), the fluid is stretched in one direction and contracts in another direction with the same elongational rate $\dot{\epsilon}$. The velocity gradient tensor for PEF is given by

$$(\boldsymbol{\nabla v})_{\mathrm{PEF}} = \begin{pmatrix} \dot{\epsilon} & 0 & 0 \\ 0 & -\dot{\epsilon} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{7.11}$$

Here, the fluid is being stretched in the x direction and contracts in the y direction. Figure 7.6 illustrates a schematic of the deformation of a cubic simulation box under planar elongational flow. Here, blue indicates the initial lattice and red indicates a lattice after some time of elongation. Using this square lattice, molecular dynamics simulations for elongational flow were first carried out by Heyes (1985). Unfortunately, they were not able to simulate for an indefinitely long time because at least one of the sides of the box becomes smaller than $2r_c$, where r_c is the cutoff radius of the inter-particle potential. If L_{2y}^{0} is the length of the simulation box in the contracting direction, then the maximum simulation

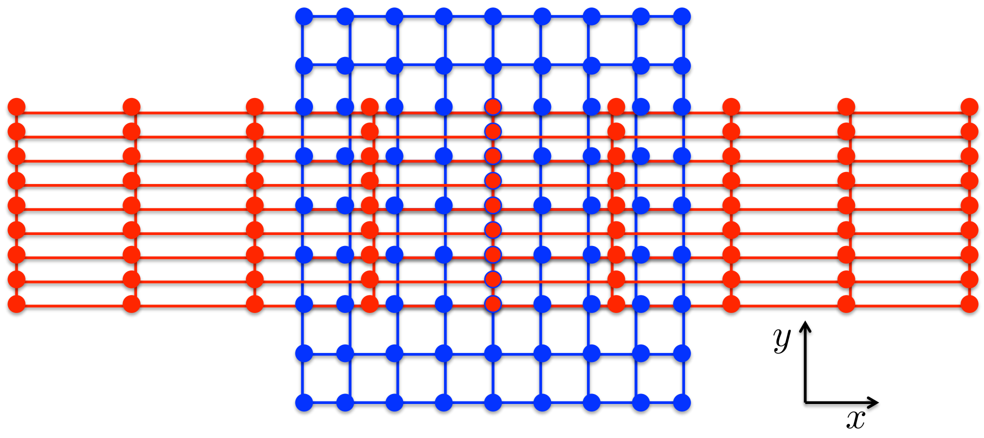


Figure 7.6: Schematic of a square lattice undergoing planar elongational flow

time can be shown to be $t_{\max} = \frac{1}{\dot{\epsilon}} \log\left(\frac{2r_c}{L^0_{2y}}\right)$.

As we have seen earlier, the lattice can be allowed to deform for an indefinite time if there is any state in the lattice evolution at which the lattice is reproducible and re-mappable to its original ($t = 0$) shape and size. It turns out that as long as reproducibility conditions are met, the compatibility condition is automatically satisfied. Kraynik and Reinelt (1992) derived periodic boundary conditions for planar elongational flow that satisfies the reproducibility condition. They discovered that if the initial lattice is inclined at a *magic angle*, then a reproducible lattice is achieved after a certain time called the *strain period*. As an example, a typical reproducible lattice is shown in Fig. 7.7. The derivation of the *magic angle*, *initial lattice vectors* and *strain period* have been carried out by Kraynik and Reinelt (1992). The starting point in the derivation is to obtain a reproducibility condition for the lattice, which is discussed here along with some key results. A detailed derivation of the Kraynik and Reinelt PBCs is presented in Appendix G.

Considering a lattice described by the unit lattice vectors $\mathbf{b_i}$ ($i = 1, 2, 3$), the

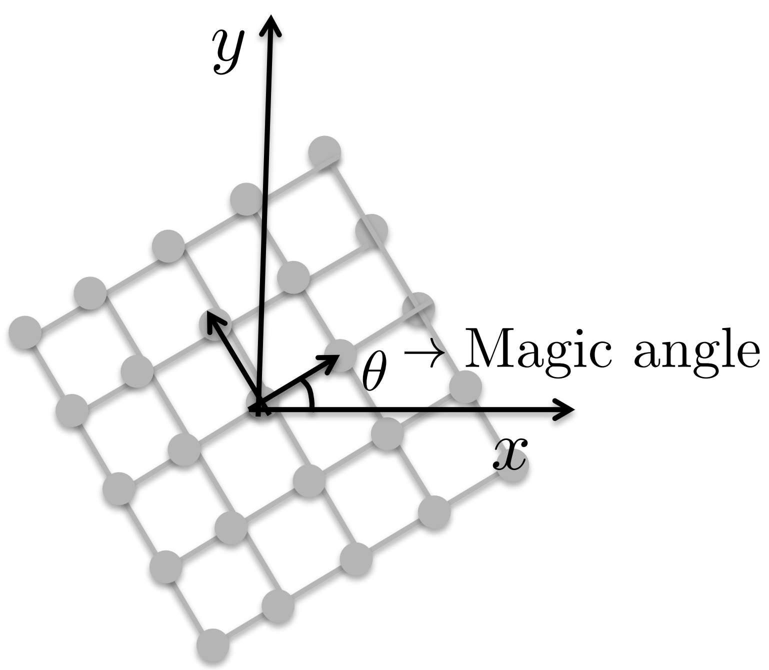


Figure 7.7: A reproducible lattice oriented at the magic angle

evolution of $\mathbf{b_i}$ can be written as,

$$\frac{d\mathbf{b_i}}{dt} = \mathbf{b_i} \cdot (\nabla \mathbf{v})_{\mathrm{PEF}} \tag{7.12}$$

If $\mathbf{b_1^0}$, $\mathbf{b_2^0}$ and $\mathbf{b_3^0}$ are the initial basis vectors, then Eq. (7.12) can be integrated to give,

$$\mathbf{b_i} = \mathbf{b_i^0} \cdot \boldsymbol{\Lambda} \tag{7.13}$$

where $\boldsymbol{\Lambda} = \exp\left[(\nabla \mathbf{v})_{\mathrm{PEF}} t\right]$. Given $(\nabla \mathbf{v})_{\mathrm{PEF}}$ is a diagonal tensor, $\boldsymbol{\Lambda}$ can be written as,

$$\boldsymbol{\Lambda} = \begin{pmatrix} e^{\dot{\epsilon}t} & 0 & 0 \\ 0 & e^{-\dot{\epsilon}t} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7.14}$$

$\boldsymbol{\Lambda}$ is also called the time evolution matrix. As Kraynik and Reinelt (1992) pointed out, for a given $\boldsymbol{\Lambda}$, a lattice is reproducible if and only if there exists integers N_{ij} such that,

$$\mathbf{b_i} = \mathbf{b_i^0} \cdot \boldsymbol{\Lambda} = N_{i1}\mathbf{b_1^0} + N_{i2}\mathbf{b_2^0} + N_{i3}\mathbf{b_3^0} \tag{7.15}$$

The justification of this reproducibility condition is explained in Appendix G. The reproducibility condition can be framed in terms of an eigenvalue problem, and the different results (such as the strain period τ_p and magic angle θ etc ...) can be interpreted in terms of eigenvalues as discussed in Appendix G. There exist multiple solutions for the strain period and the magic angle. An example of one such solution is $\theta \approx 31.72^0$ and $\tau_p = 0.9624/\dot{\epsilon}$. This example is shown

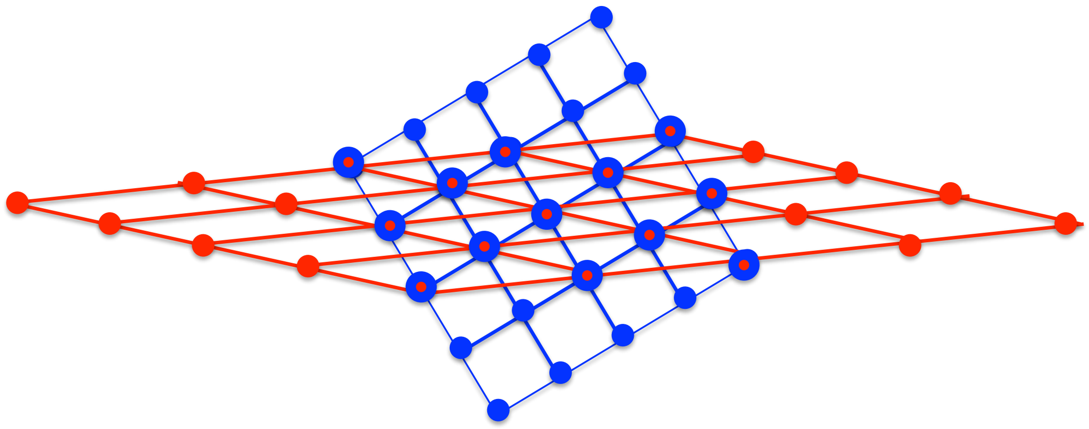


Figure 7.8: Lattice reproducibility for planar elongational flow

schematically in Fig. 7.8, where the lattice in blue indicates the initial lattice and the red lattice represents the lattice after τ_p. Clearly, the lattice points on the red and the blue lattices overlap on top of each other (indicated by the big blue circles), confirming the reproducibility. Once the magic angle θ is known, the basis vectors of the initial lattice ($\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}$) can be written as

$$\mathbf{b_1} = (\cos\theta, \sin\theta, 0) \tag{7.16}$$

$$\mathbf{b_2} = (-\sin\theta, \cos\theta, 0) \tag{7.17}$$

$$\mathbf{b_3} = (0, 0, 1) \tag{7.18}$$

The lattice based on these basis vectors is reproducible, and can be deformed till τ_p. At this time ($t = \tau_p$), the lattice is mapped back to its original shape and size. This guarantees that the lattice is not deformed any more once t reaches τ_p, which ensures that the compatibility condition is satisfied as well.

Having addressed issues regarding the initial lattice, magic angle and strain period, the only remaining question to ask is: how do the lattices evolve or how does a point in the lattice evolve? Following is a derivation for the evolution of lattice points in PEF. If $\mathbf{r}$ is a position vector of a lattice point, then the evolution equation for $\mathbf{r}$ can be written as

$$\frac{d\mathbf{r}}{dt} = \mathbf{r} \cdot (\boldsymbol{\nabla}\boldsymbol{v})_{\text{PEF}} \tag{7.19}$$

Using Eq. (7.11), the left hand side of the above equation can be written as

$$\mathbf{r} \cdot (\boldsymbol{\nabla}\boldsymbol{v})_{\text{PEF}} = \Big(\dot{\epsilon} r_x, \quad -\dot{\epsilon} r_y, \quad 0\Big) \tag{7.20}$$

where r_x and r_y are the x and y components of $\mathbf{r}$, respectively. Using Eqs. (7.19) and (7.20), the differential equations for r_x, r_y and r_z can be written as,

$$\begin{aligned} \frac{dr_x}{dt} &= \dot{\epsilon} r_x \\ \frac{dr_y}{dt} &= -\dot{\epsilon} r_y \\ \frac{dr_z}{dt} &= 0 \end{aligned} \tag{7.21}$$

This set of equations can easily be integrated, if the initial conditions for each component is known. Therefore, if r_x^0, r_y^0 and r_z^0 are the values of r_x, r_y and r_z, respectively, at time $t = 0$, then the expressions for r_x, r_y and r_z are:

$$\begin{aligned} r_x &= r_x^0 \exp\left(\dot{\epsilon} t\right) \\ r_x &= r_y^0 \exp\left(-\dot{\epsilon} t\right) \\ r_z &= r_z^0 = \text{constant} \end{aligned} \tag{7.22}$$

Hence, the lattice point $\mathbf{r}$ evolves as $\{r_x^0 \exp(\dot{\epsilon}t), r_y^0 \exp(-\dot{\epsilon}t), r_z^0\}$. In the simulation algorithm, the x and y component of any lattice vector are evolved according to Eq. (7.22). The lattice vectors $\mathbf{L_i}$ are related to $\mathbf{b_i}$ through $\mathbf{L_i} = |\mathbf{L_i}|\mathbf{b_i}$ since $\mathbf{b_i}$ are the unit vectors in the direction of $\mathbf{L_i}$. It follows that if $\mathbf{L_1}$ is a lattice vector in the direction of extension, then L_{1x} will evolve as $L_{1x}^0 \exp(\dot{\epsilon}t)$ and L_{1y} will evolve as $L_{1y}^0 \exp(-\dot{\epsilon}t)$, where L_{1x}^0 and L_{1y}^0 are the values of L_{1x} and L_{1y} at time $t = 0$. This completes our discussion of lattice evolution and the periodic boundary conditions for planar elongational flow.

7.2.3 Planar mixed flow

There has been a significant amount of work on computer simulations of planar shear and planar elongational flows because of their relative simplicity. However, in many practical applications, a linear combination of planar shear and planar elongational flows occur, which are denoted as *planar mixed flows.* An extended molecular dynamics simulation of planar mixed flow was first carried out by Hunt et al. (2010). With regard to polymer solutions undergoing mixed flow, simulation studies have been carried out only in the dilute concentration limit by Woo and Shaqfeh (2003); Dua and Cherayil (2003) and Hoffman and Shaqfeh (2007). Since only a single chain is required for simulating dilute polymer solutions, there is no need for reproducible periodic boundary conditions. However, in order to simulate planar mixed flow of semidilute polymer solutions, this is essential. To the best of our knowledge, there has been no attempt to simulate semidilute polymer solutions under planar mixed flow conditions. In this thesis, we adopt the reproducible periodic boundary conditions for planar mixed flow developed by Hunt et al. (2010), and use it in a multi-chain Brownian dynamics simulation algorithm for semidilute polymer solutions.

For planar mixed flow (PMF), the canonical form of the velocity gradient tensor $\boldsymbol{\nabla v}$ (Hunt et al., 2010) is given by,

$$(\boldsymbol{\nabla v})_{\mathrm{PMF}} = \begin{pmatrix} \dot{\epsilon} & 0 & 0 \\ \dot{\gamma} & -\dot{\epsilon} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{7.23}$$

As discussed in Section 4.5.2, for the canonical form of the velocity gradient tensor, the planar shear flow contribution is due to the velocity gradient in the y-direction, and the contribution from planar elongational flow leads to extension in the x-direction and contraction in the y-direction. In the canonical representation, the eigenvalues of $(\boldsymbol{\nabla v})_{\mathrm{PMF}}$ are $\{\dot{\epsilon}, -\dot{\epsilon}, 0\}$, and a possible choice of the corresponding eigenvectors is $\left(1, \dfrac{\dot{\gamma}}{2\dot{\epsilon}}, 0\right)$, $(0, 1, 0)$ and $(0, 0, 1)$. It is worth noting that the eigenvalues of the velocity gradient tensor of the canonical PMF are equivalent to those for PEF, where $(\boldsymbol{\nabla v})_{\text{PEF}}$ is already in a diagonal form. However, the eigenvectors corresponding to the eigenvalue $\dot{\epsilon}$ are different for $(\boldsymbol{\nabla v})_{\text{PMF}}$ and $(\boldsymbol{\nabla v})_{\text{PEF}}$. For PEF, the eigenvector corresponding to $\dot{\epsilon}$ is (1, 0, 0), which leads to the fact that the extension axis and contraction axis are orthogonal. In case of the canonical PMF, the eigenvector corresponding to $\dot{\epsilon}$ is (1, $\dot{\gamma}/2\dot{\epsilon}$, 0), resulting in a system where the extension axis and contraction axis are non-orthogonal. The angle β between the extension axis and the contraction axis depends on the ratio of $\dot{\gamma}$ to $\dot{\epsilon}$. The expression for β is displayed in Fig. 7.9 and given below

$$\beta = \cos^{-1}\left[\frac{\dot{\gamma}}{\sqrt{\dot{\gamma}^2 + 4\dot{\epsilon}^2}}\right] \tag{7.24}$$

Note that, for the sake of simplicity, we will use the phrase PMF instead of

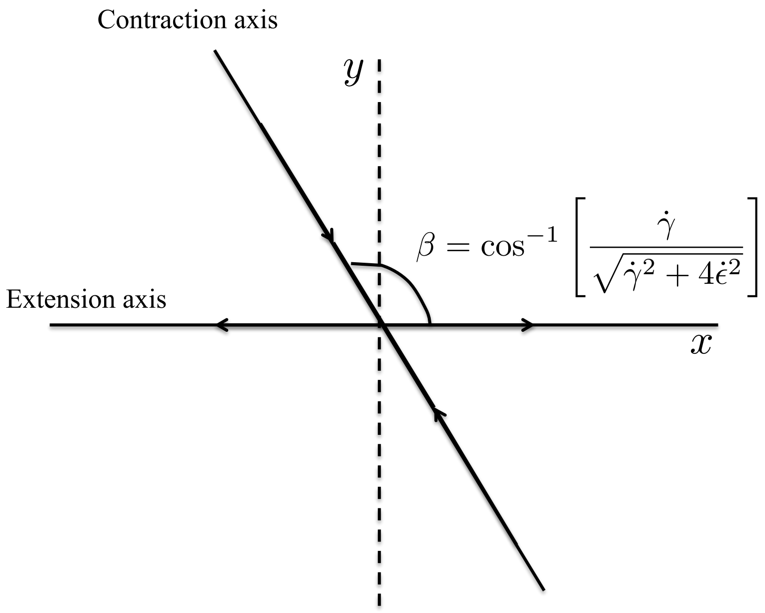

Contraction axis
y
$\beta = \cos^{-1}\left[\frac{\dot{\gamma}}{\sqrt{\dot{\gamma}^2 + 4\dot{\epsilon}^2}}\right]$
Extension axis
x

Figure 7.9: Axis of extension and contraction in planar mixed flow

"canonical" PMF.

Since the eigenvalues of the velocity gradient tensor for PMF and PEF are the same, the magic angle and strain period for PMF can be obtained in a similar manner as in the case of PEF. However, the initial lattice configuration for PMF is different from that of PEF because of the differences in the eigenvectors discussed above. An important point made by Kraynik and Reinelt (1992) in their derivation of PBCs for PEF is that $(\boldsymbol{\nabla v})_{\text{PEF}}$ can be replaced by any diagonalizable constant matrix with real eigenvalues and zero trace. Following this point, Hunt et al. (2010) realized that, in fact, $(\boldsymbol{\nabla v})_{\text{PMF}}$ is a diagonalizable matrix as written below

$$\begin{pmatrix} \dot{\epsilon} & 0 & 0 \\ \dot{\gamma} & -\dot{\epsilon} & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{\dot{\gamma}}{2\dot{\epsilon}} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\epsilon} & 0 & 0 \\ 0 & -\dot{\epsilon} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\dot{\gamma}}{2\dot{\epsilon}} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{T} \cdot \mathbf{D} \cdot \mathbf{T^{-1}} \quad (7.25)$$

where $\mathbf{T}$ is a transformation matrix that consists of the eigenvectors of $(\boldsymbol{\nabla v})_{\text{PMF}}$, and the diagonal matrix $\mathbf{D}$ has the same component form as $(\boldsymbol{\nabla v})_{\text{PEF}}$. The Kraynik-Reinelt periodic boundary condition for PEF is written in terms of the lattice evolution matrix $\boldsymbol{\Lambda} = \exp(\mathbf{D}t)$. Similarly for PMF, as the velocity gradient tensor $(\boldsymbol{\nabla v})_{\text{PMF}}$ is diagonalizable, we can write the lattice evolution matrix $\boldsymbol{\Lambda}'$ as

$$\boldsymbol{\Lambda}' = \exp\left((\boldsymbol{\nabla v})_{\text{PMF}} t\right) = \exp\left(\mathbf{T} \cdot \mathbf{D} \cdot \mathbf{T^{-1}} t\right) = \mathbf{T} \cdot \exp(\mathbf{D}t) \cdot \mathbf{T^{-1}} \quad (7.26)$$

As $(\boldsymbol{\nabla v})_{\text{PMF}} = \mathbf{T} \cdot \mathbf{D} \cdot \mathbf{T^{-1}}$ with $\mathbf{D}$ being a diagonal matrix, a new set of initial basis vectors,

$$\mathbf{b_i^{0\prime}} = \mathbf{b_i^0} \cdot \mathbf{T^{-1}} \quad (\text{for } i = 1, 2, 3) \quad (7.27)$$

exists in PMF, such that this new set is reproducible in the case of PMF (Hunt et al., 2010). The tensor $\mathbf{T}^{-1}$, thus, can be understood as a mapping necessary to make the PEF basis vectors $\mathbf{b_i^0}$ (in PEF) reproducible in the PMF. Similar to

Eqs. (7.13) and (7.15), an equation for the lattice reproducibility condition for PMF can be written as

$$\mathbf{b_i}' = \mathbf{b_i^{0\prime}} \cdot \boldsymbol{\Lambda}' \tag{7.28}$$

where $\mathbf{b_i}'$ denotes the lattice vector at time τ_p (strain period). Using this relation, and substituting $\boldsymbol{\Lambda}'$ form Eq. (7.26) in Eq. (7.28) leads to the following simplification

$$\begin{aligned}
\mathbf{b_i}'(t = \tau_p) &= \mathbf{b_i^{0\prime}} \cdot \boldsymbol{\Lambda}'(\tau_p) \\
&= \mathbf{b_i^0} \cdot \mathbf{T^{-1}} \cdot \mathbf{T} \cdot \exp\left(\mathbf{D}t\right) \cdot \mathbf{T^{-1}} \\
&= \mathbf{b_i^0} \cdot \exp\left(\mathbf{D}t\right) \cdot \mathbf{T^{-1}} \\
&= \left[N_{i1}\mathbf{b_1^0} + N_{i2}\mathbf{b_2^0} + N_{i3}\mathbf{b_3^0}\right] \cdot \mathbf{T^{-1}} \text{ (see Eq. (7.15))} \\
&= N_{i1}\mathbf{b_1^0} \cdot \mathbf{T^{-1}} + N_{i2}\mathbf{b_2^0} \cdot \mathbf{T^{-1}} + N_{i3}\mathbf{b_3^0} \cdot \mathbf{T^{-1}} \\
&= N_{i1}\mathbf{b_1^{0\prime}} + N_{i2}\mathbf{b_2^{0\prime}} + N_{i3}\mathbf{b_3^{0\prime}}
\end{aligned} \tag{7.29}$$

This equation for the reproducibility condition is identical to the one for PEF (Eq. (7.15)), except that $\mathbf{b_i^0}$ is replaced by $\mathbf{b_i^{0\prime}}$. The vectors $\mathbf{b_1^{0\prime}}$, $\mathbf{b_2^{0\prime}}$ and $\mathbf{b_3^{0\prime}}$ can be found easily since $\mathbf{b_1^0}$, $\mathbf{b_2^0}$ and $\mathbf{b_3^0}$ are known as discussed in Section 7.2.2. The mapping of Eq. (7.27) is applied to $\mathbf{b_i^0}$ to obtain a reproducible lattice under mixed flow as follows.

$$\begin{aligned}
\mathbf{b_1^{0\prime}} &= \mathbf{b_1^0} \cdot \mathbf{T^{-1}} \\
&= \begin{pmatrix} \cos\theta & \sin\theta & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\dot{\gamma}}{2\dot{\epsilon}} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \left[\left(\cos\theta - \frac{\dot{\gamma}}{2\dot{\epsilon}}\sin\theta\right), \sin\theta, 0\right]
\end{aligned} \tag{7.30}$$

$$\begin{aligned}
\mathbf{b_2^{0\prime}} &= \mathbf{b_2^0} \cdot \mathbf{T^{-1}} \\
&= \begin{pmatrix} -\sin\theta & \cos\theta & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\dot{\gamma}}{2\dot{\epsilon}} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \left[\left(-\sin\theta - \frac{\dot{\gamma}}{2\dot{\epsilon}} \cos\theta \right), \cos\theta, 0 \right]
\end{aligned} \tag{7.31}$$

$$\begin{aligned}
\mathbf{b_3^{0\prime}} &= \mathbf{b_3^0} \cdot \mathbf{T^{-1}} \\
&= \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{\dot{\gamma}}{2\dot{\epsilon}} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= [0, 0, 1]
\end{aligned} \tag{7.32}$$

where θ is the magic angle, which is similar to that for PEF. In contrast to PEF, where the basis lattice vectors are orthogonal, in the case of PMF, they are non-orthogonal and not equal in length. If the elongational rate is high or the shear rate is small, these lattice vectors becomes almost orthogonal and equal in length. These basis lattice vectors are used as an initial lattice configuration and the simulation is then run till the time approaches τ_p. The lattice is then mapped back to its original state, and this way the simulation can be carried out for extended period.

The question of how the lattices evolve in PMF can be addressed in a similar manner as has been done for PEF in Section 7.2.2. The evolution equation for a lattice point is shown in Eq. (7.19). The same equation can be used here but with $(\boldsymbol{\nabla v})_{\mathrm{PEF}}$ replaced by $(\boldsymbol{\nabla v})_{\mathrm{PMF}}$ given by Eq. (7.23). The left hand side of Eq. (7.19) for PMF becomes

$$\mathbf{r} \cdot (\boldsymbol{\nabla v})_{\mathrm{PMF}} = \begin{pmatrix} \dot{\epsilon} r_x + \dot{\gamma} r_y, & -\dot{\epsilon} r_y, & 0 \end{pmatrix} \tag{7.33}$$

Equations (7.19) and (7.33) can be used to obtain the differential equations for r_x, r_y and r_z as,

$$\begin{aligned} \frac{dr_x}{dt} &= (\dot{\epsilon} r_x + \dot{\gamma} r_y) \\ \frac{dr_y}{dt} &= -\dot{\epsilon} r_y \\ \frac{dr_z}{dt} &= 0 \end{aligned} \tag{7.34}$$

If r_x^0, r_y^0 and r_z^0 are considered to be the initial values of r_x, r_y and r_z, respectively, then these equations can be integrated and expressions for r_x, r_y and r_z can be found. For instance, r_y can be found to be

$$r_y(t) = r_y^0 \exp\left(-\dot{\epsilon} t\right) \tag{7.35}$$

and r_z is simply a constant equal to r_z^0. The governing differential equation for r_x is a first order linear differential equation that can be integrated to give

$$r_x = \exp(\dot{\epsilon} t) \left[-\frac{\dot{\gamma} r_y^0}{2\dot{\epsilon}} \exp(-2\dot{\epsilon} t) + C \right] \tag{7.36}$$

Using the initial condition for r_x, C can be found to be $r_x^0 + \dfrac{\dot{\gamma}}{2\dot{\epsilon}} r_y^0$. As a result,

$$r_x = \frac{\dot{\gamma}}{\dot{\epsilon}} r_y^0 \sinh(\dot{\epsilon} t) + r_x^0 \exp\left(\dot{\epsilon} t\right) \tag{7.37}$$

The lattice point $\mathbf{r}$ consequently evolves as $\{\frac{\dot{\gamma}}{\dot{\epsilon}} r_y^0 \sinh(\dot{\epsilon} t) + r_x^0 \exp(\dot{\epsilon} t), r_y^0 \exp(-\dot{\epsilon} t), r_z^0\}$. In the simulation algorithm, the x and y components of any lattice vector are evolved simply according to Eq. (7.22). For example, if $\mathbf{L_1}$ is a lattice vector in the direction of extension, then L_{1x} will evolve as $\frac{\dot{\gamma}}{\dot{\epsilon}} L_{1y}^0 \sinh(\dot{\epsilon} t) + L_{1x}^0 \exp(\dot{\epsilon} t)$ and L_{1y} will evolve as $L_{1y}^0 \exp(-\dot{\epsilon} t)$, where L_{1x}^0 and L_{1y}^0 are the values of L_{1x} and

L_{1y} at time $t = 0$.

There is only one issue left to discuss regarding the implementation of PMF in the simulation algorithm, and that is related to the compatibility condition. As discussed earlier, the length of one of the sides of the simulation box decreases with time. Kraynik and Reinelt (1992) showed that the reproducibility condition automatically guarantees the compatibility condition. In other words, they showed that the distance $D(\tau_p)$ between any two lattice points at time τ_p is never less than the minimum lattice spacing D_{min}, such that the lattice points do not overlap. In simulations, the cutoff radius of any inter-particle interaction potential is always chosen to be less than $D_{\text{min}}/2$, and that assures the compatibility condition is satisfied. The derivation of D_{min} has been carried out by Hunt et al. (2010), and here the detailed steps of the derivation are presented in Appendix H. This completes our discussion of the periodic boundary conditions for planar mixed flow.

7.3 Modifications in Calculating the Ewald Summation in the Presence of Flow

The Ewald summation method for hydrodynamic interactions has been presented in Section 5.2 for a cubic simulation cell in the context of equilibrium simulations. However, for far from equilibrium simulations, a cubic simulation cell cannot be considered because the simulation cell continuously deforms to be consistent with the imposed flow. A generalized Ewald sum method for a parallelepiped simulation cell was first developed by Wheeler et al. (1997), who adapted it for the Lees-Edwards periodic boundary condition. The same procedure can be adopted for any flow geometry such as planar elongational or planar mixed flows. Here, we briefly outline the important changes that need to be made in the real and reciprocal space calculations of the Ewald sum.

In general, the simulation is carried out in a parallelepiped cell which is defined by three cell basis vectors $\mathbf{L_1}$, $\mathbf{L_2}$ and $\mathbf{L_3}$, as shown in Fig. 7.10. While

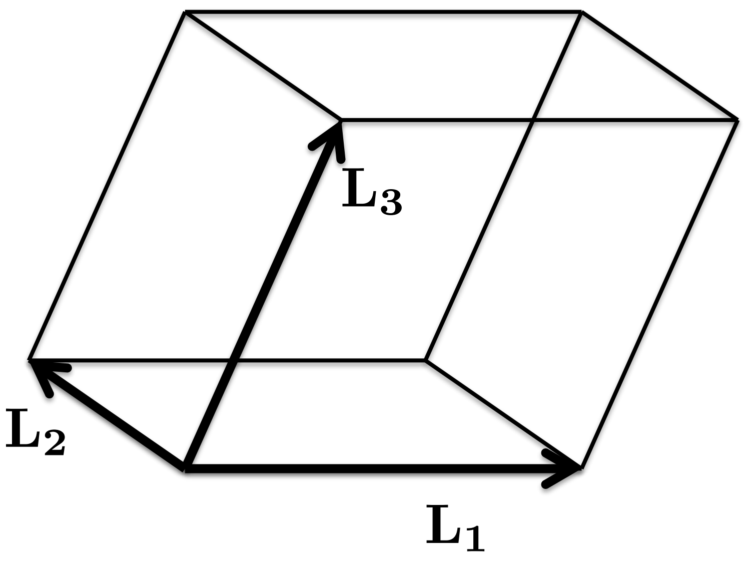


Figure 7.10: A parallelepiped simulation cell

performing real space calculations in the Ewald sum, the distance vector between particle μ and ν (where μ is the reference particle and ν is the target particle) is (Wheeler et al., 1997),

$$\mathbf{r}_{\nu\mu,\mathbf{n}} = \mathbf{r}_\nu - \mathbf{r}_\mu - \boldsymbol{\mathcal{L}} \cdot \mathbf{n} \tag{7.38}$$

where, $\boldsymbol{\mathcal{L}}$ is the cell basis matrix given by,

$$\boldsymbol{\mathcal{L}} = \begin{pmatrix} \mathbf{L_1} & \mathbf{L_2} & \mathbf{L_3} \end{pmatrix} = \begin{pmatrix} L_{1x} & L_{2x} & L_{3x} \\ L_{1y} & L_{2y} & L_{3y} \\ L_{1z} & L_{2z} & L_{3z} \end{pmatrix} \tag{7.39}$$

and $\mathbf{n}$ is the integer triplet given by $\mathbf{n} = [n_1,\, n_2,\, n_3]^T$. The volume of the simulation cell is $V = \mathbf{L_1} \cdot (\mathbf{L_2} \times \mathbf{L_3})$. Furthermore, in calculating the real space part of the sum, it is important to make sure that the magnitude of $\mathbf{r}_{\nu\mu,\mathbf{n}}$ for any pair μ and ν is always less than the real space cutoff radius r_c. In equilibrium simulations, once the cutoff radius is fixed, the number of neighboring cells or the cell neighbor-list is fixed for a given cell. However, in flow simulations, the implementation of the neighbor-list is much more complicated because the number of neighboring cells for a given cell keeps changing as the cells deform (Todd and Daivis, 1999).

In the reciprocal space part of the Ewald sum, the procedure remains the same as for equilibrium simulations, except the way in which the wave-vectors are calculated. For flow simulations, since the simulation cell is not a cube, the wave-vectors depend on the lattice vector of the simulation cell. The wave-vectors are generated through (Wheeler et al., 1997),

$$\mathbf{k} = 2\pi(\boldsymbol{\mathcal{L}}^T)^{-1} \cdot \mathbf{n} \tag{7.40}$$

In flow simulations, the difficulty in calculating the reciprocal space sum arises due to the fact that $\mathbf{M}^{(2)}(\mathbf{k})$ (see Section 5.2.1) needs to be updated at each time step, where as in the case of equilibrium simulations this term is calculated only once in the beginning of the simulation, and then used in all the time steps.

In equilibrium simulations there are a few terms in both the real and the reciprocal sums that are not updated at each time step but rather calculated in the beginning and used for the rest of the simulation. However, this is not true for flow simulations in which all the terms need to be updated at each time step. Therefore, this problem leads to an increase in the CPU cost as compared to the cost in equilibrium simulations.

7.4 Validation Studies

The validation of the extended BD algorithm has been carried out at very low concentrations, where the results of multi-chain BD can be compared with single-chain BD simulations for dilute polymer solutions. Due to time constraints, we have not carried out simulations with hydrodynamic interactions, but only with excluded volume interactions. Nevertheless, this allows us to test various aspects of the algorithm. This section presents some preliminary data on validation of the BD algorithm for flow simulations.

7.4.1 Planar shear flow

The main aspects in the development of a BD algorithm for flow simulations are the implementation of (i) a neighbor-list and (ii) periodic boundary conditions. We first validate our algorithm for a case in which these two implementations are unnecessary. This allows us to identify errors in the algorithm from sources other than due to the neighbor-list and PBCs implementation. The neighbor-list and PBCs do not play any role in the simulation when hydrodynamic and excluded volume interactions are ignored. Such a case is well described by the Rouse model for which analytical expressions for various properties are known. We switched off HI and EV interactions and computed the dimensionless mean square end-to-end distance $\langle R_e^2 \rangle$ of chains consisting of 10 beads in the ultra dilute limit, as a function of dimensionless shear rate $\dot{\gamma}$, and compared with Rouse model predictions given by (Bird et al., 1987)

$$\langle R_e^2 \rangle_{\text{Rouse}} = \langle R_e^2 \rangle_{\text{eq}} \left[1 + \frac{N_b\,(N_b+1)\,(N_b^2+1)\,\dot{\gamma}^2}{45} \right] \tag{7.41}$$

where, it may be recalled that $\langle R_e^2 \rangle_{\text{eq}} = 3(N_b - 1)$ is the mean square end-to-end distance at equilibrium (Bird et al., 1987). Red symbols in Fig. 7.11 indicate the

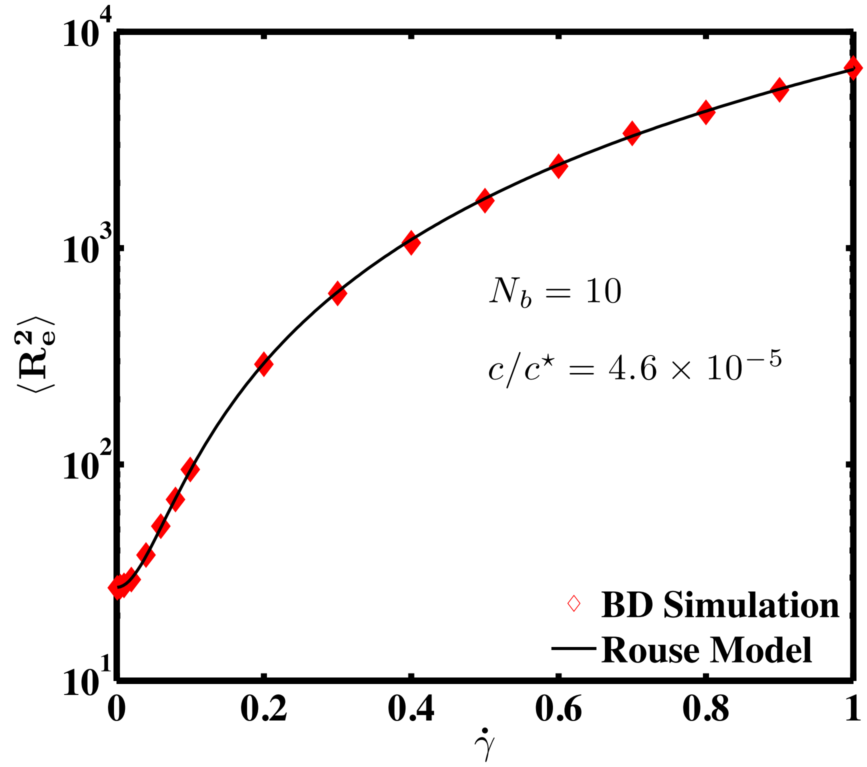
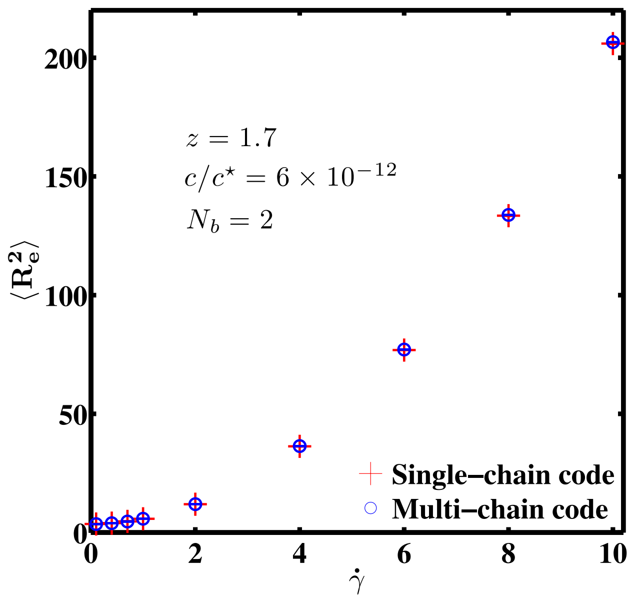


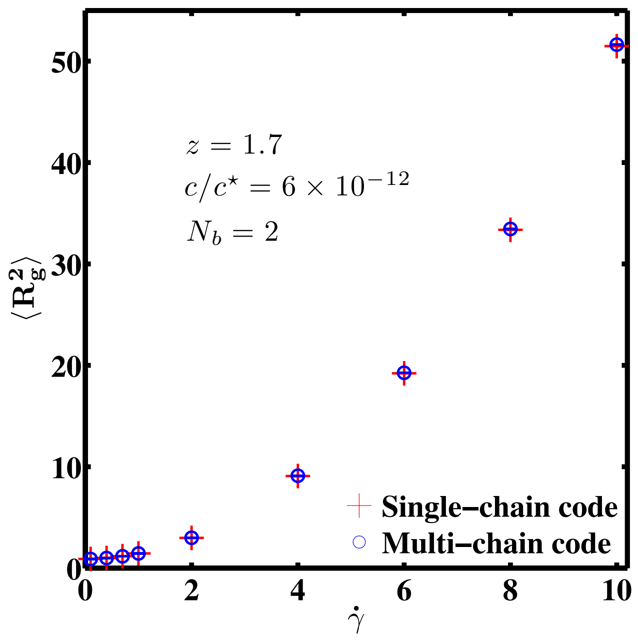
Figure 7.11: Mean square end-to-end distance obtained by BD simulations compared with Rouse model predictions, for bead-spring chains with $N_b = 10$ beads.

results for $\langle R_e^2 \rangle$ obtained by carrying out multi-chain BD simulations. The solid line in the figure shows the Rouse model predictions for a 10 bead chain. Multi-chain BD simulations were carried out by considering $N_c = 30$, $c/c^{\star} = 4.6 \times 10^{-5}$ and a time step size $\Delta t = 0.005$.

In order to test the neighbor-list and PBCs implementations, we have carried out multi-chain BD simulations of dumbbells ($N_b = 2$) in a ultra dilute system with EV interactions between the beads. In this study, simulations were carried out using $z = 1.7$, $N_c = 10$, $c/c^{\star} = 6 \times 10^{-12}$ and $\Delta t = 0.005$. A large number of independent runs (in the range of 10^3 - 10^5) were performed to obtain better statistics. Figure 7.12 shows the results for $\langle R_e^2 \rangle$ and $\langle R_g^2 \rangle$ for a range of shear



(a)



(b)

Figure 7.12: Comparison of the mean square end-to-end distance $\langle R_e^2 \rangle$ and the mean square gyration radius $\langle R_g^2 \rangle$, at various shear rate $\dot{\gamma}$, predicted by the multi-chain BD algorithm with the results of single-chain BD simulations in the dilute limit. EV interactions are taken into account but HI is switched off. The parameter values are indicated in the figure.

rates. These results are compared with the results obtained using single-chain BD simulations (for which the neighbor-list and PBCs are not required). Clearly, the agreement with single-chain simulations indicates our BD algorithm is valid.

We have also computed the viscosity η as a function of the shear rate $\dot{\gamma}$ and compared the results with single-chain BD simulations. Here η is calculated using Eq. (4.30), by considering $\chi = 0$ and $\dot{\Gamma} = \dot{\gamma}$. Blue circles in Fig. 7.13 show the multi-chain BD simulation results for the viscosity and the red plus symbols represent the data obtained by running single-chain BD simulations. Clearly, there is agreement between the multi-chain and single-chain BD simulation results. Also, shear thinning behavior can be observed in Fig. 7.13. At low shear rates, excluded volume effects increase the size of polymer chains and hence increase the viscosity. The decrease in viscosity with increasing $\dot{\gamma}$ is due to excluded volume interactions being switched off as the dumbbells undergo stretching in shear flow.

Though HI was not considered in the validation of physical properties, a basic check of the implementation of the Ewald sum for HI was carried out by comparing the numerical value of the Ewald sum obtained using the planar shear flow code at $\theta_s = 0^0$ with that obtained using the equilibrium code for the same bead configuration. We also calculated the Ewald sum at $\theta_s = 45^0$ and -45^0, and obtained results that matched with the equilibrium results. A more thorough check will be carried out by evaluating dynamic properties in the future, with the inclusion of HI.

7.4.2 Planar elongational flow

We now turn our attention to validation studies for the planar elongational flow (PEF) case. Our implementation of the Ewald sum was checked by using a similar approach to that for the case of PSF. We computed the Ewald sum for a

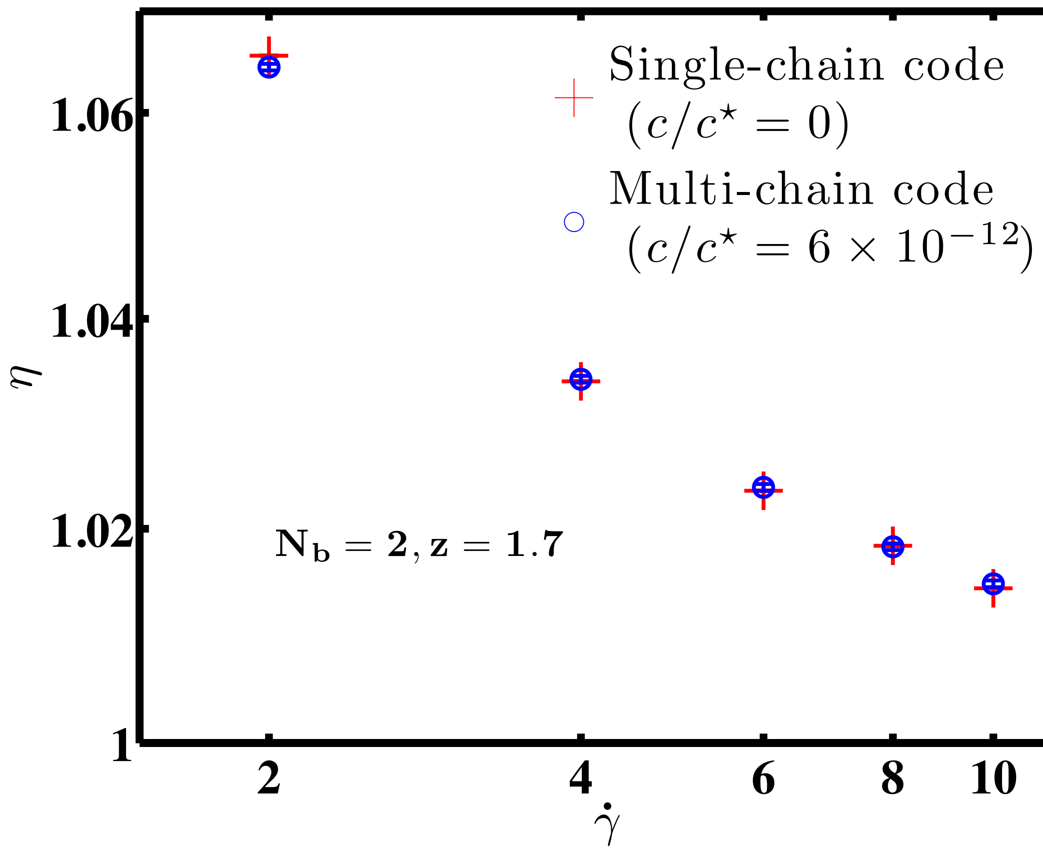


Figure 7.13: Comparison of the viscosity η, at various $\dot{\gamma}$, predicted by the multi-chain BD algorithm with the results of single-chain BD simulations in the dilute limit

given bead configuration (i) in a simulation cell oriented at the magic angle of $\theta = 31.72^0$, which corresponds to the initial simulation setup, and (ii) in a simulation cell at its maximum permissible deformed state, which occurs at the strain period. Ewald sums in both the cases were found to be identical because in these two cases the lattices are reproducible and periodic. This agreement establishes the correct implementation of the modified Ewald sum in the context of PEF as well. This study required calculating the Ewald sum at just one instant of time. A more thorough check of physical property predictions will be carried out in the future with the inclusion of HI.

We have, however, computed physical properties in PEF with EV interactions included. In PEF simulations, it turns out that as time progresses, the numerical values of the x coordinates of beads increases due to elongation in the x direction. As a result, after a long time, the numerical values are so high that the simulation suffers from numerical instability. In order to illustrate the problem, an example of PEF viscosity $\bar{\eta}_1$ for FENE dumbbells with finite extensibility parameter $b_p = 50$ and for elongation rate $\dot{\epsilon} = 0.3$ is shown as a function of time in Fig. 7.14. The correct value of $\bar{\eta}_1$ for the same parameters can be found by carrying out single-chain BD simulations, and it is found to be 4.351 ± 0.002. It is clear that the value of $\bar{\eta}_1$ reaches 4.35 very rapidly, however, after about 25 strain periods, a catastrophic change is observed. Eventually, $\bar{\eta}_1$ settles to a wrong value. A similar numerical instability has been observed in NEMD simulations of PEF, however, the source of this instability has been attributed to the lack of momentum conservation due to numerical round-off errors (Todd and Daivis, 2000).

We have solved the instability problem with the following approach. After each strain period, it is checked whether a chain is in close proximity of the simulation box. In order to check this proximity, the following steps are used. If $r_{\nu,x}$, $r_{\nu,y}$ and $r_{\nu,z}$ are the coordinates of bead ν, then, if either $|r_{\nu,x}| > 2L_1$, $|r_{\nu,y}| > 2L_2$, or $|r_{\nu,z}| > 2L_3$, the bead ν is considered to *not* be in the proximity of the simulation box. Here, L_1, L_2 and L_3 are the magnitudes of cell basis vectors $\mathbf{L_1}$, $\mathbf{L_2}$ and $\mathbf{L_3}$, respectively. If all the beads of a chain are not in the proximity of the simulation box, then we abandon this chain, and begin to follow the trajectory of the image of this chain that is in the proximity of the simulation box. This does not affect in anyway the calculation of any of the properties, since we are still tracking the trajectories of the same set of N unique particles

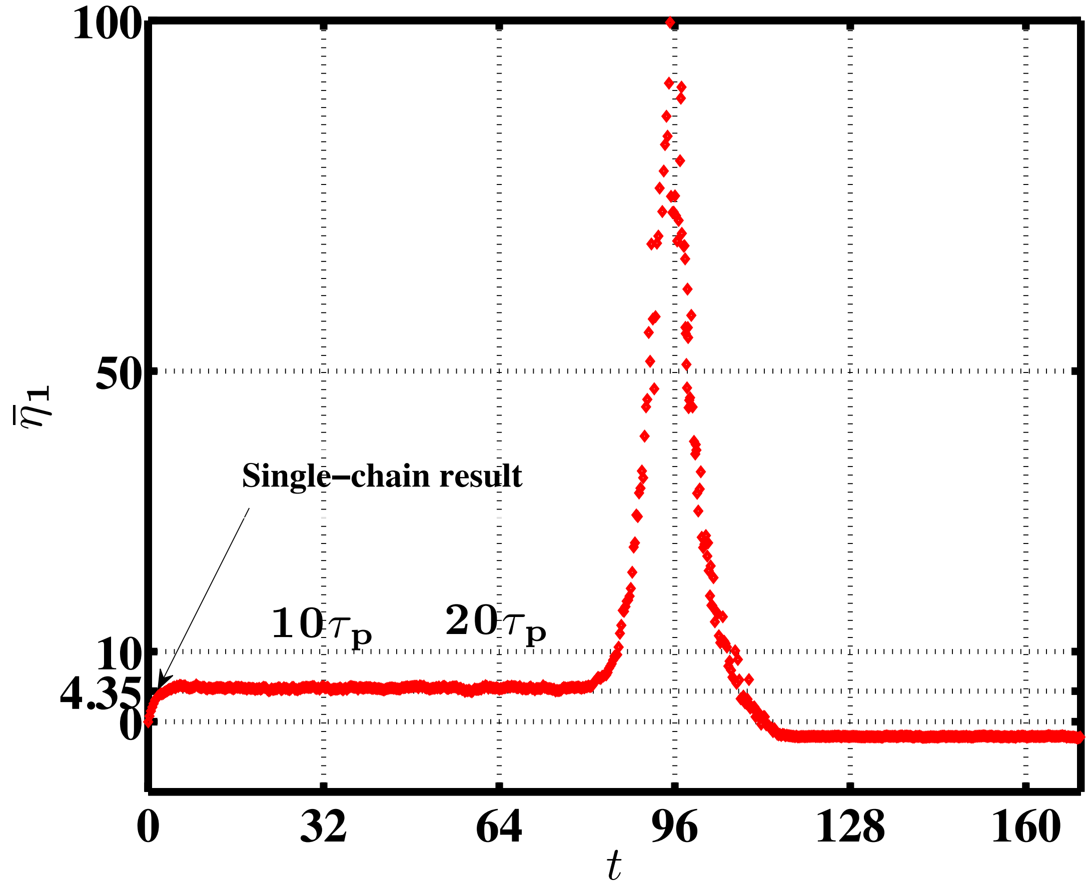
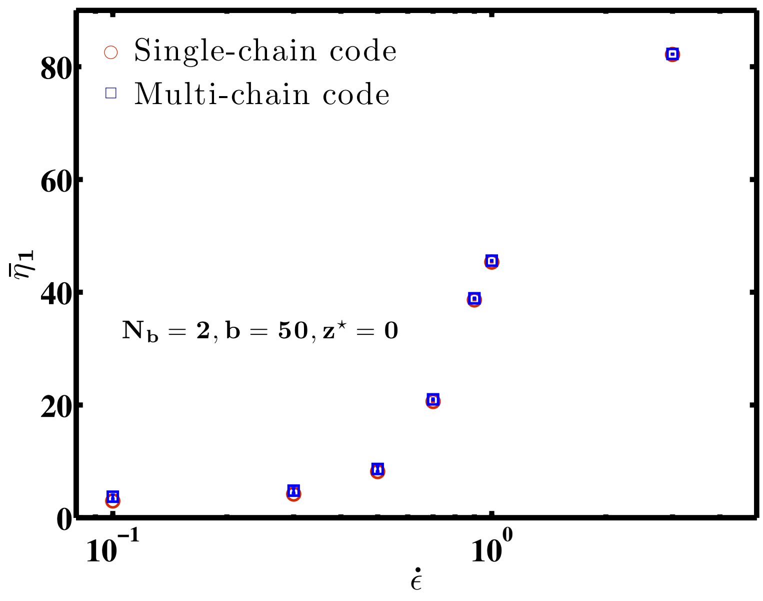


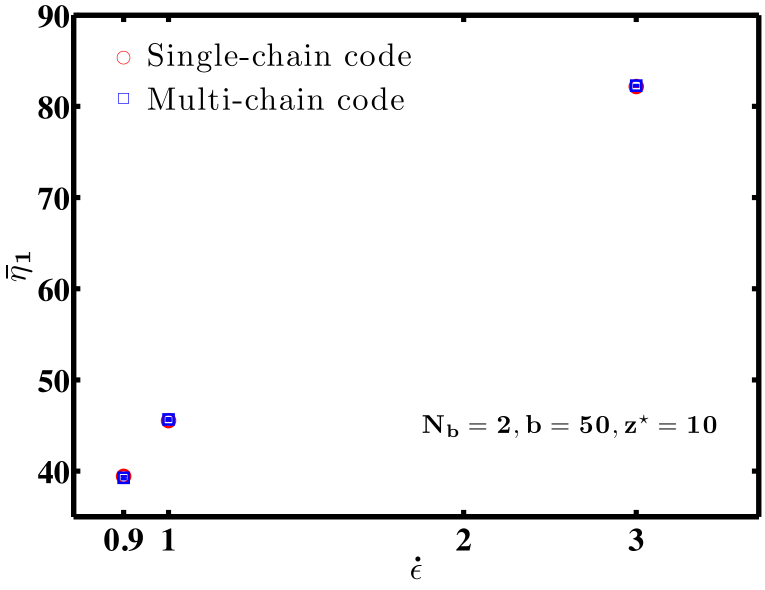
Figure 7.14: Illustration of inherent numerical instability in planar elongational flow simulations

and their images. This way, the numerical values of the coordinates of the beads never blow up and numerical instability is avoided. Note that in all our PEF simulations, we have used $k = 3$ and $N_{11} = 2$, required in calculating the strain period and the magic angle as discussed in Section 7.2.3.

We have performed multi-chain BD simulations for dumbbells with EV in the ultra dilute limit and compared our results with single-chain BD simulations. It is worth pointing out that for planar elongational flow simulations, spring forces cannot be modeled using the Hookean force law since this model



(a)



(b)

Figure 7.15: Comparison of $\bar{\eta}_1$, at various $\dot{\epsilon}$, predicted by the multi-chain BD algorithm with the results of single-chain BD simulations in the dilute limit: (a) for $z^\star = 0$ and (b) for $z^\star = 10$.

allows the molecule to be extended indefinitely, which is clearly physically unrealistic. The finite extensibility of the polymer becomes important in situations where a polymer molecule is likely to be close to full extension, such as in strong shear or elongational flows. We use FENE springs to model the spring forces in simulating planar elongational flow.

We have carried out multi-chain BD simulations to obtain the planar elongational flow viscosity $\bar{\eta}_1$ for a range of $\dot{\epsilon}$, for $z^\star = 0$ and for $z^\star = 10$. Note that $z^\star = 10$ corresponds to $z = 10 \times \sqrt{2} = 14.142$. We set the other parameter in EV potential, $d^\star$ to be equal to 1. Other than these parameters, we have used $N_c = 500$, $c/c^\star = 2 \times 10^{-16}$ and the FENE parameter $b_p = 50$. Simulation results for $z^\star = 0$ and 10 are shown in Figs. 7.15 (a) and (b), respectively, obtained by both multi-chain and single-chain simulations. Clearly, in both the cases, we can see the agreement between the multi-chain and single-chain BD simulation results for dilute solutions of FENE dumbbells.

This completes our discussion of validation studies for planar shear and planar elongational flow. Clearly, as the planar mixed flow is a linear combination of planar shear and elongational flows, the algorithm developed here can be used to predict various rheological properties of semidilute polymer solutions undergoing planar mixed flow. Some preliminary results are presented in the next section.

7.5 Preliminary Results and Discussion for Planar Mixed Flows

The purpose of this section is to show that we have a working algorithm to simulate semidilute polymer solutions at finite concentration, undergoing planar mixed flow. In order to show this, we consider a very simple system of FENE dumbbells ($N_b = 2$) at the overlap concentration ($c = c^\star$). With regard to microscopic interactions, for reasons discussed in the previous section, we do not account for hydrodynamic interactions in this study. However, we consider excluded volume interactions, which allows us to capture some of the interesting rheological behavior of polymer solutions at finite concentration in the mixed flow geometry. Various results, obtained by running BD simulations with the FENE parameter $b_p = 50$, and for excluded volume interaction parameters $z^\star = 10$ and $d^\star = 1$, are presented here. In discussing results for PMF, we have a choice of using either the pair $(\dot{\Gamma}, \chi)$, or $(\dot{\epsilon}, \dot{\gamma})$ as variables with which to explore the behavior of polymer solutions. Both give valuable insights. However, because of time constraints, only the latter pair of variables has been examined here.

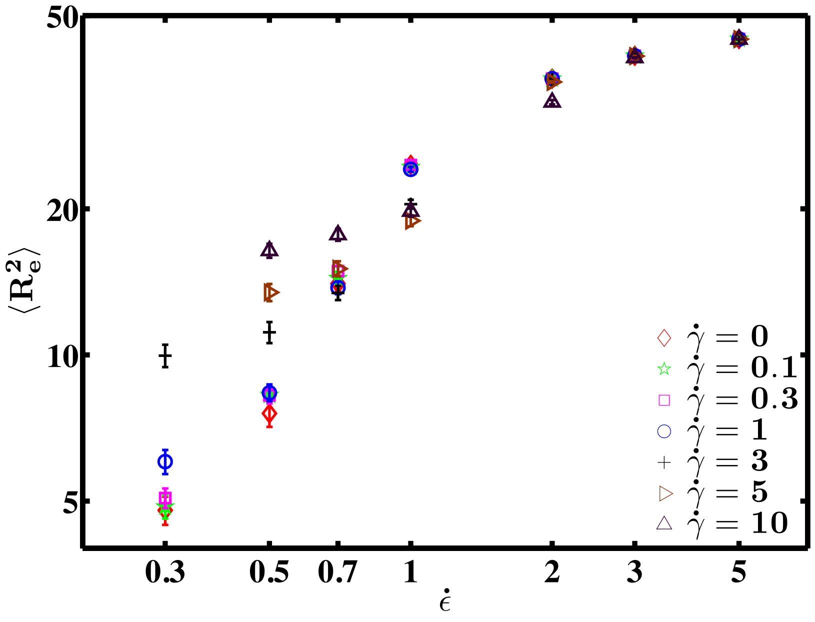
Prior to carrying out simulations for a finite concentration system, it is a prerequisite to know the value of $c^\star$, which is defined as $N_b / \left[\dfrac{4\pi}{3} (R_g^0)^3 \right]$, where R_g^0 is the gyration radius for an isolated chain at equilibrium. We carry out BD simulation for an ultra dilute system (as discussed in the previous section) at equilibrium for $N_b = 2$, $b_p = 50$, $z^\star = 10$ and $d^\star = 1$. Using a time step size of 0.005, the value of R_g^0 is found to be 1.2023 ± 0.0001, which is used to calculate $c^\star$ as $0.27473 \pm 9 \times 10^{-5}$. Using $c = c^\star$, BD simulations for PMF are then carried out for a range of shear rates $\dot{\gamma}$ and elongation rates $\dot{\epsilon}$. For the strain period and magic angle calculations, we use $k = 3$ and $N_{11} = 2$, which results in a magic angle $\theta \cong 31.72^0$. For better statistical accuracy of the various results, a large number of dumbbells ($N_c = 500 - 1000$), a sufficiently large number of independent trajectories (in the range of $500 - 1000$), and a long enough simulation length (500 dimensionless time units) are considered in our simulations.

Figure 7.16 (a) displays results for the mean square end-to-end distance $\langle R_e^2 \rangle$ as a function of elongation rate $\dot{\epsilon}$ at fixed values of shear rate $\dot{\gamma}$. The coil-stretch transition can be seen in Fig. 7.16 (a), where the dumbbells are coils for low $\dot{\epsilon}$, and become stretched objects at high elongation rates. At lower $\dot{\epsilon}$, $\langle R_e^2 \rangle$ increases with increasing $\dot{\gamma}$. However, at higher $\dot{\epsilon}$ the values of $\langle R_e^2 \rangle$ are independent of shear rate and asymptotically approach the nondimensional value of the square of the maximum stretch of the dumbbell, which is $b_p = 50$. Interestingly, at intermediate values of $\dot{\epsilon}$ the dependence of $\langle R_e^2 \rangle$ on $\dot{\gamma}$ is non-monotonic. This is seen more clearly by plotting $\langle R_e^2 \rangle$ as a function of $\dot{\gamma}$ at fixed values of $\dot{\epsilon}$ in Fig. 7.16 (b). For instance, at $\dot{\epsilon} = 0.7$, $\langle R_e^2 \rangle$ does not change with increasing shear rate until $\dot{\gamma} = 0.3$, while for $\dot{\gamma}$ between 0.3 and 3, $\langle R_e^2 \rangle$ decreases. Upon further increasing the shear rate, $\langle R_e^2 \rangle$ increases.

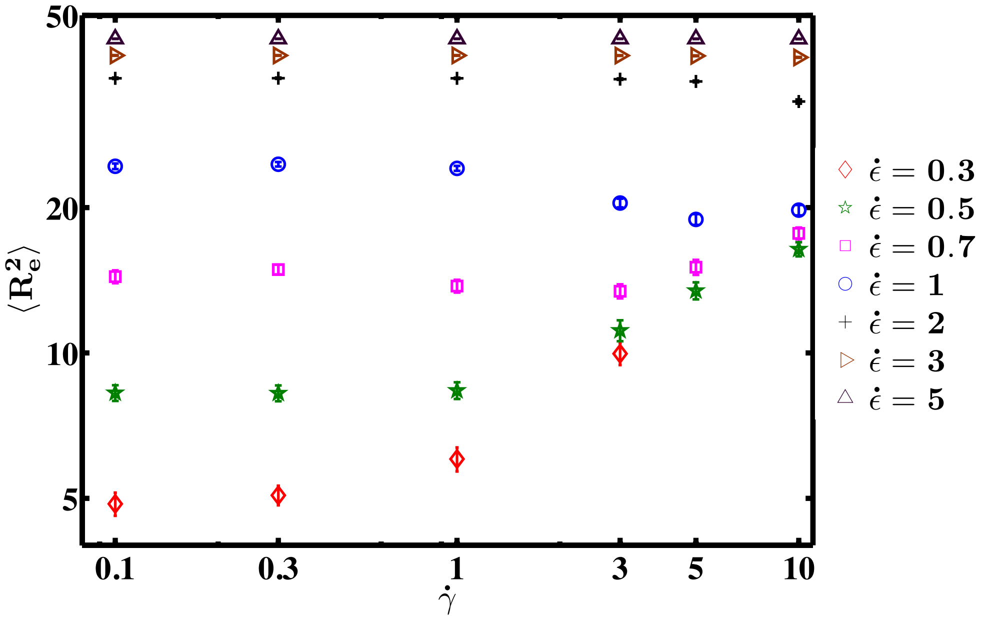
This behavior can be understood to arise as a result of competition between elongation and shear. Elongation tends to stretch a molecule in the flow direction, while shear tends to rotate and stretch a molecule through a cyclic tumbling motion. It appears that there are combinations of $\dot{\epsilon}$ and $\dot{\gamma}$ at which the mean size of the molecule exhibits non-monotonic behavior due to the effective orientation of the molecule in the flow. This non-monotonic behavior will probably not be observed if $\langle R_e^2 \rangle$ is plotted as a function of $\dot{\Gamma}$ for a fixed value of χ. However, this remains to be confirmed in the future.

A similar discussion of the coil-stretch transition and the competition between tumbling and stretching can be performed in the context of mean square gyration radius. Figure 7.17 shows the results for $\langle R_g^2 \rangle$ for the same set of $\dot{\gamma}$ and $\dot{\epsilon}$. Clearly, we can see the similar pattern in the results as observed for the case of $\langle R_e^2 \rangle$.

As discussed in Section 4.5.2, the PMF viscosity η can be calculated using Eq. (4.30), which involves the mixedness parameter χ. In order to calculate η, first the values of χ are calculated for a number of pairs of $\dot{\gamma}$ and $\dot{\epsilon}$ using Eqs. (4.26) and (4.27). Figure 7.18 displays a 3-D plot to show the values of χ for all the combinations of $\dot{\gamma}$ and $\dot{\epsilon}$. Clearly, the value of χ approaches unity as the ratio of $\dot{\epsilon}$ to $\dot{\gamma}$ becomes larger, which is the case of pure PEF. On the other hand,

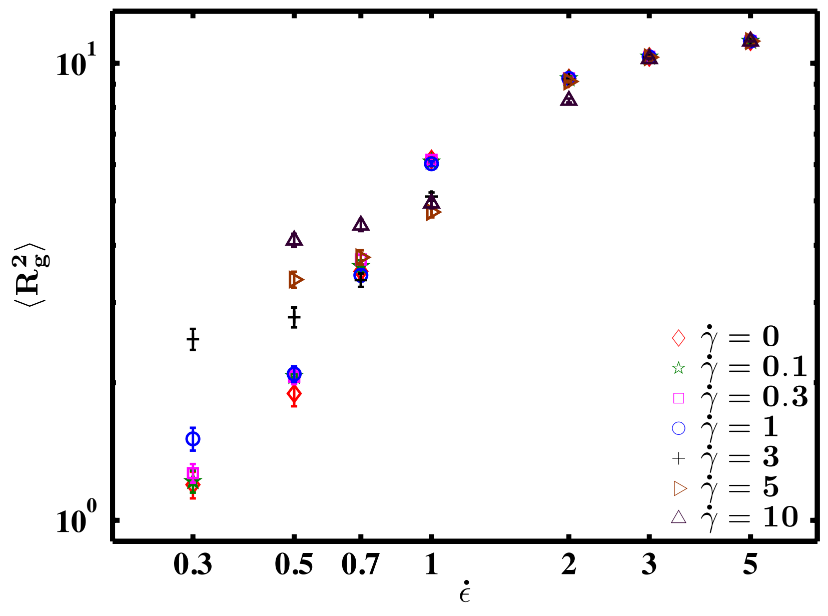


(a)

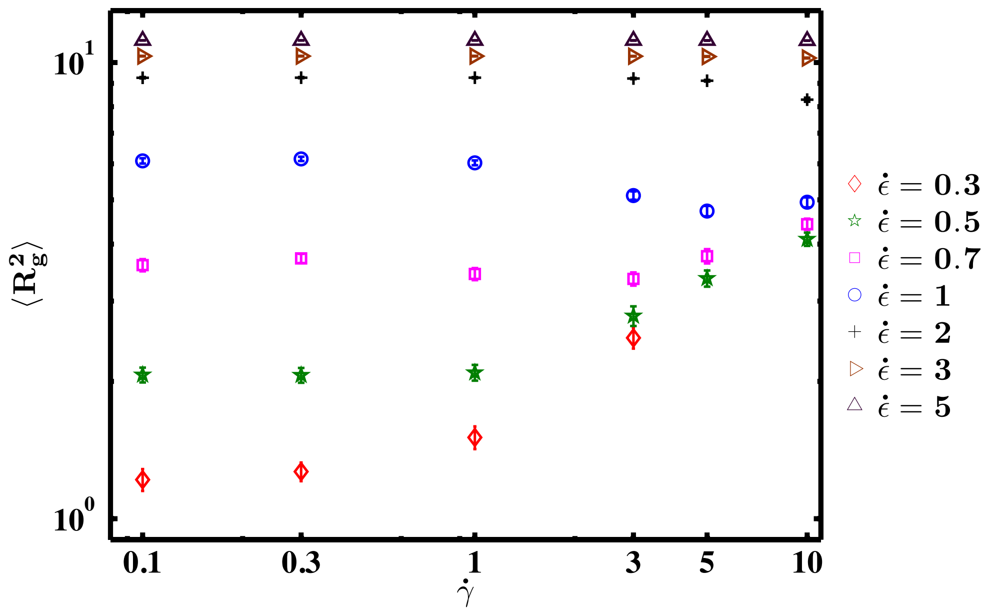


(b)

Figure 7.16: Mean square end-to-end distance of dumbbells of $b_p = 50$ in PMF at $c/c^{\star} = 1$: (a) $\langle R_e^2 \rangle$ as a function of $\dot{\epsilon}$ for various values of $\dot{\gamma}$. (b) $\langle R_e^2 \rangle$ as a function of $\dot{\gamma}$ for various values of $\dot{\epsilon}$.



(a)



(b)

Figure 7.17: Mean square gyration radius of dumbbells of $b_p = 50$ in PMF at $c/c^\star = 1$: (a) $\langle R_g^2 \rangle$ as a function of $\dot{\epsilon}$ for various values of $\dot{\gamma}$. (b) $\langle R_g^2 \rangle$ as a function of $\dot{\gamma}$ for various values of $\dot{\epsilon}$.

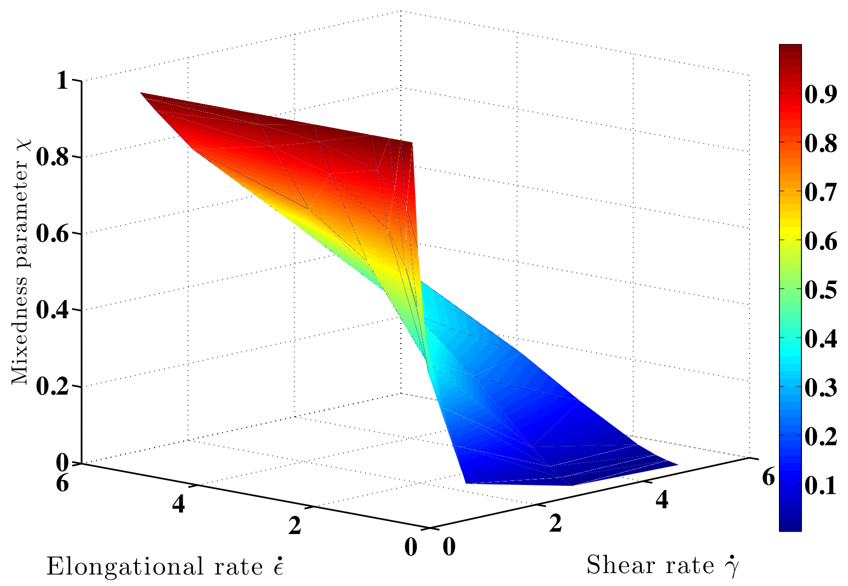


Figure 7.18: Mixedness parameter χ as function of $\dot{\gamma}$ and $\dot{\epsilon}$

$\chi \to 0$ as $\dfrac{\dot{\epsilon}}{\dot{\gamma}} \to 0$, which corresponds to pure PSF limit.

The PMF viscosity is then calculated using these values of χ for a number of values of $\dot{\gamma}$ and $\dot{\epsilon}$. Figure 7.19 (a) shows the results for η as a function of $\dot{\epsilon}$ at fixed values of $\dot{\gamma}$. Similar to the discussion in the context of Fig. 7.16, here also we can see the PMF viscosity corresponds to the coiled state at low $\dot{\epsilon}$, while it corresponds to the stretched state at high $\dot{\epsilon}$. It is worth pointing out that the PEF viscosity $\bar{\eta}_1$ for FENE dumbbells in dilute solutions at high $\dot{\epsilon}$ is given analytically by (Bird et al., 1987)

$$\bar{\eta}_1 = 2b_p(1 - \frac{b_p + 3}{2b_p}\frac{1}{\dot{\epsilon}} + \ldots) \tag{7.42}$$

which, in the limit $\dot{\epsilon} \to \infty$, becomes $\bar{\eta}_1 = 2b_p = 100$. Interestingly, Fig. 7.19 (a) shows that the PMF viscosity η seems to be approaching this value for high $\dot{\epsilon}$, irrespective of the shear rate. Note that for a given $\dot{\epsilon}$, the viscosity is lower as $\dot{\gamma}$

increases. This can be attributed to shear thinning.

The shear-thinning phenomena can be observed in Fig. 7.19 (b), which is a plot of η as function of $\dot{\gamma}$ for various values of $\dot{\epsilon}$. It is interesting to observe that the shear-thinning sets in later when the elongational rate is higher. For instance, for $\dot{\epsilon} = 5$, the shear-thinning starts at about $\dot{\gamma} = 1$, while it sets in at about $\dot{\gamma} = 0.3$ for $\dot{\epsilon} = 1$. The phenomena of shear-thinning is related to the alignment of polymer chains. Tumbling brought about by shear favors alignment. For higher $\dot{\epsilon}/\dot{\gamma}$ ratios, it appears that tumbling is hindered, and as a result alignment is weaker.

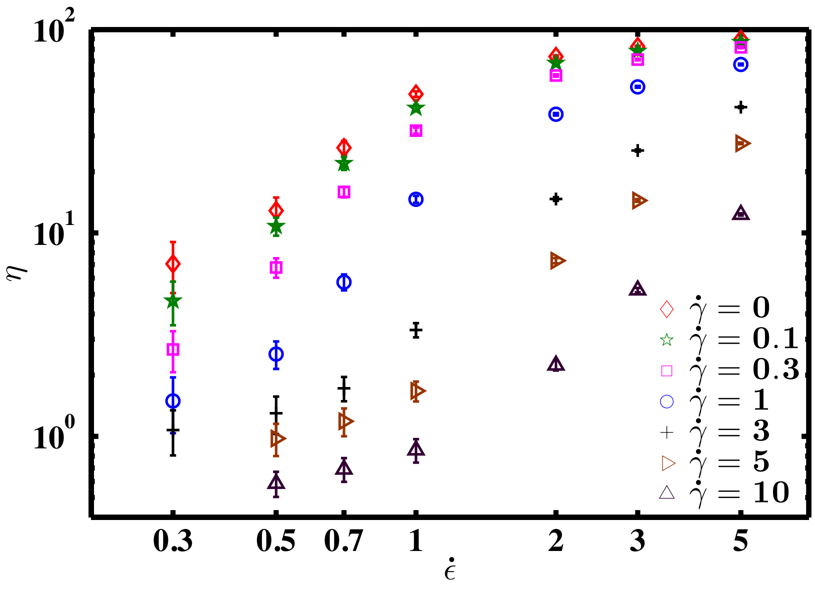
This completes our discussion on the few preliminary results obtained for semidilute polymer solutions of FENE dumbbells at overlap concentration undergoing PMF. These results which appear physically meaningful, give us confidence that the algorithm is correctly implemented, and paves the way to carry out more detailed studies of PMF in the future.

7.6 Conclusions

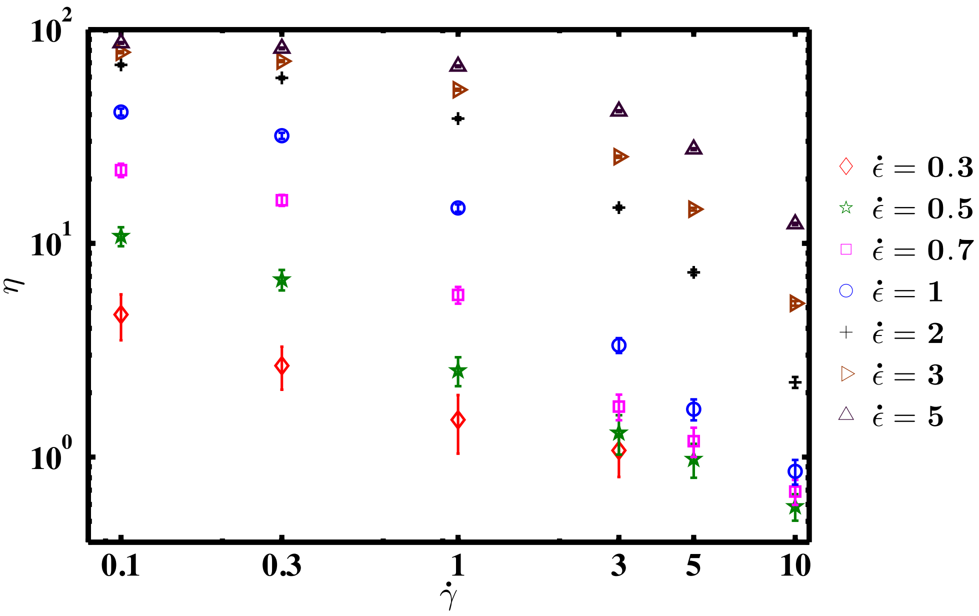
The implementation of a variety of flows in the context of a BD simulation algorithm for semidilute polymer solutions was presented in this chapter. Though, a detailed simulation study was not carried out using the algorithm we have developed, validation studies and some preliminary results show that this algorithm can indeed be used to study planar mixed flows in detail in the future. In summary,

1. Periodic boundary conditions for planar shear flow have been implemented using the Lees Edwards periodic boundary condition (Lees and Edwards, 1972).

2. The Kraynik-Reinelt periodic boundary condition (Kraynik and Reinelt,



(a)



(b)

Figure 7.19: PMF viscosity of a solution of dumbbells of $b_p = 50$ at $c/c^{\star} = 1$: (a) η as a function of $\dot{\epsilon}$ for various values of $\dot{\gamma}$. (b) η as a function of $\dot{\gamma}$ for various values of $\dot{\epsilon}$.

1992) has been used to implement planar elongational flow. This PBC has also been extended to the case of planar mixed flow, developed by Hunt et al. (2010) in the context of NEMD simulations.

3. The calculation of the Ewald sum for hydrodynamics interactions has been modified to account for a non-cubical simulation box.

4. Planar shear and planar elongational flows have been validated in the dilute concentration regime by comparing results of the multi-chain code with the results obtained from running single-chain BD simulations.

5. Some preliminary results have been obtained for FENE dumbbells at the overlap concentration for a range of shear and elongational rates.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The broad objective of this work has been to understand the behavior of polymer solutions in the semidilute regime at and far from equilibrium with the help of Brownian dynamics simulations. In particular, we have focused greater attention on understanding the concentration and solvent quality dependence of various static and dynamic properties at equilibrium. With regard to far from equilibrium systems, a BD algorithm for planar mixed flow, which is a linear combination of shear and elongational flows, was developed for semidilute polymer solutions. In particular, the following is a brief list of the key aspects of this thesis:

1. Semidilute polymer solutions have largely been understood through scaling theories, particularly in the limit of theta and good solvents. We have developed a scaling theory (Chapter 3) that is valid in the entire double crossover region of the phase diagram of polymer solutions.

2. Scaling theory has helped us arrive at the general conclusion that the

crossover scaling functions ϕ_R, ϕ_D and ϕ_η depend on the scaling variables z and $(c/c^\star)$. This can be tested either by experiments or computer simulations. However, scaling theory does not reveal the specific forms of these functions.

3. We have developed an optimized BD algorithm (Chapter 5) that is capable of (i) simulating semidilute polymer solutions with hydrodynamic and excluded volume interactions, and (ii) controlling solvent quality across the entire domain from theta to good solvents. We have also developed a systematic procedure for obtaining universal predictions in the long-chain limit.

4. With the help of the BD algorithm, we have shown the general validity of the scaling predictions, and in particular, we have teased out the specific dependence of the crossover scaling functions on z and $(c/c^\star)$ (Chapter 6). It appears that the dependence on $(c/c^\star)$ is identical to the good solvent limit, while the dependence on z comes in through an effective exponent.

It can be said that by these steps, a fairly complete understanding of the static and dynamic behavior of semidilute solutions has been achieved at equilibrium.

5. The challenge in the future is to study semidilute solutions far from equilibrium and to carry out a systematic comparison with experimental observations. Not much is currently understood of the behavior of semidilute polymer solutions across a range of concentrations. With a view to facilitating these studies in the future, we have extended the BD algorithm to describe planar mixed flows (Chapter 7). The advantage of our implementation is that it allows one to study the limit of planar shear and elongational flows

and all the combinations in between. Preliminary results show that our implementation is robust and agrees with earlier results where relevant.

8.2 Future Work

Some directions in which the current work can be taken further are listed below.

1. An improvement in the speed of the current BD algorithm is essential in order to be able to tackle the large number of long polymer chains that are required in order to study the universal behavior of semidilute solutions. There is plenty of scope for improvement of the current BD algorithm. Firstly, the algorithm is implemented based on a simple Euler integrator. A semi-implicit predictor-corrector integrator may lead to an increased time step, as observed previously for dilute polymer solutions. Secondly, a fast Fourier transform technique can be used to make the calculation of the Fourier space part of the Ewald sum much faster. This will lead to an algorithm of computational complexity of $O(N^{1.3} \log N)$ (Sierou and Brady, 2001; Banchio and Brady, 2003). Further, a Verlet table can be implemented to make the evaluation of the real space part of the Ewald sum more efficient. These improvements would make the current code competitive with other mesoscopic simulation algorithms.

2. The scaling theory in the present work is restricted to polymer solutions close to equilibrium. In the presence of flow, an additional scaling variable, namely, the Weissenberg number determines the behavior of polymer solutions. It is important to develop a scaling theory for flowing semidilute polymer solutions since it would guide both experiments and simulations, as demonstrated here at equilibrium. Clearly, the concept of a *Pincus blob* (de Gennes, 1979; Rubinstein and Colby, 2003), that determines relevant

length scales in the presence of flow would play an important role in such a scaling theory.

3. There is currently very little that is known experimentally about the behavior of semidilute solutions in flow, particularly across a range of molecular weights, concentrations and temperatures, except for some early work by Shaqfeh and co-workers (Hur et al., 2001). There is currently work in the rheology lab at Monash where benchmark data on semidilute solutions of DNA molecules is being acquired. The challenge for simulations would be to obtain a parameter free comparison with this experimental data. Such a comparison would establish that a predictive understanding of the behavior of semidilute solutions has been attained.

Appendix A

Averaging Methods

We have used the ensemble averaging method to estimate static properties and the sliding average method to estimate diffusivity. Details of both these averaging methods are discussed below.

A.1 The Ensemble Averaging Method

Average values and error bars of static properties are conventionally estimated based on the block averaging method (Rapaport, 2004). We tried the block averaging method to calculate the various static properties but it turned out that the block averaging method was not useful to us to obtain properties with acceptable error estimates. Below is a brief discussion to show why the block averaging method was unsuitable.

Consider a trajectory or time series of any property A containing N_s data points, which are divided into n_b blocks. Hence each block contains N_s/n_b data points. If A_{ij} is the j^{th} element in i^{th} block for A, an average over block i can be estimated as

$$\bar{A}_i = \frac{1}{(N_s/n_b)} \sum_{j=1}^{N_s/n_b} A_{ij} \qquad \text{(A.1)}$$

Consequently, the block averaging method estimates the average value of A with the expression

$$\langle A \rangle = \frac{1}{n_b} \sum_{i=1}^{n_b} \bar{A}_i \tag{A.2}$$

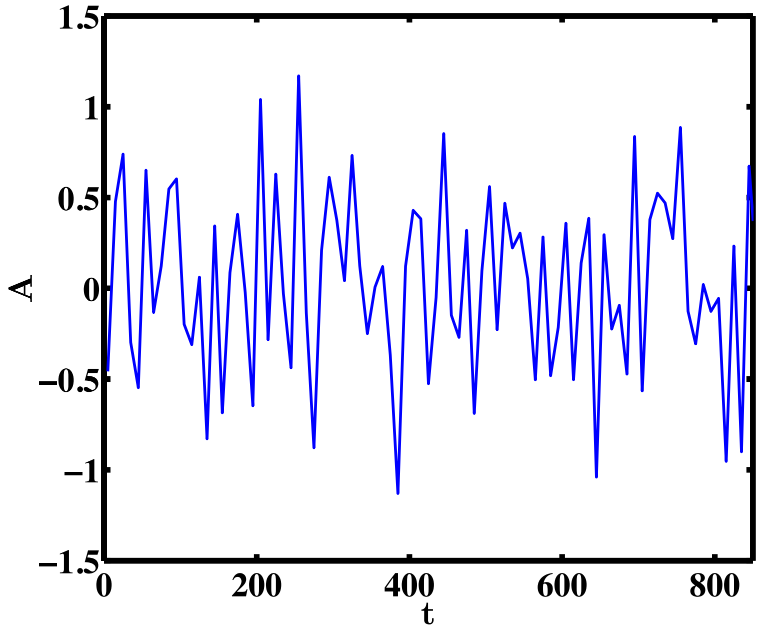
Note that the mean value of A is always the same, irrespective of the choice of n_b because the same data points are always used in the sum. However, the standard deviation, which is obtained by computing the variance of the block averages, depends on the choice of n_b. For any n_b, an estimate of the standard deviation of the mean can be shown to be

$$\sigma_A^b = \sqrt{\frac{1}{(n_b - 1)} \sum_{i=1}^{n_b} \bar{A}_i^{\,2} - \langle A \rangle^2} \tag{A.3}$$

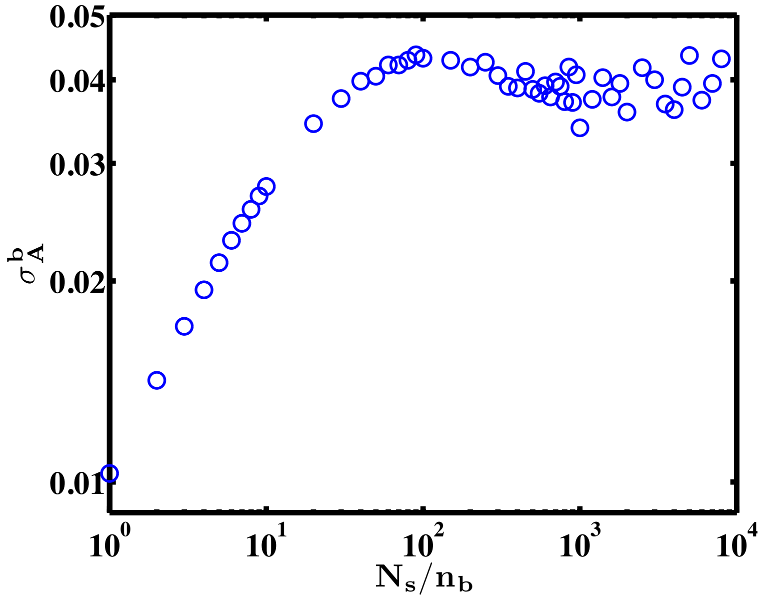
Clearly, one can envisage that the choice of n_b could affect the estimated value of the standard deviation. In order to illustrate this argument, the Ornstein−Uhlenbeck process (which is a stochastic process that describes the velocity of a massive Brownian particle under the influence of friction) is used as a tool. This process is also considered to be a modification of the random walk in continuous time, or Wiener process. The equation below represents the Ornstein−Uhlenbeck (OU) process

$$A(t + \Delta t) = A(t) + \alpha\,[\mu - A(t)]\,\Delta t + \sigma\, dW_t \tag{A.4}$$

where $\alpha > 0$, μ and σ are OU parameters, Δt is the time step and W_t denotes the Wiener process. Using Eq. (A.4), with $\alpha = 0.9$, $\mu = 0$, $\Delta t = 0.01$ and $\sigma = 1$, the process $A(t)$ is generated, of which a small snap shot is shown in Fig. A.1 (a). The error analysis for A is carried out using the block averaging method for different values of n_b. As anticipated, Fig. A.1 (b) shows that for a range of values of n_b, σ_A^b increases until a limiting value is reached, and this



(a)



(b)

Figure A.1: Demonstration of block averaging method using the Ornstein−Uhlenbeck process: (a) A typical trajectory in the Ornstein−Uhlenbeck process (b) The standard deviation reaches a limiting value.

value corresponds to the true standard deviation of the mean. If the number of data points are not sufficient, the limiting value is not reached. The error for small values of n_b indicates poor statistics.

In our case, the calculation of error estimates for $\langle \mathbf{R_e}^2 \rangle$ and $\langle \mathbf{R_g}^2 \rangle$ using the block averaging method failed because the standard deviation for both these properties never reached their corresponding limiting values. An example is shown in Fig. A.2 for the mean value and the standard deviation of $\langle \mathbf{R_e}^2 \rangle$, denoted as $\sigma^b_{\langle R_e^2 \rangle}$, as a function of block size. The failure of using the block averaging method for $\langle \mathbf{R_e}^2 \rangle$ and $\langle \mathbf{R_g}^2 \rangle$ stems from the fact that the current code is not able to generate sufficiently long trajectories, which is mainly because of the poor speed of the code and insufficient computational resources.

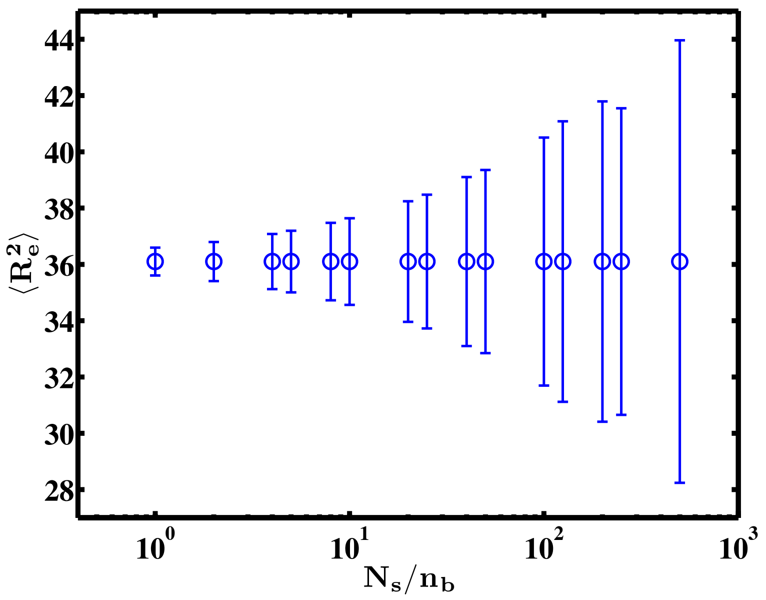
An alternative approach is used in this work to calculate the error estimates. In this approach many trajectories for any static property A are generated simultaneously and the time average of each trajectory is calculated. Next, an ensemble average of all these mean values is calculated, along with the standard error of the ensemble average obtained in this process as shown below

$$\langle A \rangle = \frac{1}{N_s\, N_T} \sum_{i=1}^{N_T} \sum_{j=1}^{N_s} A_{ij} \tag{A.5}$$

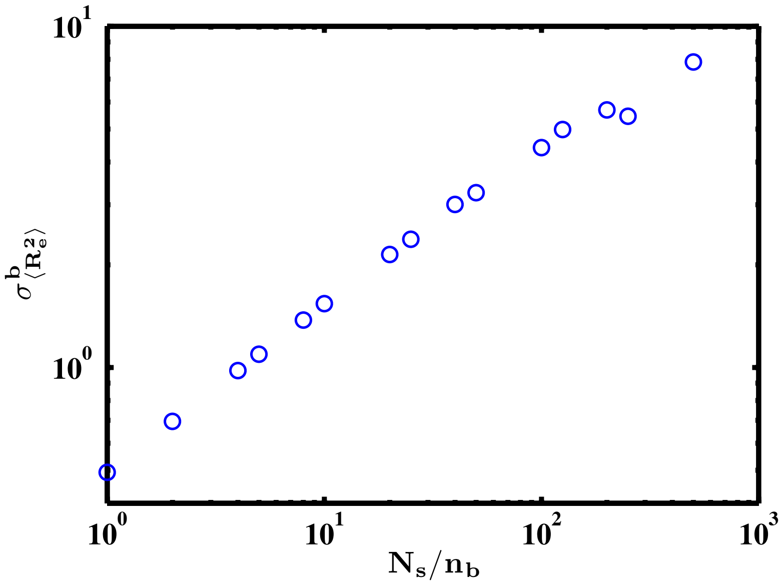
where, N_s is the number of data points in a trajectory, N_T is the number of independent trajectories and A_{ij} is the j^{th} element in the i^{th} trajectory. Also, the standard deviation on the mean is then estimated using the expression given below

$$\sigma_A = \sqrt{\frac{1}{(N_s\, N_T - 1)} \sum_{i=1}^{N_T} \sum_{j=1}^{N_s} \left(A_{ij} - \langle A \rangle\right)^2} \tag{A.6}$$

An example to calculate the mean value and the standard deviation of $\langle \mathbf{R_e}^2 \rangle$



(a)



(b)

Figure A.2: Illustration of block averaging method to calculate the mean value and the standard deviation of $\langle \mathbf{R_e}^2 \rangle$: (a) the mean value along with the standard deviation (b) the standard deviation does not reach a limiting value.

using this approach is shown in Fig. A.3. As expected, the standard deviation decreases by increasing N_T, however, it does not reach a limiting value. In this approach the length of simulation can be small (of the order of $10-70$ relaxation times) but the number of trajectories needs to be large to obtain highly accurate results.

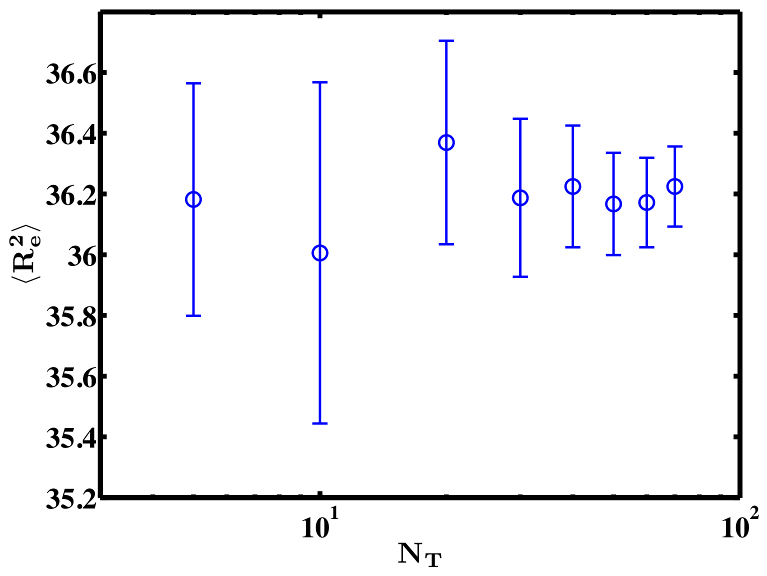
Comparing between Figs. A.2 (a) and A.3 (a), gives more confidence in the mean value of $\langle \mathbf{R_e}^2 \rangle$ when an ensemble averaging method is used. We have used the ensemble averaging method for estimating all the static properties.

A.2 The Sliding Average Method

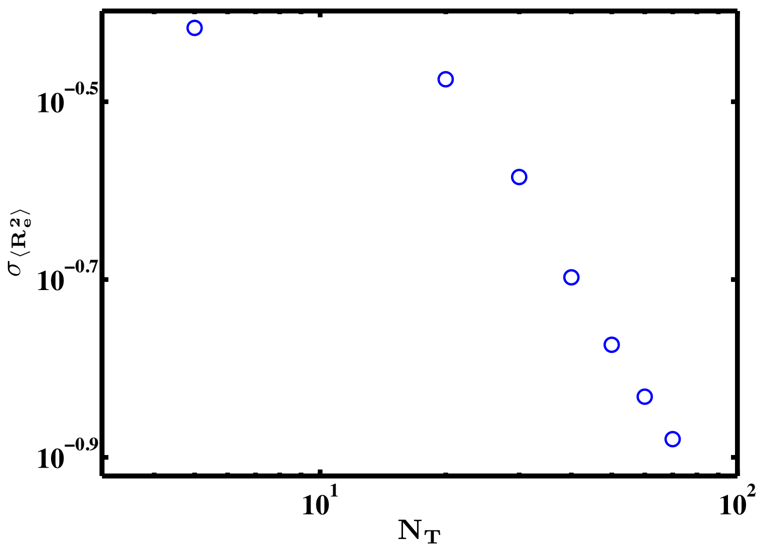
In the sliding average method, the time series of any property is analyzed by creating a series of averages of different subsets of the full time series. If the number of data points in a time series are N_s and the size of a subset is denoted by N_τ, then for a fixed N_τ, the mean-square-displacement MSD(τ), where $\tau = N_\tau \Delta t$, for a multi-chain system is given by

$$MSD(\tau) = \frac{k_B T}{\zeta} \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{1}{N_s - N_\tau} \sum_{j=1}^{N_s - N_\tau} \left\langle |\mathbf{r}^i{}_{\text{cm}}(j + N_\tau) - \mathbf{r}^i{}_{\text{cm}}(j)|^2 \right\rangle \tag{A.7}$$

Equation (A.7) is used to obtain $MSD(\tau)$ for $N_\tau = 1, 2, 3.....N_s$. In other words, we obtain the time series of MSD. Each stochastic trajectory (simulation run) leads to a time series for the MSD. An ensemble average over all the trajectories then gives the time series of the mean of the MSD (denoted here as MSD_{avg}) along with standard error. An example of MSD_{avg} for some values of τ in the interval $0 < \tau < 400$ is shown in Fig. A.4 for $z = 0.7, c/c^\star = 1, N_b = 6, N_c = 46, N_s = 20000$, where error bars on each data point is shown. Since the initial MSD_{avg} data represents transient short time diffusivity, while the data at large



(a)



(b)

Figure A.3: Illustration of ensemble averaging method to calculate the mean value and the standard deviation of $\langle \mathbf{R_e}^2 \rangle$: (a) the mean value along with the standard deviation, showing that the level of confidence increases as N_T increases (b) the standard deviation decreases with N_T but does not reach a limiting value.

times have large error bars (they are based on a larger τ or a smaller number of blocks in the sliding average method), typically the first 15-20% and last 20-30%

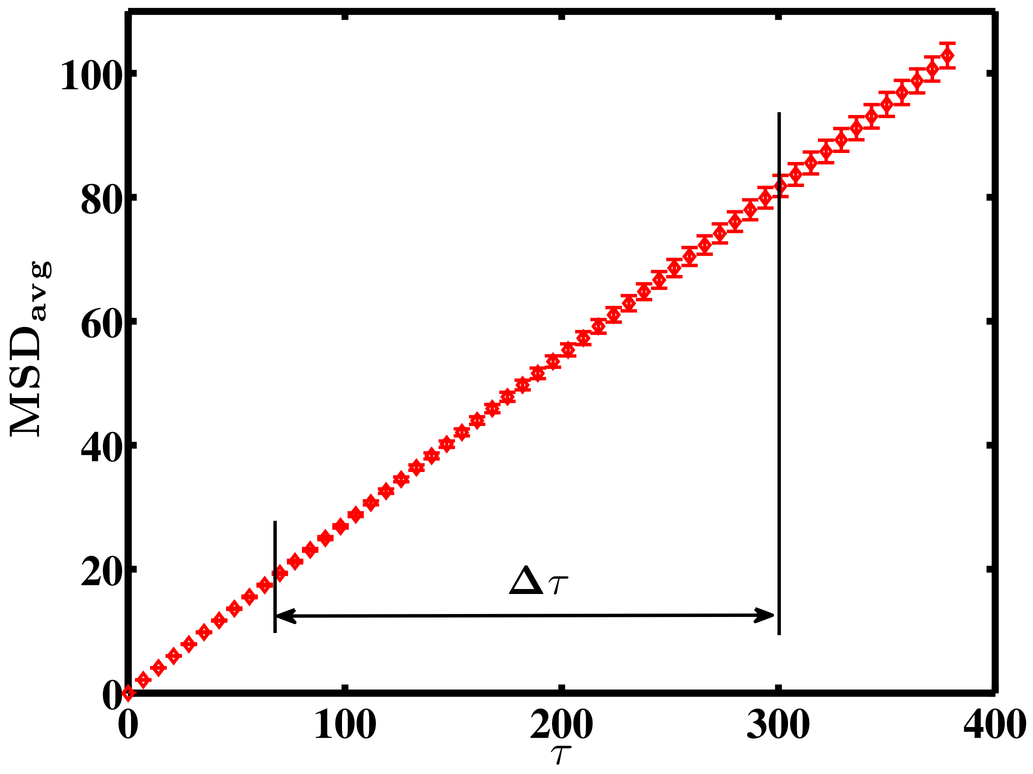


Figure A.4: Illustration of window of times Δ_τ in the MSD$_{\text{avg}}$ data

of the data are discarded. This leads to a window of times $\Delta\tau$ in the MSD$_{\text{avg}}$ data that can be fitted with a straight line. We use the window $\Delta\tau$ estimated in this way to find the diffusivity from the ensemble of MSD trajectories as follows. Basically, for a fixed set of parameters, the slopes of the lines fitted to each of the MSD trajectories in the ensemble, over the range of times $\Delta\tau$, is used to obtain an ensemble of predicted diffusion coefficients. The mean diffusion coefficient D and the standard error of mean is then determined from this ensemble.

Appendix B

Ewald Summation of Rotne-Prager-Yamakawa Mobility Tensor

Originally the Ewald summation method was developed for electrostatic interactions, therefore, a quick introduction to this method is first discussed in the context of electrostatic interactions. The remainder of this appendix then presents the derivation of the Ewald sum for hydrodynamic interactions based on the Rotne-Prager-Yamakawa mobility tensor.

B.1 Brief introduction to the Ewald Sum for Electrostatic Interactions

In a system of particles, the inter-particle force is defined to be short ranged if it decreases with distance faster than r^{-d} where d is the dimensionality of the system. In a system where there are long-range interactions such as Coulombic interactions, and a large number of particles, it becomes crucial to avoid the computing of all pair interactions, as otherwise the computational effort would be proportional to the square of the number of particles. If the electrostatic potential is truncated at a distance r_c, then the contribution of the tail of the potential $u(r)$ is (in 3-D),

$$U^{tail} = \frac{N\rho}{2} \int_{r_c}^{\infty} dr\, u(r)\, 4\, \pi\, r^2 \tag{B.1}$$

where N is the number of particles and ρ is the number density. This equation shows that the tail correction to the potential energy diverges, unless $u(r)$ decays faster than r^{-3}. This is why one cannot use a truncation procedure for long-ranged Coulombic interactions. In order to solve the problem of having to account for all the long-ranged interactions, the Ewald summation method is used (Frenkel and Smit, 2002). Physically the Ewald method works by surrounding each point charge in the system by a charge distribution of equal magnitude and opposite sign. This distribution is usually considered a Gaussian distribution, although this choice is arbitrary. The counter-charge screens the original potential and thus making it short-ranged. This is then summed in real space. Then a second imaginary charge distribution of opposite sign to the first (and of the same sign as the point charges) is added to cancel out the screening charges. We can call it the compensating charge distribution. As this screening distribution is a smooth function, its Fourier transform is rapidly convergent. Therefore, this second term is summed in reciprocal space. One more term, which does not involve any summation, takes care of undesired self-interactions. Details of the Ewald summation formula for electrostatic interactions are not covered here. However, the remaining sections of this appendix discuss the detailed derivation of Ewald summation for hydrodynamic interactions which was originally carried out by Beenakker (1986).

B.2 Rotne-Prager-Yamakawa (RPY) Tensor and the Lattice Sum

Consider a 3-D periodic lattice in which each unit cell (of nondimensional volume V, numbered by the index $\mathbf{n}$) contains N spherical particles (nondimensional radius a, numbered by the index μ). The position vector of particle μ is given by

$$\mathbf{r}_{\mu,\mathbf{n}} = \mathbf{r}_\mu + \mathbf{r}_\mathbf{n} \tag{B.2}$$

where, $\mathbf{r}_\mathbf{n}$ is the lattice vector. It is convenient to visualize a position vector with the help of Fig. 5.1 in Section 5.2.1. The force on a particle is denoted by $\mathbf{F}_\nu$ and it is assumed that the total force on all the particles in a unit cell vanishes:

$$\sum_{\mu=1}^{N} \mathbf{F}_\mu = \mathbf{0} \tag{B.3}$$

If μ is a given particle in the original cell $\mathbf{n_0}$ then the branch $\mathcal{A}$ of the RPY mobility tensor (in which the distance between two particles is more than the particle diameter) (Rotne and Prager, 1969; Yamakawa, 1970) is given by

$$\mathsf{D}_{\mu\mathbf{n_0}\,,\,\nu\mathbf{n}} = \left(\frac{3\,a}{4\,x}\left\{\mathsf{I} + \hat{\mathbf{x}}\hat{\mathbf{x}}\right\} + \frac{a^3}{2\,x^3}\left\{\mathsf{I} - 3\,\hat{\mathbf{x}}\hat{\mathbf{x}}\right\}\right) \qquad \text{for } (\nu,\mathbf{n}) \neq (\mu,\mathbf{n_0}) \tag{B.4}$$

$$\mathsf{D}_{\mu\mathbf{n_0}\,,\,\mu\mathbf{n_0}} = \mathsf{I} \tag{B.5}$$

Here the vector $\mathbf{x}$ (with magnitude x and unit vector $\hat{\mathbf{x}}$) represents the separation vector $\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}}$. For a periodic lattice, we can write the lattice sum for particle

μ as

$$\mathbf{S}_\mu = \sum_{\nu=1}^{N} \mathsf{D}_{\mu\nu} \cdot \mathbf{F}_\nu \tag{B.6}$$

For simplicity, we have represented the sum $\sum_{\mathbf{n}} \left(\sum_{\nu=1}^{N} \mathsf{D}_{\mu\mathbf{n_0}\,,\nu\mathbf{n}} \cdot \mathbf{F}_\nu \right)$ by the sum $\sum_{\nu=1}^{N} \mathsf{D}_{\mu\nu} \cdot \mathbf{F}_\nu$ in above equation to be consistent with Eq. (5.1), since both of the sums are equivalent. In this appendix, these two forms of the sum will be used interchangeably for convenience.

Because of the long-range of the RPY tensor, the sum in Eq. (B.6) converges only slowly. Using the Ewald summation method, this sum can be written as two fast converging sums. Note that the Ewald summation formula for the branch $\mathcal{A}$ (See Section 4.4) of the RPY mobility matrix will first be derived, as was done by Beenakker (1986). A correction term to account for the presence of the branch $\mathcal{B}$ (See Section 4.4) of the RPY tensor has already been derived in Section 5.2.2.

B.3 Derivation of the Ewald Sum of RPY Tensor

Following Beenakker (1986), an alternative representation of the two-sphere RPY tensor (Eq. (B.4)) is written as:

$$\begin{aligned}\mathsf{D}_{\mu\mathbf{n_0},\nu\mathbf{n}} = \left(\frac{3a}{4} + \frac{a^3}{4}\nabla^2 \right) (\nabla^2 \mathsf{I} - \boldsymbol{\nabla\nabla}) \mid \mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}} \mid \\ \text{for } (\nu, \mathbf{n}) \neq (\mu, \mathbf{n_0}),\end{aligned} \tag{B.7}$$

where $\boldsymbol{\nabla} = \partial/\partial(\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}})$. Now, exploiting the fact that the sum of the error function and the complementary error function is 1, Eq. (B.7) can be written as

$$\begin{aligned}\mathsf{D}_{\mu\mathbf{n_0}\,,\nu\mathbf{n}} = \mathsf{M}^{(1)}(\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}}) + \mathsf{M}^{(2)}(\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}}) \,, \\ \text{for } (\nu, \mathbf{n}) \neq (\mu, \mathbf{n_0}),\end{aligned} \tag{B.8}$$

where,

$$\mathbf{M}^{(1)}(\mathbf{r}) = \left(\frac{3\,a}{4} + \frac{a^3}{4}\nabla^2\right) \left(\nabla^2 \mathbf{I} - \boldsymbol{\nabla}\boldsymbol{\nabla}\right) \{r \operatorname{erfc}(\alpha r)\} \tag{B.9}$$

$$\mathbf{M}^{(2)}(\mathbf{r}) = \left(\frac{3a}{4} + \frac{a^3}{4}\nabla^2\right) (\nabla^2 \mathbf{I} - \boldsymbol{\nabla}\boldsymbol{\nabla})\{r \operatorname{erf}(\alpha r)\} \tag{B.10}$$

and α is an arbitrary parameter. Here, r is the magnitude of the vector $\mathbf{r} = \mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}}$. Upon substituting the decomposition Eq. (B.8) into Eq. (B.6), the lattice sum becomes,

$$\begin{aligned}\mathbf{S}_\mu = \mathbf{F}_\mu + \left[\sum_{\mathbf{n}}{}' \sum_{\nu=1}^{N} \left(\mathbf{M}^{(1)}(\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}}) \cdot \mathbf{F}_\nu\right)\right] - \left[\mathbf{M}^{(2)}(\mathbf{r}=0) \cdot \mathbf{F}_\mu\right] \\ + \left[\sum_{\mathbf{n}} \sum_{\nu=1}^{N} \left(\mathbf{M}^{(2)}(\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}}) \cdot \mathbf{F}_\nu\right)\right] ,\end{aligned} \tag{B.11}$$

The first summation on the right-hand side of Eq. (B.11) is rapidly converging. The $'$ over the sum indicates that the terms with self-interactions in the original cell $\mathbf{n_0}$ are to be omitted. If the self-interactions are not excluded, then the second summation on the right-hand side is perfectly continuous and periodic and hence it converges rapidly on the reciprocal lattice. However, as the self-interactions lead to spurious results, there is a requirement of a term to correct this. The third term on the right-hand side compensates for self-interactions, and is subtracted from the Ewald sum as shown in Eq. (B.11). The transformation to reciprocal space is performed by means of the formula given by Nijboer and de Wette (1957)

$$\sum_{\mathbf{n}} g(\mathbf{r}_{\mathbf{n}}) = \frac{1}{V} \sum_{\lambda=-\infty}^{\infty} g(\mathbf{k}_\lambda) \tag{B.12}$$

with the Fourier transform of a function g defined by

$$g(\mathbf{k}) = \int d\mathbf{r}\ \exp(\mathrm{i}\mathbf{k}\cdot\mathbf{r})\ g(\mathbf{r}) \tag{B.13}$$

We may, therefore, write

$$\sum_{\mathbf{n}}\sum_{\nu=1}^{N}\mathsf{M}^{(2)}(\mathbf{r}_{\nu\mathbf{n}}-\mathbf{r}_{\nu\mathbf{n_0}})\cdot\mathbf{F}_\nu=\frac{1}{V}\sum_{\lambda}\sum_{\nu=1}^{N}\left(e^{-\mathrm{i}\mathbf{k}_\lambda\cdot(\mathbf{r}_\nu-\mathbf{r}_\mu)}\;\mathsf{M}^{(2)}(\mathbf{k}_\lambda)\cdot\mathbf{F}_\nu\right) \tag{B.14}$$

where terms with $\mathbf{k}_\lambda=\mathbf{0}$ are excluded in the sum in Eq. (B.14) by virtue of Eq. (B.3). Now the main focus is on calculating $\mathsf{M}^{(1)}(\mathbf{r})$, $\mathsf{M}^{(2)}(\mathbf{k})$ and $\mathsf{M}^{(2)}(\mathbf{r}=0)$.

B.3.1 Calculation of $\mathsf{M}^{(1)}(\mathbf{r})$

For the sake of simplicity, let us consider that $\frac{3a}{4}=A$, $\frac{a^3}{4}=B$ and r erfc$(\alpha r)=f(r)=f$, then Eq. (B.9) can be written in index notation as

$$\begin{aligned}\mathsf{M}^{(1)}(\mathbf{r})&=\left[(A+B\,\nabla^2)\;(\nabla^2\mathbf{I}-\boldsymbol{\nabla}\boldsymbol{\nabla})\right]\,f\\&=\left[A\;({\partial_k}^2\delta_{ij}-\partial_i\partial_j)\right]\,f\;+\left[B\;{\partial_k}^2({\partial_k}^2\delta_{ij}-\partial_i\partial_j)\right]\,f\end{aligned} \tag{B.15}$$

where, $\partial_k=\partial/\partial x_k$. Now, each term of Eq. (B.15) can be calculated one by one:

$$\begin{aligned}\partial_i\,\partial_j\,f(r)&=\partial_i\left(f'\frac{r_j}{r}\right)\\&=f'\,\partial_i\left(\frac{r_j}{r}\right)\;+\;\frac{r_j}{r}\,f''\left(\frac{r_i}{r}\right)\\&=\frac{f'}{r}\delta_{ij}\;-\;\frac{f'}{r^2}\,r_j\,\frac{r_i}{r}\;+\;f''\,\frac{r_ir_j}{r^2}\\&=\delta_{ij}\,\left\{\frac{f'}{r}\right\}\;+\;\frac{r_ir_j}{r^2}\,\left\{f''-\frac{f'}{r}\right\}\end{aligned} \tag{B.16}$$

$$\begin{aligned}{\partial_k}^2\,\delta_{ij}\,f(r)&=\delta_{ij}\,\partial_k\left(f'\frac{r_k}{r}\right)\\&=\delta_{ij}\left\{\frac{r_k}{r}\,f''\,\frac{r_k}{r}\;+\;f'\,\partial_k\left(\frac{r_k}{r}\right)\right\}\\&=\delta_{ij}\left\{f''\;+\;3\frac{f'}{r}\;-\;\frac{f'}{r^2}\,r_k\,\frac{r_k}{r}\right\}\\&=\delta_{ij}\left\{f''+\frac{2f'}{r}\right\}\end{aligned} \tag{B.17}$$

Subtracting Eq (B.16) from Eq (B.17) leads to

$$\left(\partial_k{}^2\,\delta_{ij} - \partial_i\,\partial_j\right) f = \delta_{ij}\left\{f'' + \frac{f'}{r}\right\} + \frac{r_i r_j}{r^2}\left\{\frac{f'}{r} - f''\right\} \tag{B.18}$$

which is the first part of Eq. (B.15). The second part of Eq. (B.15) requires the second derivative of the left hand side of Eq. (B.18). Hence,

$$\begin{aligned}\partial_k{}^2\left(\partial_k{}^2\,\delta_{ij} - \partial_i\,\partial_j\right) f &= \partial_k{}^2\left(\delta_{ij}\left\{f'' + \frac{f'}{r}\right\} + \frac{r_i r_j}{r^2}\left\{\frac{f'}{r} - f''\right\}\right)\\ &= \delta_{ij}\,\partial_k{}^2\left(f'' + \frac{f'}{r}\right) + \frac{r_i r_j}{r^2}\,\partial_k{}^2\left(\frac{f'}{r} - f''\right)\\ &+ \left(\frac{f'}{r} - f''\right)\partial_k{}^2\left(\frac{r_i r_j}{r^2}\right)\end{aligned} \tag{B.19}$$

The derivatives of each of the three terms on the right hand side of Eq. (B.19) will be calculated one by one.

(i) Derivative in the first term on the right hand side of Eq. (B.19):

Let us consider that $g_1 = f'' + \frac{f'}{r}$. From Eq. (B.17), we know that $\partial_k{}^2(g) = g'' + \frac{2g'}{r}$ for any function $g(r)$. Therefore,

$$\begin{aligned}\partial_k{}^2\left(f'' + \frac{f'}{r}\right) &= \partial_k{}^2\left(g_1\right)\\ &= g_1{}'' + \frac{2g_1{}'}{r}\\ &= \left(f'''' + \frac{f'''}{r} - \frac{2f''}{r^2} + \frac{2f'}{r^3}\right) + \left(\frac{2f'''}{r} + \frac{2f''}{r^2} - \frac{2f'}{r^3}\right)\\ &= f'''' + \frac{3f'''}{r}\end{aligned} \tag{B.20}$$

(ii) Derivative in the second term on the right hand side of Eq. (B.19):

Let us consider that $g_2 = \frac{f'}{r} - f''$. From Eq. (B.17), it follows that,

$$\begin{aligned}
{\partial_k}^2 \left(\frac{f'}{r} - f'' \right) &= {\partial_k}^2 \left(g_2 \right) \\
&= {g_2}'' + \frac{2{g_2}'}{r} \\
&= \left(-f'''' + \frac{f'''}{r} - \frac{2f''}{r^2} + \frac{2f'}{r^3} \right) + \left(-\frac{2f'''}{r} + \frac{2f''}{r^2} - \frac{2f'}{r^3} \right) \\
&= -f'''' - \frac{f'''}{r}
\end{aligned} \tag{B.21}$$

(iii) Derivative in the third term on the right hand side of Eq. (B.19):

$$\begin{aligned}
{\partial_k}^2 \left(\frac{r_i r_j}{r^2} \right) &= \partial_k \left(\frac{\delta_{ki}\, r_j}{r^2} + \frac{\delta_{kj}\, r_i}{r^2} - r_i\, r_j\, \frac{2}{r^3}\, \frac{r_k}{r} \right) \\
&= \frac{\delta_{ki}\, \delta_{kj}}{r^2} - \frac{2\,\delta_{ki}\, r_j}{r^3}\, \frac{r_k}{r} + \frac{\delta_{kj}\, \delta_{ki}}{r^2} - \frac{2\,\delta_{kj}\, r_i}{r^3}\, \frac{r_k}{r} \\
&\quad - \frac{6\, r_i\, r_j}{r^4} + \frac{8\, r_i\, r_j\, r_k}{r^5}\, \frac{r_k}{r} - \frac{2\, r_k}{r^4}\, r_i\, \delta_{kj} - \frac{2\, r_k}{r^4}\, r_j\, \delta_{ki} \\
&= \frac{\delta_{ij}}{r^2} - \frac{2\, r_i\, r_j}{r^4} + \frac{\delta_{ij}}{r^2} - \frac{2\, r_i\, r_j}{r^4} - \frac{6\, r_i\, r_j}{r^4} + \frac{8\, r_i\, r_j}{r^4} - \frac{2\, r_i r_j}{r^4} - \frac{2\, r_i\, r_j}{r^4} \\
&= \delta_{ij} \left\{ \frac{2}{r^2} \right\} - \frac{r_i\, r_j}{r^2} \left\{ \frac{6}{r^2} \right\}
\end{aligned} \tag{B.22}$$

Therefore, Eq. (B.19) now becomes,

$$\begin{aligned}
{\partial_k}^2 \left({\partial_k}^2\, \delta_{ij} - \partial_i\, \partial_j \right) f &= \delta_{ij} \left\{ f'''' + \frac{3\, f'''}{r} \right\} - \frac{r_i\, r_j}{r^2} \left\{ f'''' + \frac{f'''}{r} \right\} \\
&\quad + \left(\frac{f'}{r} - f'' \right) \left(\delta_{ij} \left\{ \frac{2}{r^2} \right\} - \frac{r_i\, r_j}{r^2} \left\{ \frac{6}{r^2} \right\} \right) \\
&= \delta_{ij} \left\{ f'''' + \frac{3\, f'''}{r} + \frac{2\, f'}{r^3} - \frac{2\, f''}{r^2} \right\} \\
&\quad + \frac{r_i\, r_j}{r^2} \left\{ -f'''' - \frac{f'''}{r} - \frac{6\, f'}{r^3} + \frac{6\, f''}{r^2} \right\}
\end{aligned} \tag{B.23}$$

which can be used to obtain the second term on the right hand side of Eq. (B.15). Hence, combining Eqs. (B.18) and (B.23), Eq. (B.15) can be written as

$$\begin{aligned}\mathbf{M}^{(1)}(\mathbf{r}) &= \delta_{ij}\left\{A\,f'' + \frac{A\,f'}{r}\right\} + \frac{r_i\,r_j}{r^2}\left\{\frac{A\,f'}{r} - A\,f''\right\} \\ &+ \delta_{ij}\left\{B\,f'''' + \frac{3\,B\,f'''}{r} + \frac{2\,B\,f'}{r^3} - \frac{2\,B\,f''}{r^2}\right\} \\ &+ \frac{r_i\,r_j}{r^2}\left\{-B\,f'''' - \frac{B\,f'''}{r} + \frac{6\,B\,f''}{r^2} - \frac{6\,B\,f'}{r^3}\right\} \\ &= \delta_{ij}\left\{A\,f'' + \frac{A\,f'}{r} + B\,f'''' + \frac{3\,B\,f'''}{r} + \frac{2\,B\,f'}{r^3} - \frac{2\,B\,f''}{r^2}\right\} \\ &+ \frac{r_i\,r_j}{r^2}\left\{\frac{A\,f'}{r} - A\,f'' - B\,f'''' - \frac{B\,f'''}{r} - \frac{6\,B\,f'}{r^3} + \frac{6\,B\,f''}{r^2}\right\}\end{aligned} \tag{B.24}$$

Using the definition of $f = r\ \mathrm{erfc}(\alpha r)$, we can evaluate the derivatives of f as follows

$$f' = \mathrm{erfc}(\alpha r) - 2\,\alpha\,r\,\beta \tag{B.25}$$

$$f'' = \left(4\,\alpha^3\,r^2 - 4\,\alpha\right)\,\beta \tag{B.26}$$

$$f''' = \left(16\,\alpha^3\,r - 8\,\alpha^5\,r^3\right)\,\beta \tag{B.27}$$

$$f'''' = \left(16\,\alpha^7\,r^4 - 56\,\alpha^5\,r^2 + 16\,\alpha^3\right)\,\beta \tag{B.28}$$

where $\beta = \frac{\exp(-\alpha^2 r^2)}{\sqrt{\pi}}$. Substitution of f', f'', f''', f'''', A and B into Eq. (B.24) leads to

$$\begin{aligned}
\mathbf{M}^{(1)}(\mathbf{r}) = \mathbf{I}\, \Bigg[& 3\,a\,\alpha^3\,r^2\,\beta \,-\, 3\,a\,\alpha\,\beta \,+\, \frac{3\,a}{4r}\,\mathrm{erfc}(\alpha\,r) \,-\, \frac{3\,a\,\alpha\,\beta}{2} \,+\, 4\,a^3\,\alpha^7\,r^4\,\beta \\
& -14\,a^3\,\alpha^5\,r^2\,\beta \,+\, 4\,a^3\,\alpha^3\,\beta \,-\, 6\,a^3\,\alpha^5\,\beta \,-\, 6\,a^3\,\alpha^5\,r^2\,\beta \,+\, 12\,a^3\,\alpha^3\,\beta \\
& + \frac{a^3\,\mathrm{erfc}(\alpha\,r)}{2\,r^3} \,-\, \frac{\alpha\,a^3\,\beta}{r^2} \,-\, 2\,a^3\,\alpha^3\,\beta \,+\, \frac{2\,a^3\,\alpha\,\beta}{r^2}\Bigg] \\
& + \hat{\mathbf{r}}\,\hat{\mathbf{r}}\,\Bigg[\frac{3\,a}{4\,r}\,\mathrm{erfc}(\alpha\,r) \,-\, \frac{3\,a\,\alpha\,\beta}{2} \,-\, 3\,a\,\alpha^3\,r^2\,\beta \,+\, 3\,a\,\alpha\,\beta \\
& -\, 4\,a^3\,\alpha^7\,r^4\,\beta \,+\, 14\,a^3\,\alpha^5\,r^2\,\beta \,-\, 4\,a^3\,\alpha^3\,\beta \,+\, 2\,a^3\,\alpha^5\,r^2\,\beta \,-\, 4\,a^3\,\alpha^3\,\beta \\
& -\, \frac{3\,a^3}{2\,r^3}\,\mathrm{erfc}(\alpha\,r) \,+\, \frac{3\,a^3\,\alpha\,\beta}{r^2} \,+\, 6\,a^3\,\alpha^3\,\beta \,-\, \frac{6\,a^3\,\alpha\,\beta}{r^2}\Bigg] \\
\\
= \mathbf{I}\, \Bigg[& \mathrm{erfc}(\alpha\,r)\,\left(\frac{3\,a}{4\,r} \,+\, \frac{a^3}{2\,r^3}\right) \,+\, \frac{\exp(-\alpha^2\,r^2)}{\sqrt{\pi}}\,\Bigg(3\,a\,\alpha^3\,r^2 \,-\, \frac{9\,a\,\alpha}{2} \\
& +\, 4\,a^3\,\alpha^7\,r^4 \,-\, 20\,a^3\,\alpha^5\,r^2 \,+\, 14\,a^3\,\alpha^3 \,+\, \frac{a^3\,\alpha}{r^2}\Bigg)\Bigg] \\
& + \hat{\mathbf{r}}\hat{\mathbf{r}}\,\Bigg[\mathrm{erfc}(\alpha\,r)\,\left(\frac{3\,a}{4\,r} \,-\, \frac{3\,a^3}{2\,r^3}\right) \,+\, \frac{\exp(-\alpha^2\,r^2)}{\sqrt{\pi}}\,\Bigg(\frac{3\,a\,\alpha}{2} \,-\, 3\,a\,\alpha^3\,r^2 \\
& -\, 4\,a^3\,\alpha^7\,r^4 \,+\, 16\,a^3\,\alpha^5\,r^2 \,-\, 2\,a^3\,\alpha^3 \,-\, \frac{3\,a^3\,\alpha}{r^2}\Bigg)\Bigg]
\end{aligned} \tag{B.29}$$

where $\hat{\mathbf{r}}$ is unit vector in the direction of $\mathbf{r}$. This completes the derivation of $\mathbf{M}^{(1)}(\mathbf{r})$ required for the real space sum.

B.3.2 Calculation of $\mathsf{M}^{(2)}(\mathbf{k})$

To derive $\mathbf{M}^{(2)}(\mathbf{k})$, the Fourier transform of $\mathbf{M}^{(2)}(\mathbf{r})$ using Eq. (B.10) is performed. Considering $r\ \mathrm{erf}(\alpha r) = g(r) = g$, we can write,

$$\mathbf{M}^{(2)}(\mathbf{k}) = \int d\mathbf{r}\ \exp(\mathrm{i}\mathbf{k}.\mathbf{r})\,\left(\frac{3\,a}{4} \,+\, \frac{a^3}{4}\,\nabla^2\right)\,\left(\nabla^2\,\mathbf{I} \,-\, \boldsymbol{\nabla}\,\boldsymbol{\nabla}\right)\,g \tag{B.30}$$

Partial integration of Eq. (B.30) results in

$$\mathbf{M}^{(2)}(\mathbf{k}) = \int d\mathbf{r}\, g \left(\frac{3\,a}{4} + \frac{a^3}{4}\nabla^2\right) \left(\nabla^2\,\mathbf{I} - \boldsymbol{\nabla}\,\boldsymbol{\nabla}\right) \exp\left(\mathrm{i}\mathbf{k}.\mathbf{r}\right) \tag{B.31}$$

Note that the constant term obtained during partial integration is zero. Let us now simplify $\left(\frac{3\,a}{4} + \frac{a^3}{4}\nabla^2\right) \left(\nabla^2\,\mathbf{I} - \boldsymbol{\nabla}\,\boldsymbol{\nabla}\right) \exp\left(\mathrm{i}\mathbf{k}.\mathbf{r}\right)$,

$$\begin{aligned}\left(\nabla^2\,\mathbf{I} - \boldsymbol{\nabla}\,\boldsymbol{\nabla}\right) \exp\left(\mathrm{i}\mathbf{k}.\mathbf{r}\right) &= \left(\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\,\mathbf{I} - \boldsymbol{\nabla}\,\boldsymbol{\nabla}\right) \exp\left(\mathrm{i}\mathbf{k}.\mathbf{r}\right)\\ &= \left(\mathrm{i}\,\mathbf{k}\cdot\mathrm{i}\,\mathbf{k}\,\mathbf{I} - \mathrm{i}\,\mathbf{k}\;\mathrm{i}\,\mathbf{k}\right) \exp\left(\mathrm{i}\mathbf{k}.\mathbf{r}\right)\\ &= \left(-k^2\,\mathbf{I} + \mathbf{k}\,\mathbf{k}\right) \exp\left(\mathrm{i}\mathbf{k}.\mathbf{r}\right)\end{aligned} \tag{B.32}$$

Hence,

$$\begin{aligned}\left(\frac{3\,a}{4} + \frac{a^3}{4}\nabla^2\right) \left(\nabla^2\,\mathbf{I} - \boldsymbol{\nabla}\,\boldsymbol{\nabla}\right) \exp\left(\mathrm{i}\mathbf{k}.\mathbf{r}\right) &= \frac{3\,a}{4}\left(-k^2\,\mathbf{I} + \mathbf{k}\,\mathbf{k}\right) \exp\left(\mathrm{i}\mathbf{k}\cdot\mathbf{r}\right)\\ &\quad + \frac{a^3}{4}\left(k^4\,\mathbf{I} - k^2\,\mathbf{k}\,\mathbf{k}\right) \exp\left(\mathrm{i}\mathbf{k}\cdot\mathbf{r}\right)\\ &= k^2\left\{\frac{3\,a}{4}\left(-\mathbf{I} + \hat{\mathbf{k}}\,\hat{\mathbf{k}}\right) + \frac{a^3\,k^2}{4}\left(\mathbf{I} - \hat{\mathbf{k}}\,\hat{\mathbf{k}}\right)\right\} \exp\left(\mathrm{i}\mathbf{k}\cdot\mathbf{r}\right)\\ &= -\frac{3\,k^2}{4}\left\{\left(\mathbf{I} - \hat{\mathbf{k}}\,\hat{\mathbf{k}}\right)\left(a - \frac{a^3\,k^2}{3}\right)\right\} \exp\left(\mathrm{i}\mathbf{k}\cdot\mathbf{r}\right)\end{aligned} \tag{B.33}$$

where $\hat{\mathbf{k}}$ is the unit vector. Inserting Eq. (B.33) in Eq. (B.31) gives,

$$\mathbf{M}^{(2)}(\mathbf{k}) = -\frac{3\,k^2}{4}\left\{\left(\mathbf{I} - \hat{\mathbf{k}}\,\hat{\mathbf{k}}\right)\left(a - \frac{a^3\,k^2}{3}\right)\right\} \int d\mathbf{r}\, \exp\left(\mathrm{i}\mathbf{k}\cdot\mathbf{r}\right) g \tag{B.34}$$

The integral in Eq. (B.34) can be evaluated as follows. It is known that for a spherical coordinate system $d\mathbf{r} = r^2\,\sin\theta\, dr\, d\theta\, d\phi$ and $\mathbf{k}\cdot\mathbf{r} = k\,r\,\cos\theta$ (Champeney, 1973), where θ is the angle between the vectors $\mathbf{k}$ and $\mathbf{r}$. We can write,

$$\begin{aligned}\int d\mathbf{r}\,\exp\left(\mathrm{i}\mathbf{k}\cdot\mathbf{r}\right) f &= \int_0^\infty dr \int_0^\pi d\theta \int_0^{2\pi} d\phi\;\exp\left(\mathrm{i}kr\,\cos\theta\right) r^2\,\sin\theta\;\; g \\ &= \int_0^\infty dr \int_0^\pi d\theta\;\, 2\,\pi\,r^2\,\sin\theta\;\; g\;\exp\left(\mathrm{i}kr\,\cos\theta\right)\end{aligned} \tag{B.35}$$

Considering $\cos\theta = x$,

$$\begin{aligned}\int d\mathbf{r}\,\exp\left(\mathrm{i}\mathbf{k}\cdot\mathbf{r}\right) f &= -\int_0^\infty dr \int_{+1}^{-1} 2\,\pi\,r^2\,dx\;\,g\;\;\left(\cos\left(krx\right)\,+\,\mathrm{i}\,\sin\left(krx\right)\right) \\ &= -\int_0^\infty dr\,\frac{2\,\pi\,r^2}{k\,r}\;g\;\;\left(\sin\left(-kr\right)\,-\,\mathrm{i}\,\cos\left(-kr\right)\,-\,\sin\left(kr\right)\,+\,\mathrm{i}\,\cos\left(kr\right)\right) \\ &= \int_0^\infty \frac{4\,\pi\,r^2}{k}\;\sin\left(kr\right)\mathrm{erf}(\alpha r)\,dr \\ &= -\frac{4\,\pi}{k}\,\frac{\partial^2}{\partial k^2}\left[\int_0^\infty \sin\left(kr\right)\mathrm{erf}(\alpha r)\,dr\right] \\ &= \frac{4\,\pi}{k}\,\frac{\partial^2}{\partial k^2}\left[\frac{\exp\left(\frac{-k^2}{4\,\alpha^2}\right)}{k}\right] \\ &= \frac{4\,\pi}{k}\,\frac{\partial}{\partial k}\left[\exp\left(\frac{-k^2}{4\,\alpha^2}\right)\left(\frac{1}{k^2}\,+\,\frac{1}{2\,\alpha^2}\right)\right] \\ &= -\frac{4\,\pi}{k}\left\{\exp\left(\frac{-k^2}{4\,\alpha^2}\right)\left(\frac{1}{2\,k\,\alpha^2}\,+\,\frac{k}{4\,\alpha^4}\,+\,\frac{2}{k^3}\right)\right\}\end{aligned} \tag{B.36}$$

Using this integral, Eq. (B.34) now becomes,

$$\begin{aligned}\mathbf{M}^{(2)}(\mathbf{k}) &= \left(\mathbf{I}\,-\,\hat{\mathbf{k}}\,\hat{\mathbf{k}}\right)\left(a\,-\,\frac{a^3\,k^2}{3}\right)\left(3\,\pi\,k\right)\;\exp\left(\frac{-k^2}{4\,\alpha^2}\right)\left(\frac{1}{2\,k\,\alpha^2}\,+\,\frac{k}{4\,\alpha^4}\,+\,\frac{2}{k^3}\right) \\ &= \left(\mathbf{I}\,-\,\hat{\mathbf{k}}\,\hat{\mathbf{k}}\right)\left(a\,-\,\frac{a^3\,k^2}{3}\right)\;\exp\left(\frac{-k^2}{4\,\alpha^2}\right)\left(\frac{3\,\pi}{2\,\alpha^2}\,+\,\frac{3\,\pi\,k^2}{4\,\alpha^4}\,+\,\frac{6\,\pi}{k^2}\right) \\ &= \left(\mathbf{I}\,-\,\hat{\mathbf{k}}\,\hat{\mathbf{k}}\right)\left(a\,-\,\frac{a^3\,k^2}{3}\right)\left(1\,+\,\frac{k^2}{4\,\alpha^2}\,+\,\frac{k^4}{8\,\alpha^4}\right)\left(\frac{6\,\pi}{k^2}\right)\;\exp\left(\frac{-k^2}{4\,\alpha^2}\right)\end{aligned} \tag{B.37}$$

This completes the derivation of $\mathsf{M}^{(2)}(\mathbf{k})$ required for the Fourier space sum. Recall that in the derivation of $\mathsf{M}^{(2)}(\mathbf{k})$ we have *included* the self interaction between a smooth function ($\mathsf{M}^{(2)}(\mathbf{r})$) and a point force $\mathbf{F}_\nu$ located at ν. In the following section, a compensating term $\mathsf{M}^{(2)}(\mathbf{r}=0)$ is derived which takes care of the self correction.

B.3.3 Calculation of $\mathsf{M}^{(2)}(\mathbf{r}=0)$ (self-correction term)

In order to proceed, first an inverse Fourier transform of $\mathsf{M}^{(2)}(\mathbf{k})$ is performed,

$$\mathsf{M}^{(2)}(\mathbf{r}) = \frac{1}{(2\,\pi)^3} \int\int\int \mathsf{M}^{(2)}(\mathbf{k})\ \exp\left(\mathrm{i}\mathbf{k}\cdot\mathbf{r}\right) d\mathbf{k} \tag{B.38}$$

Or,

$$\begin{aligned}
\mathsf{M}^{(2)}(\mathbf{r}=0) &= \frac{1}{(2\,\pi)^3} \int\int\int \mathsf{M}^{(2)}(\mathbf{k})\, d\mathbf{k} \\
&= \frac{1}{(2\,\pi)^3} \int_0^\infty \int_0^\pi \int_0^{2\pi} \mathsf{M}^{(2)}(\mathbf{k})\ k^2\ \sin\delta\ dk\ d\delta\ d\eta \\
&= \frac{2\,\pi}{(2\,\pi)^3} \int_0^\infty dk \int_0^\pi d\delta\ \sin\delta\ k^2\ \mathsf{M}^{(2)}(\mathbf{k}) \\
&= \frac{2}{(2\,\pi)^2} \int_0^\infty k^2\ \mathsf{M}^{(2)}(\mathbf{k})\ dk
\end{aligned} \tag{B.39}$$

Inserting $\mathsf{M}^{(2)}(\mathbf{k})$ from Eq. (B.37) into Eq. (B.39) leads to

$$\begin{aligned}
\mathsf{M}^{(2)}(\mathbf{r}=0) = \frac{3}{\pi} \int_0^\infty dk\ \mathsf{I} \left(a - \frac{a^3\,k^2}{3}\right) \left(1 + \frac{k^2}{4\,\alpha^2} + \frac{k^4}{8\,\alpha^4}\right) \exp\left(\frac{-k^2}{4\,\alpha^2}\right) \\
- \frac{3}{\pi} \int_0^\infty dk\ \hat{\mathbf{k}}\hat{\mathbf{k}} \left(a - \frac{a^3\,k^2}{3}\right) \left(1 + \frac{k^2}{4\,\alpha^2} + \frac{k^4}{8\,\alpha^4}\right) \exp\left(\frac{-k^2}{4\,\alpha^2}\right)
\end{aligned} \tag{B.40}$$

Note that the result of integrating a function of the dyadic product of a unit vector with itself, over all space, must be isotropic. Therefore, Eq. (B.40) can also be written as

$$\mathsf{M}^{(2)}(\mathbf{r}=0) = (I_1 - I_2)\,\mathbf{I} \tag{B.41}$$

where I_1 and I_2 are the scalar parts in the first and second terms on the right hand side of Eq. (B.40) respectively. Both I_1 and I_2 are evaluated as follows:

Evaluation of I_1:

$$\begin{aligned}
I_1 &= \frac{3}{\pi}\int_0^\infty dk\ \left(a - \frac{a^3\,k^2}{3}\right)\left(1 + \frac{k^2}{4\,\alpha^2} + \frac{k^4}{8\,\alpha^4}\right)\exp\left(\frac{-k^2}{4\,\alpha^2}\right) \\
&= \frac{3}{\pi}\left[\frac{-a}{12\,\alpha^2}\left\{4\,\alpha^3\,(20\,a^2\,\alpha^2 - 9)\sqrt{\pi}\ \mathrm{erf}\left(\frac{k}{2\,\alpha}\right)\right\}\right. \\
&\quad \left. + \frac{a}{12\,\alpha^2}\exp\left(\frac{-k^2}{4\,\alpha^2}\right)\left\{80\,k\,a^2\,\alpha^4 + 12\,k\,\alpha^2\,(a^2\,k^2 - 2) + k^3\,(a^2\,k^2 - 3)\right\}\right]_0^\infty \\
&= \frac{a\,\alpha}{\sqrt{\pi}}\left(9 - 20\,a^2\,\alpha^2\right)
\end{aligned} \tag{B.42}$$

Evaluation of I_2:

$$\begin{aligned}
\mathbf{I}\,I_2 &= \frac{3}{\pi}\int_0^\infty dk\ \hat{\mathbf{k}}\hat{\mathbf{k}}\left(a - \frac{a^3\,k^2}{3}\right)\left(1 + \frac{k^2}{4\,\alpha^2} + \frac{k^4}{8\,\alpha^4}\right) \\
&\quad \times \exp\left(\frac{-k^2}{4\alpha^2}\right)
\end{aligned} \tag{B.43}$$

Since $\hat{\mathbf{k}}\hat{\mathbf{k}} : \mathbf{I} = \hat{k_x}^2 + \hat{k_y}^2 + \hat{k_z}^2 = |\hat{\mathbf{k}}|^2 = 1$, a double dot product with $\mathbf{I}$ on both sides of Eq. (B.43) is carried out to give

$$\mathbf{I} : \mathbf{I}\,I_2 = \frac{3}{\pi}\int_0^\infty 1\ dk\ \left(a - \frac{a^3\,k^2}{3}\right)\left(1 + \frac{k^2}{4\,\alpha^2} + \frac{k^4}{8\,\alpha^4}\right)\exp\left(\frac{-k^2}{4\alpha^2}\right) \tag{B.44}$$

The same integration result that was evaluated for the case of I_1, can be used here. Also, noting that $\mathsf{I} : \mathsf{I} = 3$, we obtain,

$$I_2 \;=\; \frac{a\,\alpha}{3\sqrt{\pi}}\,(9 \,-\, 20\,a^2\,\alpha^2) \tag{B.45}$$

Using I_1 and I_2, Eq. (B.41) can be simplified to obtain $\mathsf{M}^{(2)}(\mathbf{r}=0)$

$$\begin{aligned}\mathsf{M}^{(2)}(\mathbf{r}=0) &= (I_1 - I_2)\,\mathsf{I} \\ &= \mathsf{I}\left\{\frac{a\,\alpha}{\sqrt{\pi}}\left(1 \,-\, \frac{1}{3}\right)(9 - 20a^2\alpha^2)\right\} \\ &= \mathsf{I}\,\frac{1}{\sqrt{\pi}}\left(6\,a\,\alpha \,-\, \frac{40}{3}\,a^3\,\alpha^3\right)\end{aligned} \tag{B.46}$$

Finally, using $\mathsf{M}^{(1)}(\mathbf{r})$, $\mathsf{M}^{(2)}(\mathbf{k})$ and $\mathsf{M}^{(2)}(\mathbf{r}=0)$, the Ewald summation formula for the branch $\mathcal{A}$ of RPY mobility tensor can be written (using Eq. (B.11)) as follows:

$$\begin{aligned}\mathsf{S}_\mu = \sum_{\nu=1}^{N} \mathsf{D}_{\mu\nu}\cdot\mathbf{F}_\nu = &\underbrace{\left(1 \,-\, \frac{6\,a\,\alpha}{\sqrt{\pi}} \,+\, \frac{40\,a^3\,\alpha^3}{3\sqrt{\pi}}\right)\mathbf{F}_\mu}_{\text{Point force and self correction}} \\ &+ \underbrace{\sum_{\mathbf{n}=0}^{\infty}{}' \sum_{\nu=1}^{N} \mathsf{M}^{(1)}(\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}})\cdot\mathbf{F}_\nu}_{\text{Real space sum}} \\ &+ \underbrace{\frac{1}{V}\sum_{\lambda}\sum_{\nu=1}^{N} \mathsf{M}^{(2)}(\mathbf{k}_\lambda)\cdot\mathbf{F}_\nu\,\cos\{\mathbf{k}_\lambda\cdot(\mathbf{r}_\nu - \mathbf{r}_\mu)\}}_{\text{Reciprocal space sum}}\end{aligned} \tag{B.47}$$

Recall that the first sum on the right-hand side of Eq. (B.47) converges rapidly. The $'$ over the sum indicates that the terms with self interactions are to be omitted. The second sum converges rapidly on the reciprocal lattice. The self-correction appears in the first term along with the point force. The summations in Eq. B.47 are carried out for all possible pairs of particles ($|\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}}| > 0$), however it should be used only for those pairs for which $|\mathbf{r}_{\nu\mathbf{n}} - \mathbf{r}_{\mu\mathbf{n_0}}| > 2a$. A modified Ewald summation formula which accounts for the overlapping beads is derived in Section (5.2.2). This modified formula involves an additional term which is evaluated only in the original simulation cell, and in particular only for overlapping beads. This additional term is discussed in the next section.

B.3.4 Derivation of the form of the additional term in the modified Ewald sum

In this section, an expression for the tensor M^* is derived, which makes up the correction term shown in Section (5.2.2). From Eq. (5.1), the sum $\sum_{\mu=1}^{N} \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_\mu$ for the two branches of the RPY tensor can be written as (where the overlapping particles have been explicitly identified),

$$\left[\sum_{\mu=1}^{N} \mathsf{D}_{\nu\mu} \cdot \mathbf{F}_\mu\right]_{\mathcal{A}} = \left[\left(1 - \frac{6a\alpha}{\sqrt{\pi}} + \frac{40a^3\,\alpha^3}{3\sqrt{\pi}}\right)\mathbf{F}_\nu\right]_{\mathcal{A}} + \left[\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{N^*} \mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}}) \cdot \mathbf{F}_\mu\right]_{\mathcal{A}}$$
$$+ \left[\sum_{\mathbf{n}\neq\mathbf{0}} \sum_{\mu=1}^{N^*} \mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}}) \cdot \mathbf{F}_\mu\right]_{\mathcal{A}} + \left[\sum_{\mathbf{n}} \sum_{\mu=N^*+1}^{N} \mathsf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}}) \cdot \mathbf{F}_\mu\right]_{\mathcal{A}}$$
$$+ \left[\sum_{\mathbf{k}\neq\mathbf{0}} \mathsf{M}^{(2)}(\mathbf{k}) \cdot \left\{\cos(\mathbf{k}\cdot\mathbf{r}_\nu) \sum_{\mu=1}^{N} \cos(\mathbf{k}\cdot\mathbf{r}_\mu)\mathbf{F}_\mu - \sin(\mathbf{k}\cdot\mathbf{r}_\nu) \sum_{\mu=1}^{N} \sin(\mathbf{k}\cdot\mathbf{r}_\mu)\mathbf{F}_\mu\right\}\right]_{\mathcal{A}} \tag{B.48}$$

and

$$\left[\sum_{\mu=1}^{N} \mathbf{D}_{\nu\mu}\cdot\mathbf{F}_\mu\right]_{\mathcal{B}} = \left[\left(1-\frac{6a\alpha}{\sqrt{\pi}}+\frac{40a^3\,\alpha^3}{3\sqrt{\pi}}\right)\mathbf{F}_\nu\right]_{\mathcal{B}} + \sum_{\substack{\mu=1\\ \mu\neq\nu}}^{N^*} \mathbf{M}_{\mathcal{B}}{}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\cdot\mathbf{F}_\mu$$
$$+\left[\sum_{\mathbf{n}\neq\mathbf{0}}\sum_{\mu=1}^{N^*}\mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})\cdot\mathbf{F}_\mu\right]_{\mathcal{B}} + \left[\sum_{\mathbf{n}}\sum_{\mu=N^*+1}^{N}\mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}})\cdot\mathbf{F}_\mu\right]_{\mathcal{B}}$$
$$+\left[\sum_{\mathbf{k}\neq\mathbf{0}}\mathbf{M}^{(2)}(\mathbf{k})\cdot\left\{\cos(\mathbf{k}\cdot\mathbf{r}_\nu)\sum_{\mu=1}^{N}\cos(\mathbf{k}\cdot\mathbf{r}_\mu)\mathbf{F}_\mu - \sin(\mathbf{k}\cdot\mathbf{r}_\nu)\sum_{\mu=1}^{N}\sin(\mathbf{k}\cdot\mathbf{r}_\mu)\mathbf{F}_\mu\right\}\right]_{\mathcal{B}} \tag{B.49}$$

Subtracting Eq. (B.48) from (B.49) and noting that the first, third, fourth and fifth terms in both the equations are the same, we can write,

$$\left[\sum_{\mu=1}^{N} \mathbf{D}_{\nu\mu}\cdot\mathbf{F}_\mu\right]_{\mathcal{B}} - \left[\sum_{\mu=1}^{N} \mathbf{D}_{\nu\mu}\cdot\mathbf{F}_\mu\right]_{\mathcal{A}}$$
$$= \sum_{\substack{\mu=1\\ \mu\neq\nu}}^{N^*} \mathbf{M}_{\mathcal{B}}{}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\cdot\mathbf{F}_\mu - \left[\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{N^*} \mathbf{M}^{(1)}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\cdot\mathbf{F}_\mu\right]_{\mathcal{A}} \tag{B.50}$$

The right-hand side of Eq. (B.50) is the quantity $\sum_{\substack{\mu=1\\ \mu\neq\nu}}^{N^*} \mathbf{M}^{\star}(\mathbf{r}_{\nu\mu,\mathbf{n}=\mathbf{0}})\cdot\mathbf{F}_\mu$ defined in Eq. (5.6). The difference between the sums on the left hand side of Eq. (B.50) can be calculated using explicit expressions of RPY functions given in Eqs. (4.3) - (4.6). As a result,

$$\mathbf{M}^{*}(\mathbf{x}) = \boldsymbol{\delta}\left[1-\frac{1}{2x^3}\left(\frac{3x^2}{4}+1\right)^2\right] + \hat{\mathbf{x}}\,\hat{\mathbf{x}}\left[\frac{1}{2x^3}\left(\frac{3x^2}{4}-1\right)^2\right] \tag{B.51}$$

where $\mathbf{x} = \mathbf{r}_{(\nu\mu,\mathbf{n}=\mathbf{0})}/a$ and $\hat{\mathbf{x}}$ is the unit vector in the direction of $\mathbf{r}_{(\nu\mu,\mathbf{n}=\mathbf{0})}$.

Appendix C

Scaling of Computational Cost with Chain Size

Asymptotic predictions in the long chain limit have been obtained for each state point $(z, c/c^{\star})$ by extrapolating finite chain data accumulated for chain lengths ranging from $N_b = 6$ to $N_b = 20$ as illustrated in Fig. 6.1. The use of this rather limited range of N_b values is necessitated by the computational cost of the Brownian dynamics simulation algorithm used here. As shown below, an estimate of the computational cost of the algorithm can be derived by using some simple scaling arguments, and the resulting expression can be verified by comparison with the CPU time required in the current simulations.

The computational cost of carrying out an Euler integration of the governing stochastic differential equation *for a single time step* has been shown in Chapter 5 to scale with system size as N^x, where $x = 2.1$. Recall that $N = N_b \times N_c$ is the total number of beads in a cubic cell of edge length L, with N_c being the number of bead-spring chains. Since typical simulations consist of runs extending over several relaxation times τ_1, followed by averaging over many independent runs, it is necessary to find the dependence of τ_1 on N_b in order to find the scaling of the total CPU cost. This is done as follows for simulations carried out in the semidilute regime C.

From Eq. (3.20), the requirement that $L \geq 2R_e$ (where R_e is the end-to-end distance of a chain) in order to prevent chains from wrapping over themselves leads to

$$L \sim b\left(\frac{c}{c^\star}\right)^{-\frac{1}{2}\frac{2\nu-1}{3\nu-1}} z^{2\nu-1} N_b^{1/2} \tag{C.1}$$

The concentration of monomers in the simulation box is $c = (N_b N_c)/L^3$. As a result, from Eq. (C.1)

$$c \sim \left(\frac{N_c}{b^3}\right)\left(\frac{c}{c^\star}\right)^{\frac{3}{2}\frac{2\nu-1}{3\nu-1}} z^{-3(2\nu-1)} N_b^{-1/2} \tag{C.2}$$

Using Eqs. (3.6) and (2.10), the overlap concentration $c^\star$ can be written in terms of the solvent quality z as

$$c^\star \sim b^{-3} z^{-3(2\nu-1)} N_b^{-1/2} \tag{C.3}$$

It follows from Eqs. (C.2) and (C.3) that N_c is independent of z, and related to the scaled concentration through the relation

$$N_c \sim \left(\frac{c}{c^\star}\right)^{\frac{1}{2}\frac{1}{3\nu-1}} \tag{C.4}$$

The number of chains in a simulation box is consequently *constant* when $(c/c^\star)$ is maintained constant.

The relaxation time of a macromolecule $\tau_1 \sim R^2/D$. From Eqs. (3.20) and (3.23), this implies

$$\tau_1 \sim \tau_0 \left(c/c^\star\right)^{\frac{2-3\nu}{3\nu-1}} z^{3(2\nu-1)} N_b^{3/2} \tag{C.5}$$

where $\tau_0 = \eta_s b^3/k_B T$ is the monomer relaxation time.

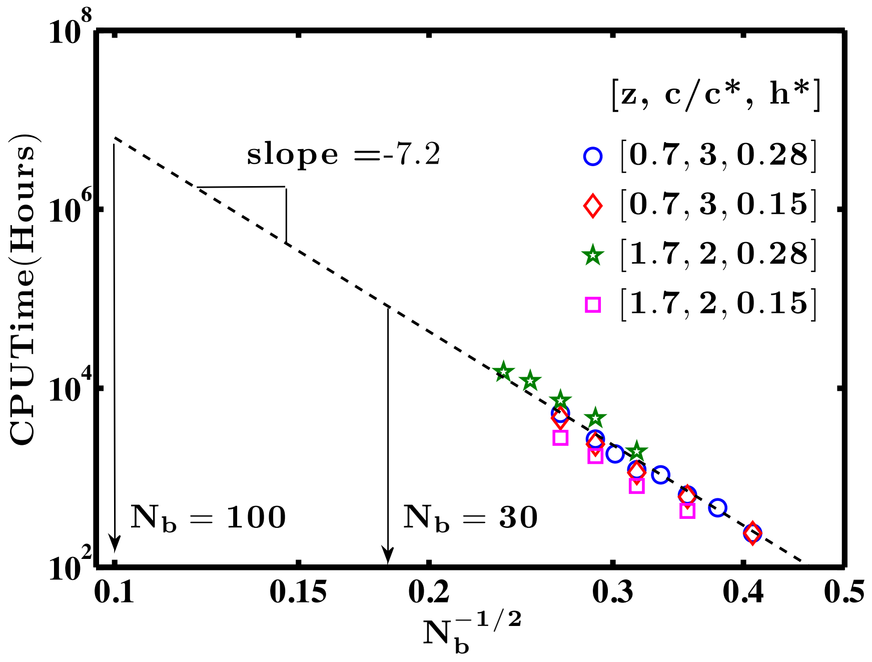


Figure C.1: CPU time as a function of chain length for Brownian dynamics simulations carried out at the two state points $(z, c/c^{\star}) = (0.7, 3)$ and $(1.7, 2)$ for two different values of the hydrodynamic interaction parameter ($h^{\star} = 0.15$ and 0.28), on a 156 SGI Altix XE 320 cluster. Symbols represent the various values of chain length N_b used in the simulations, while the dashed line is drawn with a slope predicted by the scaling relation Eq. (C.6).

The total computational cost of a single stochastic trajectory consequently scales as $\tau_1\,(N_c N_b)^x$. For fixed values of z and $(c/c^{\star})$ therefore

$$\text{Total CPU time per run} \sim N_b^{\frac{3}{2}+x} \sim \left(N_b^{-\frac{1}{2}}\right)^{-7.2} \tag{C.6}$$

where in the last expression on the right-hand side we have substituted the value $x = 2.1$ for the current algorithm, and used an exponent for N_b that enables a representation of the CPU cost as displayed in Fig. C.1.

The various symbols in Fig. C.1 represent simulations carried out at the two state points $(z, c/c^{\star}) = (0.7, 3)$ and $(1.7, 2)$, for two different values of the hydrodynamic interaction parameter $h^\star$, on a 156 SGI Altix XE 320 cluster. The simulations are identical to those used to display the ratio D/D_Z in Fig. 6.1. It is immediately apparent that the prediction in Eq. (C.6) for the scaling of the total CPU time with chain size is obeyed closely by the present simulations, independent of solvent quality and scaled concentration.

It is clearly desirable to add data for longer chains in order to improve the accuracy of the asymptotic value obtained by extrapolation to the long chain limit. Unfortunately, the additional CPU cost this would entail (which can be estimated from the scaling of computational cost with chain size shown in Fig. C.1) makes this highly infeasible. For instance, simulating a chain with $N_b = 30$ would require roughly $\mathcal{O}(10^4)$ CPU hours, while $N_b = 100$ would require roughly $\mathcal{O}(10^6)$ CPU hours! Indeed, by including simulations for a chain with $N_b = 30$, the CPU time for obtaining the static and dynamic properties of a semidilute solution at a single state point $(z, c/c^\star) = (3, 4)$ in the phase diagram is estimated to increase from its current value of approximately 7×10^4 hours to roughly 3×10^5 hours.

Appendix D

Error in the Extrapolated Value at $N_b \to \infty$

At each state point $(z, c/c^{\star})$ the mean value of D and the error in the mean, for a set of finite size chains with $N_b = 6, 8, ..., 20$ is obtained as described in Section 4.5.1 and Appendix A. Single chain BD simulations are used to compute D_z for the same values of N_b and z, and then the mean and error-of-mean of the ratio D/D_z is easily obtained . As described earlier, the asymptotic value of D/D_z in the long-chain limit is obtained by plotting the data at finite N_b as a function of $N_b^{-1/2}$, and extrapolating a straight line fit to the data, to the limit $N_b \to \infty$. The fitting of the straight line, and the estimation of the error in the extrapolated value is carried out here with the help of "Least-Squares Fitting" numerical routines provided in the GNU Scientific Library (GSL). Essentially, the GSL routine finds the least-squares fit by minimizing $\tilde{\chi}^2$, the weighted sum of squared residuals, for the straight line model $D/D_z = c_0 + c_1 N_b^{-1/2}$. The weights are the inverse of the error at each data point. The fitting routine gsl_fit_wlinear returns the best-fit parameters c_0 and c_1, along with a 2×2 covariance matrix that measures the statistical errors on c_0 and c_1 resulting from the errors in the data. The standard deviations of the best-fit parameters are then given by the square root of the corresponding diagonal elements of the covariance matrix. Particularly conveniently, the routine gsl_fit_linear_est uses the best-fit coefficients c_0, c_1 and their estimated covariance to compute the fitted function and its standard deviation at any desired point. By using gsl_fit_linear_est to find the value of the fitted function and its error at $N_b^{-1/2} = 0$, we determine both the infinite chain length limit value of D/D_z and its associated error. We find the error bars on the extrapolated values to be very large because the GSL routines account for the error on each data point and that results to a conservative estimate of the error bar on the extrapolated value. Though the error bars on the extrapolated results are large, a very good fit gives us the confidence that the actual error bar on the extrapolated result is much smaller.

Appendix E

Running LB/MD Simulations

In the LB/MD method, a polymer chain is modeled as a bead-spring chain consisting of N_b beads connected through $N_b - 1$ finitely extensible nonlinear elastic (FENE) massless springs. The short-ranged excluded volume interactions are accounted through a WCA potential given by,

$$U_{\text{WCA}} = 4\epsilon \left(\frac{\sigma^{12}}{r^{12}} - \frac{\sigma^6}{r^6} \right) + \epsilon, \qquad r \leq 2^{1/6}\sigma \tag{E.1}$$

and the FENE potential is given by,

$$U_{\text{FENE}} = -\frac{k_{\text{FENE}}\; q_0^2}{2} \ln \left(1 - \left(\frac{r}{q_0} \right)^2 \right) \tag{E.2}$$

where r is the distance between a pair of beads, and σ and ϵ are the length scale and the energy scale parameters of the WCA potential, respectively. In Eq. (E.2), k_{FENE} is the spring constant and q_0 is the maximum extension of a spring. In the LB/MD method, the natural unit system is based on the WCA potential parameters, i.e., the length unit is σ, the energy unit is ϵ and the time unit is considered to be τ, where $\tau = \sqrt{m\sigma^2/\epsilon}$ (where m is the mass of a monomer). Conversely, in the BD method with FENE springs, the length unit is $l_k = \sqrt{k_BT/k_{\text{FENE}}}$, the energy unit is k_BT and the time unit is $\tau_k = \zeta/(4k_{\text{FENE}})$ (where ζ is the bead friction coefficient). As the unit system is different in the LB/MD and BD methods, in order to compare these two methods, it becomes a necessary step to represent all the data in a common unit system. Pham et al. (2009) have carried out the parameter mapping between these two methods in detail. Here, their findings are briefly discussed.

Pham et al. (2009) make the point that the LB/MD method is based on an inertial time scale while the BD method is based on a diffusive time scale. This is relevant to the existence of a difference between the short-time and long-time friction coefficients. The short-time friction coefficient is denoted by ζ_{bare} which is an input parameter to the LB/MD method, where as the long-time friction coefficient ζ_{eff} is identified with the BD bead friction coefficient ζ. Dünweg and Ladd (2009) and Ahlrichs and Dünweg (1999) showed that ζ_{bare} and ζ_{eff} can be related through

$$\frac{1}{\zeta_{\text{eff}}} = \frac{1}{\zeta_{\text{bare}}} + \frac{1}{g\eta_s a'} \tag{E.3}$$

where, they showed that, $g \approx 25$ if a' is the LB lattice spacing. Pham et al. (2009) point out that it is intrinsically impossible to run the two simulations with the same unit system. As a result, they scaled all the parameters of LB/MD and BD based on the LB/MD unit system. Firstly, we set $l^{\dagger}\sigma = l^{\ddagger}l_k$ and $t^{\dagger}\tau = t^{\ddagger}\tau_k$, where the $\ddagger$ superscript denotes BD nondimensionalization, while $\dagger$ denotes a nondimensionalization for LB/MD. It follows that k_{FENE} can be written as $k^{\dagger}_{\text{FENE}}(\epsilon/\sigma^2)$, resulting in $l_k = \sqrt{\dfrac{\sigma^2 k_BT}{\epsilon k^{\dagger}_{\text{FENE}}}}$. Hence,

$$\frac{\sigma}{l_k} = \frac{l^{\ddagger}}{l^{\dagger}} = \sqrt{\frac{k^{\dagger}_{\text{FENE}}\epsilon}{k_BT}} \tag{E.4}$$

Similarly using $\zeta = \zeta_{\text{eff}}^{\dagger}\sqrt{\dfrac{m\epsilon}{\sigma^2}}$,

$$\frac{\tau}{\tau_k} = \frac{\tau^{\ddagger}}{\tau^{\dagger}} = \frac{4k_{\text{FENE}}^{\dagger}}{\zeta_{\text{eff}}^{\dagger}} \tag{E.5}$$

Note that k_BT/ϵ is the nondimensionalized temperature in LB/MD units, and is an input parameter to the LB/MD method. Another BD parameter that needs to be evaluated in terms of LB/MD parameters is the nondimensional bead radius $a^{\ddagger}$. This parameter is related to the conventionally defined (Thurston and Peterlin, 1967; Bird et al., 1987) hydrodynamic interaction parameter $h^{\star}$ by $a^{\ddagger} = \sqrt{\pi}h^{\star}$. We can find,

$$a^{\ddagger}l_k = \frac{\zeta_{\text{eff}}}{6\pi\eta_s} = \frac{\zeta_{\text{eff}}^{\dagger}\sigma}{6\pi\eta_s} \tag{E.6}$$

which implies

$$a^{\ddagger} = \frac{\zeta_{\text{eff}}^{\dagger}}{6\pi\eta^{\dagger}}\sqrt{\frac{k_{\text{FENE}}^{\dagger}\epsilon}{k_BT}} \tag{E.7}$$

In order to compare simulation results of these two methods, firstly, for given values of N_b, N_c and $c/c^{\star}$, the LB/MD simulation is run using the LB/MD input parameters $k_{\text{FENE}}^{\dagger}$, k_BT/ϵ, $\eta^{\dagger}$ and $\zeta_{\text{bare}}^{\dagger}$. Secondly, Eqs. (E.3), (E.4), (E.5) and (E.7) are used to calculate the values of corresponding BD parameters to run the BD simulations. The choice of LB/MD simulation parameters made in this study, as was in the case of Pham et al. (2009), are shown in the Table E.1. Using these parameters the LB/MD simulation is carried out. An example of a *Tcl script* (Example.tcl), and the command to execute the ESPResSo package are given at the end of this appendix. After running this script file, particles trajectories are stored in a text file ('Trajectory.dat"), which can be post-processed to obtain various physical properties. In order to run BD simulations, as mentioned

Nondimensional LB/MD simulation parameters	**Values**
k_BT/ϵ (Temperature)	1.2
$\rho^\dagger$ (Density)	0.864
$\nu^\dagger = \eta^\dagger/\rho^\dagger$ (Kinematic viscosity)	2.8
$a'^\dagger$ (Lattice spacing)	1
$\zeta^\dagger_{\text{bare}}$ (Short-time friction coefficient)	20.8
$k^\dagger_{\text{FENE}}$ (Spring constant)	7
$q^\dagger_0$ (Maximum extension of a spring)	2
$\Delta t^\dagger$ (Time step)	0.01

Table E.1: Choice of LB/MD simulation parameters

earlier, the BD parameters are first calculated. For example, if the box size in LB/MD units is L_{LB}, then in BD units the box size can be calculated using Eq. (E.4) to be $L_{\text{BD}} = L_{\text{LB}}\sqrt{\dfrac{k^\dagger_{FENE}}{k_BT/\epsilon}}$. Similarly, if the time step in LB/MD method is Δt_{LB}, then in BD the time step can be estimated using Eq. (E.5) to be $\Delta t_{\text{BD}} = \Delta t_{\text{LB}}(4k^\dagger_{\text{FENE}}/\zeta^\dagger_{\text{eff}})$. Once the results are obtained by running BD simulations, they need to be interpreted in terms of LB/MD units in order to make a proper comparison.

An example of tcl script to run ESPResSo

```
#########################################
Example.tcl
#########################################
set pi          3.1415926535
set verbose "yes"
set tcl_precision 12
#########################################
# System parameters
#########################################
# Following is a list of parameters used to define the system
# and also to define interaction parameters
set n_beads      <insert value here>  # Number of beads in a chain
set temp         1.2
set time_step    0.01 # Time step used in the main integrator
set skin         0.4
set n_chains     <insert value here> # Number of chains
set bond_l       0.95
set min_dist     0.9
set cap_start    5
set n_total     [expr $n_beads*$n_chains]
set fene_k       7.0 #FENE parameter
set box_l       <insert value here> #Box size L
set ccs         <insert value here> #c/c*
set fene_r       2.0 #FENE parameter
set lj_eps       1.0 #WCA parameter
set lj_sig       1.0 #WCA parameter
set lj_cut       1.122462048 #WCA parameter
set lj_shift     0.25 #WCA parameter
set lj_off       0.0 #WCA parameter
#########################################
# Lattice Boltzmann parameters
#########################################
set lb_dens      0.864 # Density
set lb_visc      2.8 # Kinematic viscosity
set lb_grid      1.0 # Lattice spacing
set lb_tau       0.02 # time step used equilibration process
set lb_zeta      20.8 # bare friction coefficient
set zeta_eff     [expr 1/(1/$lb_zeta+1/(25.0*$lb_dens
                        *$lb_visc*$lb_grid))] # This one is similar to
                                                BD friction coefficient
#########################################
# Integration parameters
#########################################
# All warm_* variables are used in Langevin warm up step
set warm_steps   <insert value here> # Number of time steps
set warm_loops   <insert value here>  # Number of loops
# All diff_* variables are used in LB warp up step
set diff_time   <insert value here> #Physical time for the diffusion process
set diff_steps  [expr int($diff_time/$time_step)] # Number of time steps
# All n_* variables are used in the main integration step
set n_steps      <insert value here> # Number of time steps
set n_loops      <insert value here>  # Number of loops
set data_collection_interval    <insert value here> # Number of data points
```

```
                        after which the data will be entered in to output file
#########################################
# System identification
#########################################
set filename [format "Trajectory.dat" ] # Setting the name of output file
##########################################
# Procedures
##########################################
proc output { args } {
    global verbose
    if { $verbose == "yes" } {
        set l [llength $args]
        if { $l == 1 } {
            puts [lindex $args 0]
        } elseif { $l == 2 } {
            puts [lindex $args 0] [lindex $args 1]
        } else {
            puts [lindex $args 0] [lindex $args 1] [eval concat
                    [lrange $args 2 $l]]
        }
    }
}
proc write_positions { dest } {
    global n_total
    for { set np 0 } { $np < $n_total} { incr np } {
        set pos [part $np print pos]
        lappend xpos [lindex $pos 0]
        lappend ypos [lindex $pos 1]
        lappend zpos [lindex $pos 2]
    }
    puts $dest "[setmd time] $xpos"
    puts $dest "[setmd time] $ypos"
    puts $dest "[setmd time] $zpos"
}
#########################################
# Setup the system
#########################################
# Random number generator
#########################################
t_random seed 54919
#########################################
```

```
# Setting the simulation box
#########################################
setmd box_l $box_l $box_l $box_l
setmd periodic 1 1 1
cellsystem domain_decomposition -no_verlet_list
#########################################
# Setting Interactions (manual of Espresso can be see for the details)
#########################################
inter 0 FENE $fene_k $fene_r #FENE interaction
inter 0 0 lennard-jones $lj_eps $lj_sig $lj_cut $lj_shift $lj_off
                #LJ or WCA interaction
#########################################
# Setting Polymer Chains (achieving SAW)
#########################################
polymer $n_chains $n_beads  $bond_l type 0 0
                                bond 0 mode SAW 0.5 1000000
#########################################
# Integration
#########################################
setmd time_step $time_step
setmd skin $skin
#########################################
# Run the simulation
#########################################
# warmup with Langevin equation (part of equilibration)
#########################################
thermostat langevin $temp 0.5
set cap $cap_start
set act_min_dist [analyze mindist]
inter ljforcecap $cap
set i 1
while { $i <= $warm_loops || $act_min_dist < $min_dist } {
    integrate $warm_steps
    set act_min_dist [analyze mindist]
    set cap [expr $cap+10]
    inter ljforcecap $cap
    incr i
}
output ""
inter ljforcecap 0
galileiTransformParticles
```

```
invalidate_system
########################################
# warmup with Lattice Boltzmann (Diffusion process and part of equilibration)
########################################
thermostat off
lbfluid density $lb_dens viscosity $lb_visc agrid $lb_grid tau $lb_tau
lbfluid friction $lb_zeta
thermostat lb $temp
for { set i 1 } { $i <= $warm_loops } { incr i } {
    output -nonewline "\[Warmup $i/$warm_loops\]\r"; flush stdout
    integrate $diff_steps
}
output ""

# Langevin and Lattice Boltzmann warm ups processes equilibrate the system
########################################
# Integration (This is he main integration step which is used to store
# particle trajectories and also to compare CPU time)
########################################
setmd time 0.0
set file [open "$filename" "w"]
write_positions $file
for { set i 1 } { $i <= $n_loops } { incr i } {
    output -nonewline "\[Loop $i/$n_loops\]\r"; flush stdout
    for { set j 1 } { $j <= $n_steps/$data_collection_interval }
                        { incr j } {
        #integrate $n_steps
        integrate $data_collection_interval
        write_positions $file
    }
}
#output ""
close $file
########################################
```

In order to execute this script file, ESPResSo must be installed and the following command is used:

```
>> Espresso Example.tcl
```

Appendix F

Static Structure Factor Calculations

As discussed in Chapter 3, in semidilute polymer solutions both hydrodynamic and excluded volume interactions are screened on a length scale of ξ_c, which denotes the size of a concentration blob. The information about the screening length or the concentration driven crossover in semidilute polymer solutions is reflected in the polymer structure factor (Huang et al., 2010). The aim of this section is to compute the structure factor for a range of $c/c^{\star}$ and z, and we then hope to interpret the size of a concentration blob in terms of structure factor results. The static structure factor of a polymer chain is given by

$$S(k) = \frac{1}{N_b} \sum_{\mu\nu} \langle \exp\left(\mathrm{i}\mathbf{k} \cdot \mathbf{r}_{\mu\nu}\right) \rangle = \frac{1}{N_b} \sum_{\mu\nu} \left\langle \frac{\sin\left(k r_{\mu\nu}\right)}{k r_{\mu\nu}} \right\rangle \tag{F.1}$$

where $\mathbf{k}$ is the scattering vector and k is the magnitude of $\mathbf{k}$, also known as wavenumber. The distance vector $\mathbf{r}_{\mu\nu}$ connecting the beads μ and ν is an abbreviation for $\mathbf{r}_{\nu} - \mathbf{r}_{\mu}$ and $r_{\mu\nu}$ is the magnitude of $\mathbf{r}_{\mu\nu}$.

Scaling arguments can be used to show that the structure factor $S(k)$ of a chain in a dilute solution scales as $k^{-1/\nu}$ (Doi and Edwards, 1986) in the range $2\pi/R_g < k < 2\pi/b$ for polymers with $R_g \sim N_b^{\nu}$, where R_g is the gyration radius of a chain and b is the size of a monomer. Values of ν are 0.5 and 0.588 for ideal chain and good solvent chain, respectively. As discussed in Chapter 3, the length scale ξ_c lies between R_g and b, and two regimes are separated by ξ_c: (i) good solvent regime when the length scale is in the range b to ξ_c, and (ii) θ solvent regime when the length scale is between ξ_c and R_g. This leads to expectation of two different scaling laws for $S(k)$ in semidilute solutions: (i) $S(k) \sim k^{-1/\nu}$ in the good solvent regime where $2\pi/\xi_c < k < 2\pi/b$, and (ii) $S(k) \sim k^{-2}$ in the θ solvent regime where $2\pi/R_g < k < 2\pi/\xi_c$. As discussed by Yamakov et al. (1997) and Huang et al. (2010), these two scaling laws can be clearly distinguished if very long polymer chains are considered.

Indeed, Huang et al. (2010), have shown, by considering long polymer chains ($N_b = 50, 250$), that there exist two scaling laws for $S(k)$ in the semidilute regime by carrying out hybrid Multi-particle Collision Dynamics/Molecular Dynamics (MPCD) simulations. They plotted $S(k)$ as a function of k on a log - log scale, and showed that for lower values of k, $S(k)$ decays as k^{-2} while for higher wavenumbers, $S(k)$ scales as $k^{-1/\nu}$. As a result, they were able to predict the size of a concentration blob $\xi_c = 2\pi/k_c$, where k_c is the wavenumber at which k^{-2} slope changes to $k^{-1/\nu}$ slope.

Due to the computational limitations of our algorithm, it is not feasible for us to simulate very long chains. We thought it worthwhile to check if the existence of two scaling laws for $S(k)$ in semidilute solutions, and the values of ξ_c for a range of $c/c^\star$ and z could be predicted, with the help of the extrapolation procedure mentioned in Section 6.2.

F.1 Simulation results

As discussed in earlier sections, the behavior of semidilute polymer solutions is captured in the asymptotic limit $N_b \to \infty$ and only in this limit can the universal crossover scaling functions be verified. The same set of input simulation parameters are considered here (see Table 6.1) that were used to obtain crossover scaling functions ϕ_R, ϕ_D and ϕ_η. As a matter of fact, simulations are run only once for a set of parameters and then all the properties (gyration radius, diffusivity, structure factors) are calculated based on the beads' trajectories in the post-processing stage.

A similar extrapolation technique, as was used for other properties, is used here. Extrapolation of the finite chain data to the limit $N_b \to \infty$ is performed in order to obtain the asymptotic result for static structure factor $S(k)$. Though the procedure to obtain extrapolated data is explained in Section 6.2, because of a certain subtlety, it is important to discuss the procedure in the context of structure factor calculations. This subtlety arises because rather than plotting $S(k)$ vs. k, we plot $S(k)/N_b$ vs. kR_g since it provides a wider window in the range of k in which we are interested. In order to obtain the extrapolated data, it is necessary to fix the value of kR_g and calculate $S(k)/N_b$ for each N_b. Note that kR_g can have different value for different N_b at a fixed value of k. Therefore, a linear interpolation is used to obtain $S(k)/N_b$ around a fixed value of kR_g. Once $S(k)/N_b$ data is calculated for all N_b at a fixed value of kR_g, $S(k)/N_b$ is plotted as a function of N_b^{-2} as shown in Fig. F.1. A straight line is fitted through the data and finally the fitted line is used to evaluate $S(k)/N_b$ at $N_b^{-2} = 0$ or $N_b = \infty$. It is found, for this particular case, that the best extrapolation result is obtained when a straight line is chosen, and when the x-axis is selected to be N_b^{-2}. The selection of whether to choose a straight line or to choose a polynomial of a certain order can be different at different values of kR_g, based on which fits

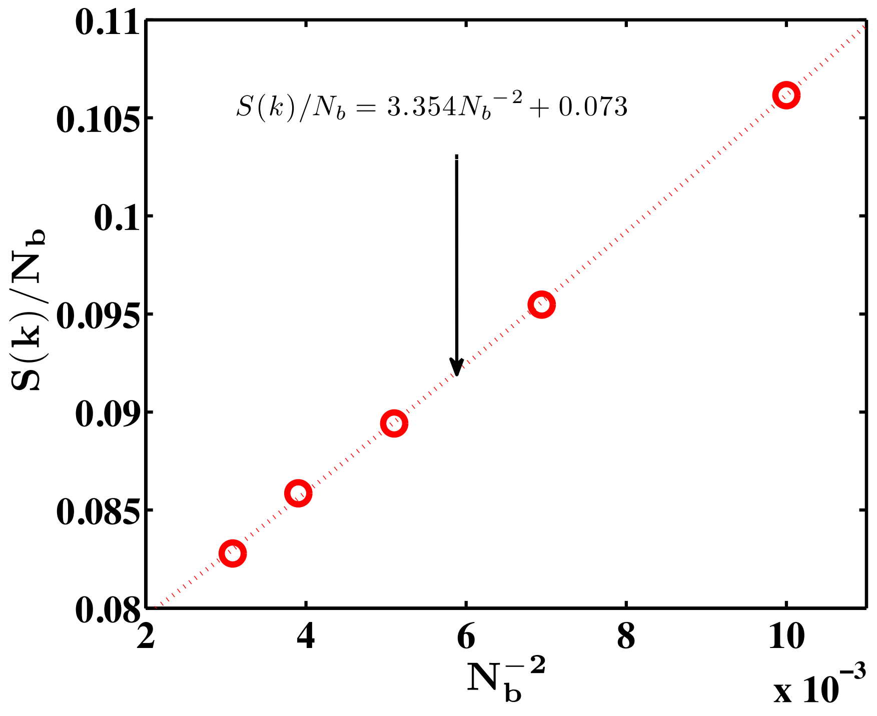


Figure F.1: A straight line is fitted to obtain $S(k)/N_b$ in the long-chain limit by extrapolating finite N_b results at a fix value of $kR_g = 5.025$. Following parameters are used: $z = 1.7$, $c/c^{\star} = 1$.

best with the simulation data. This extrapolation procedure is repeated for a range of kR_g values (ranging from 0.4 to 16), and the results are shown in Fig. F.2 for $z = 1.7$ and $c/c^{\star} = 1$, where we find that $S(k)/N_b$ scales as a power law in kR_g. As discussed earlier, for θ-solutions, the exponent in the power law leads to $\nu = 0.5$ and for solutions under good solvent conditions it leads to $\nu = 0.588$. However, for $z = 1.7$, which lies in a crossover regime between θ and good solvent condition, the concept of effective exponent ν_{eff} is more appropriate. The slope of the extrapolated data is -1.72, which corresponds to $\nu_{\text{eff}} = 0.55$. It is interesting to see that this slope is not achieved for finite size chains but achieved only after extrapolating the finite chain data to $N_b \to \infty$ limit.

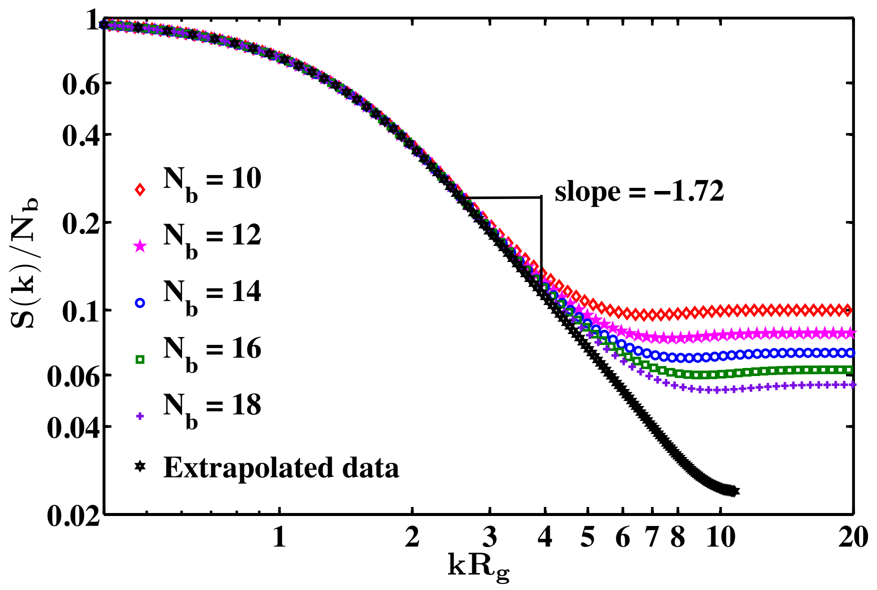


Figure F.2: Illustration of extrapolation for $S(k)/N_b$ at $z = 1.7$ and $c/c^{\star} = 1$

It is clear from Fig. F.2 that there is only a single slope of -1.72, and hence, a single scaling law for the static structure factor. This means that, for this set of $c/c^{\star}$ and z, the extrapolation procedure fails to capture information about the concentration blob.

We have checked for a few more values of $c/c^{\star}$ to see if the extrapolation procedure is successful in predicting the blob size. Figure F.3 shows the structure factor for $c/c^{\star} = 1$ and 4, simulated at a fixed value of $z = 1.7$. For $c/c^{\star} = 1$, as pointed out earlier, the effective exponent $\nu_{\text{eff}} = 0.55$. However, by increasing the concentration to $c/c^{\star} = 4$, the values of ν_{eff} reduces to 0.52, which is close to an exponent value for θ-solutions. This can be explained using Flory theory, according to which the excluded volume interaction begin to be screened as the concentration is increased.

A similar pattern was observed when z was varied at a fixed value of $c/c^{\star}$. The values of ν_{eff} tend towards the expected asymptotic values in θ and good

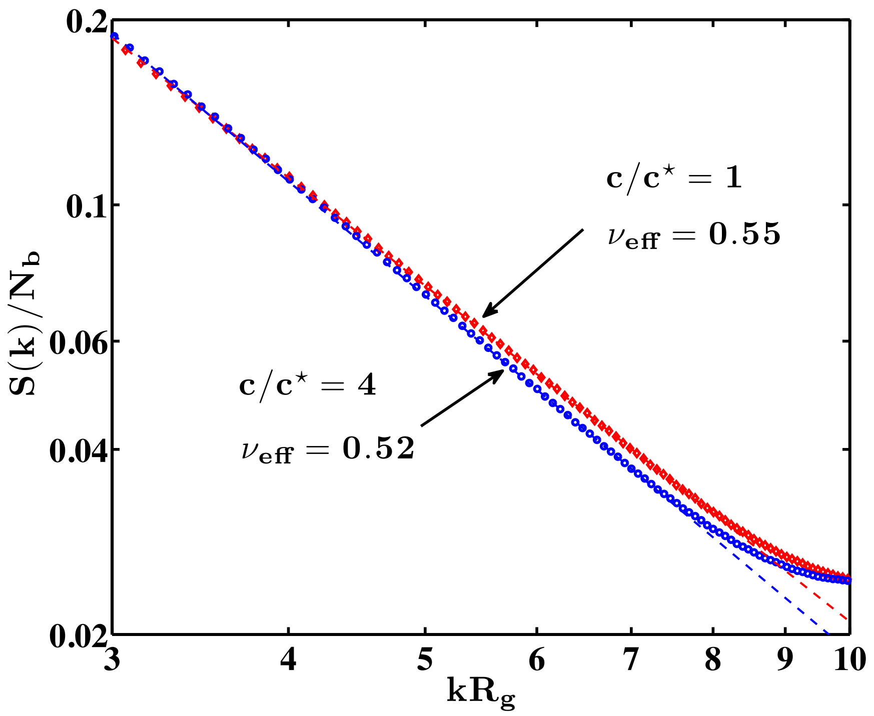


Figure F.3: Effect of $c/c^\star$ on structure factor scalings at a fixed value of $z = 1.7$.

solvents. However, the use of an extrapolation procedure is insufficient to tease out the subtle dependences of the blob size on z and $(c/c^\star)$, and simulations of much longer chains are required for this purpose.

Appendix G

Periodic Boundary Condition for Planar Elongational Flow

G.1 Introduction

In this appendix, periodic boundary conditions (PBCs) are discussed for planar elongational flows (PEF). The derivation of PBCs for PEF was first carried out by Kraynik and Reinelt (1992). Here the intention is to elaborate many steps in their derivation in detail. The key questions to be addressed in this appendix are the following: (i) At what angle should the lattice be inclined initially? (ii) What should the initial lattice vectors be? (iii) At what time should the lattice be mapped back to its original configuration? The very first step in order to address these questions is to recognize the condition for the reproducibility of a lattice. A lattice is said to be reproducible with another lattice if the lattice points on both the lattices coincide. The equations for the lattice reproducibility condition leads to an eigenvalue problem, and the solution of the eigenvalue problem is used to answer these three questions.

G.2 Condition for Reproducibility

We start by constructing an arbitrary 3-D lattice consisting of all the points $\mathbf{R_n}$ through the expression,

$$\mathbf{R_n} = n_1\mathbf{b_1} + n_2\mathbf{b_2} + n_3\mathbf{b_3} \tag{G.1}$$

where $\mathbf{b_1}$, $\mathbf{b_2}$ and $\mathbf{b_3}$ are linearly independent basis vectors, and $\mathbf{n} = \{n_1, n_2, n_3\}$ is a set of integer numbers. In order to be compatible with the elongational flow, these basis vectors need to evolve suitably with respect to time. The evolution of these basis lattice vectors can be written in terms of the following differential equation,

$$\frac{d\mathbf{b_i}}{dt} = \mathbf{b_i} \cdot \nabla\mathbf{u} = \mathbf{b_i} \cdot \mathbf{D} \tag{G.2}$$

where $\nabla\mathbf{u}$ or $\mathbf{D}$ is the flow gradient tensor. For planar elongational flows, it is given by

$$\mathbf{D} = \begin{pmatrix} \dot\epsilon & 0 & 0 \\ 0 & \dot\epsilon & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

If $\mathbf{b_1^0}$, $\mathbf{b_2^0}$ and $\mathbf{b_3^0}$ are considered to be the initial basis vectors, then Eq. (G.2) can be integrated to give,

$$\mathbf{b_i} = \mathbf{b_i^0} \cdot \mathbf{\Lambda} \tag{G.3}$$

where $\mathbf{\Lambda} = \exp(\mathbf{D}t)$. As $\mathbf{D}$ is a diagonal tensor, $\mathbf{\Lambda}$ can be written as,

$$\mathbf{\Lambda} = \begin{pmatrix} e^{\dot\epsilon t} & 0 & 0 \\ 0 & e^{-\dot\epsilon t} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{G.4}$$

$\mathbf{\Lambda}$ is also called the time evolution matrix.

As pointed out by Kraynik and Reinelt (1992), for a given $\mathbf{\Lambda}$, a lattice is

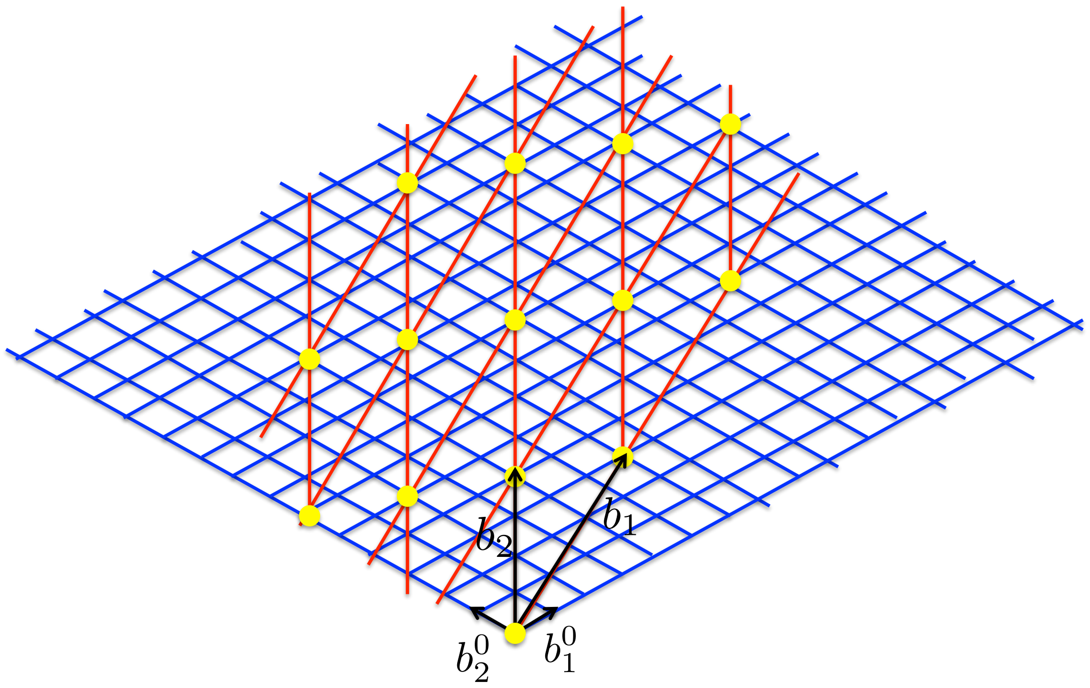


Figure G.1: Schematic illustration of lattice reproducibility condition for a 2-D lattice with $\{N_{11}, N_{12}, N_{21}, N_{22}\} = \{6, 3, 4, 4\}$.

reproducible if and only if there exist integers N_{ij} such that,

$$\mathbf{b_i} = \mathbf{b_i^0} \cdot \mathbf{\Lambda} = N_{i1}\mathbf{b_1^0} + N_{i2}\mathbf{b_2^0} + N_{i3}\mathbf{b_3^0} \tag{G.5}$$

This can be understood easily by considering an example of the 2-D lattice shown in Fig. G.1. The blue lattice in Fig. G.1 is generated using the initial basis vectors $\mathbf{b_1^0}$ and $\mathbf{b_2^0}$. If we consider, for example, $\{N_{11}, N_{12}, N_{21}, N_{22}\} = \{6, 3, 4, 4\}$, then $\mathbf{b_1}$ and $\mathbf{b_2}$ can be calculated from Eq. (G.5). The red lattice in Fig. G.1 is generated using $\mathbf{b_1}$ and $\mathbf{b_2}$. It is clear that the red lattice points overlap with the blue lattice points, as indicated by yellow points in the figure. This illustrates the reproducibility of a lattice. Figure G.1 is just a schematic representation of these lattices, and a red cell is really a deformed version of a blue cell.

If we need to remap the lattice to its original configuration after a certain time, we must make sure that the lattice remains periodic after the remapping. Todd and Daivis (1998) and Baranyai and Cummings (1999) have shown that the reproducibility of the lattice can be successfully utilized to ensure that periodic boundary conditions are maintained throughout the simulation. We can obtain a reproducible lattice only for a certain set of integers that satisfy Eq. (G.5), and therefore, our problem now reduces to solving for such integers.

G.3 The Eigenvalue Problem

The reproducibility condition (Eq. (G.5)) can be turned into an eigenvalue problem, as discussed below. Equation (G.5) is first expanded in the form,

$$\mathbf{b_1^0} \cdot \boldsymbol{\Lambda} = N_{11}\mathbf{b_1^0} + N_{12}\mathbf{b_2^0} + N_{13}\mathbf{b_3^0} \tag{G.6}$$

$$\mathbf{b_2^0} \cdot \boldsymbol{\Lambda} = N_{21}\mathbf{b_1^0} + N_{22}\mathbf{b_2^0} + N_{23}\mathbf{b_3^0} \tag{G.7}$$

$$\mathbf{b_3^0} \cdot \boldsymbol{\Lambda} = N_{31}\mathbf{b_1^0} + N_{32}\mathbf{b_2^0} + N_{33}\mathbf{b_3^0} \tag{G.8}$$

Since, $\boldsymbol{\Lambda}$ is a diagonal matrix or $\Lambda_{ij} = 0$ for $i \neq j$, Eqs. (G.6), (G.7) and (G.8) can be written in component form as,

$$\Lambda_{11}b_{1,x}^0 = N_{11}b_{1,x}^0 + N_{12}b_{2,x}^0 + N_{13}b_{3,x}^0 \tag{G.9}$$

$$\Lambda_{22}b_{1,y}^0 = N_{11}b_{1,y}^0 + N_{12}b_{2,y}^0 + N_{13}b_{3,y}^0 \tag{G.10}$$

$$\Lambda_{33}b_{1,z}^0 = N_{11}b_{1,z}^0 + N_{12}b_{2,z}^0 + N_{13}b_{3,z}^0, \tag{G.11}$$

$$\Lambda_{11}b_{2,x}^0 = N_{21}b_{1,x}^0 + N_{22}b_{2,x}^0 + N_{23}b_{3,x}^0 \tag{G.12}$$

$$\Lambda_{22}b_{2,y}^0 = N_{21}b_{1,y}^0 + N_{22}b_{2,y}^0 + N_{23}b_{3,y}^0 \tag{G.13}$$

$$\Lambda_{33}b_{2,z}^0 = N_{21}b_{1,z}^0 + N_{22}b_{2,z}^0 + N_{23}b_{3,z}^0 \tag{G.14}$$

and

$$\Lambda_{11} b^0_{3,x} = N_{31} b^0_{1,x} + N_{32} b^0_{2,x} + N_{33} b^0_{3,x} \tag{G.15}$$

$$\Lambda_{22} b^0_{3,y} = N_{31} b^0_{1,y} + N_{32} b^0_{2,y} + N_{33} b^0_{3,y} \tag{G.16}$$

$$\Lambda_{33} b^0_{3,z} = N_{31} b^0_{1,z} + N_{32} b^0_{2,z} + N_{33} b^0_{3,z} \tag{G.17}$$

Considering all the x component equations in Eqs. (G.9), (G.12) and (G.15), we can write the matrix equation,

$$\Lambda_{11} \begin{pmatrix} b^0_{1,x} \\ b^0_{2,x} \\ b^0_{3,x} \end{pmatrix} = \begin{pmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & N_{33} \end{pmatrix} \cdot \begin{pmatrix} b^0_{1,x} \\ b^0_{2,x} \\ b^0_{3,x} \end{pmatrix} \tag{G.18}$$

Similarly, all the y components can be combined in a matrix equation,

$$\Lambda_{22} \begin{pmatrix} b^0_{1,y} \\ b^0_{2,y} \\ b^0_{3,y} \end{pmatrix} = \begin{pmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & N_{33} \end{pmatrix} \cdot \begin{pmatrix} b^0_{1,y} \\ b^0_{2,y} \\ b^0_{3,y} \end{pmatrix} \tag{G.19}$$

and the z components equations can be represented by

$$\Lambda_{33} \begin{pmatrix} b^0_{1,z} \\ b^0_{2,z} \\ b^0_{3,z} \end{pmatrix} = \begin{pmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & N_{33} \end{pmatrix} \cdot \begin{pmatrix} b^0_{1,z} \\ b^0_{2,z} \\ b^0_{3,z} \end{pmatrix} \tag{G.20}$$

Eqs. (G.18), (G.19) and (G.20) can be written in the general form

$$\lambda_j \begin{pmatrix} b^0_{1,j} \\ b^0_{2,j} \\ b^0_{3,j} \end{pmatrix} = \begin{pmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & N_{33} \end{pmatrix} \cdot \begin{pmatrix} b^0_{1,j} \\ b^0_{2,j} \\ b^0_{3,j} \end{pmatrix} \tag{G.21}$$

where $\lambda_j = \Lambda_{jj} = \exp(D_{jj}t)$, and $j = 1,2,3$ or x,y,z. If we consider $\mathbf{c_j}$ to be a column vector of components $\{b^0_{1,j}, b^0_{2,j}, b^0_{3,j}\}$, then Eq. (G.21) can be written as,

$$\lambda_j \mathbf{I} \cdot \mathbf{c_j} = \mathbf{N} \cdot \mathbf{c_j} \tag{G.22}$$

which can be expressed as the eigenvalue problem,

$$(\mathbf{N} - \lambda_j \mathbf{I}) \cdot \mathbf{c_j} = \mathbf{0} \tag{G.23}$$

The solution of this eigenvalue problem leads to results for the strain period, the magic angle and the initial lattice vectors.

G.4 The Strain Period of the Lattice

The strain period is obtained by first finding the eigenvalues of the $\mathbf{N}$ tensor. In order to obtain the eigenvalues λ_j, Eq. (G.23) is solved to give the following characteristic equation

$$\lambda_i^3 - k\lambda_i^2 + m\lambda_i - l = 0 \tag{G.24}$$

where,

$k = N_{11} + N_{22} + N_{33} = \mathrm{tr}(\mathbf{N})$,

$m = N_{11}N_{22} + N_{11}N_{33} + N_{22}N_{33} - N_{32}N_{23} - N_{12}N_{21} - N_{13}N_{31}$
$= \frac{1}{2}\left[\{\mathrm{tr}(\mathbf{N})\}^2 - \mathrm{tr}(\mathbf{NN})\right]$,

and

$l = N_{11}N_{22}N_{33} - N_{11}N_{32}N_{23} - N_{12}N_{21}N_{33} + N_{12}N_{31}N_{23} + N_{13}N_{21}N_{32} - N_{13}N_{31}N_{22}$
$= \det(\mathbf{N})$.

We can evaluate k, l and m by making use of the fact that they are the invariants of the matrix $\mathbf{N}$, and λ_i $(i = 1,2,3)$ are the eigenvalues of $\mathbf{N}$.

1.
$$k = \mathrm{tr}(\mathbf{N}) = \lambda_1 + \lambda_2 + \lambda_3 \qquad \text{(G.25)}$$

2.
$$m = \frac{1}{2}\left[\{\mathrm{tr}(\mathbf{N})\}^2 - \mathrm{tr}(\mathbf{NN})\right] = \lambda_2\lambda_3 + \lambda_1\lambda_3 + \lambda_1\lambda_2 \qquad \text{(G.26)}$$

Dividing the above equation by $\lambda_1\,\lambda_2\,\lambda_3$ $(= 1)$, leads to

$$m = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \frac{1}{\lambda_3} \qquad \text{(G.27)}$$

3.
$$l = \det(\mathbf{N}) = \lambda_1\,\lambda_2\,\lambda_3 \qquad \text{(G.28)}$$

For isochoric deformations, $\det(\mathbf{N}) = 1$, hence, $l = 1$.

If $\lambda_1 = \phi$, then for planar elongational flow, it follows from Eq. (G.4) that $\lambda_2 = 1/\phi$ and $\lambda_3 = 1$. When these eigenvalues are inserted in Eq. (G.25) we get,

$$\phi + \frac{1}{\phi} + 1 = k \qquad \text{(G.29)}$$

which is a quadratic equation for ϕ, with a solution

$$\phi = \frac{(k-1) + \sqrt{(k-1)^2 - 4}}{2} = \lambda_1 \qquad \text{(G.30)}$$

which is one of the roots of the quadratic equation. It can be easily shown that $\dfrac{1}{\phi} = \dfrac{(k-1) - \sqrt{(k-1)^2 - 4}}{2}$. As a result, the second root is simply $1/\phi$ or λ_2. Note that from Eq. (G.27), and the known values of λ_1, λ_2 and λ_3, it follows that $m = k$.

For a discrete set of k, a discrete set of ϕ can be calculated. From Eq. (G.30) it is clear that k cannot be less than 3. Since $m = k$, it follows that $m \geq 3$. In the case of no deformation, the matrix $\mathbf{\Lambda}$ is a unit matrix, and for this trivial case $k = m = 3$. We are interested in nontrivial cases in which the lattices are deforming, i.e., we require integers $k > 3$ and $m > 3$ for which there exist lattices that are reproducible and periodic. For $k = 4, 5, 6, \ldots$, discrete values of ϕ can be calculated, and at only these discrete values the lattice will exhibit periodic planar elongational flow. Since $\phi = e^{\dot{\epsilon}t}$, for a given $\dot{\epsilon}$, the times at which the lattice will be periodic are $t = \dfrac{\log\phi}{\dot{\epsilon}}$.

Finding the information for strain period is useful but not complete. It is also required to derive the initial lattice configuration with which the simulation starts, for instance, the initial lattice vectors and the orientation of the lattice (the magic angle). These can be achieved by finding out a suitable $\mathbf{N}$ matrix.

G.5 Magic Angle and Initial Lattice

In planar flows, the lattice in the direction perpendicular to the plane of flow remains undeformed. As a result $\mathbf{b_3} = \mathbf{b_3^0}$ at all times, and it is sufficient to consider the 2-D lattice in the plane of flow. The 2-D lattice can be defined in terms of (i) a, the ratio of the magnitudes of $\mathbf{b_2}$ to $\mathbf{b_1}$, and (ii) Φ, the angle between $\mathbf{b_1}$ and $\mathbf{b_2}$, as shown in Fig. G.2. If θ is the angle at which the lattice is initially oriented, then we can write the basis vectors as

$$\mathbf{b_1} = (\cos\theta, \sin\theta) \quad \text{and} \quad \mathbf{b_2} = (a\cos(\theta+\Phi), a\sin(\theta+\Phi)) \tag{G.31}$$

Here, θ is the *magic angle*, and the derivation of it is discussed here. We start with rewriting Eq. (G.23) for $j = 1$ and 2 as,

$$(\mathbf{N} - \lambda_1\mathbf{I}) \cdot \mathbf{c_1} = \mathbf{0} \tag{G.32}$$

$$(\mathbf{N} - \lambda_2\mathbf{I}) \cdot \mathbf{c_2} = \mathbf{0} \tag{G.33}$$

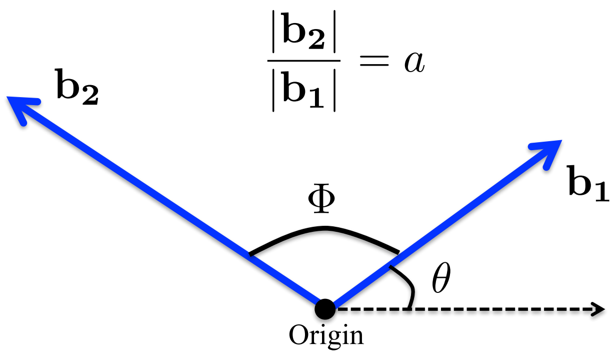


Figure G.2: A general 2-D lattice with magic angle θ

For a 2-D system, Eq. (G.32) can be written as two set of equations. In this case, $\lambda_1 = \phi$ and $\mathbf{c_1} = (\cos\theta, a\cos(\theta + \Phi))$, and the two equations are

$$(N_{11} - \phi)\cos\theta + aN_{12}\cos(\theta + \Phi) = 0 \tag{G.34}$$

$$N_{21}\cos\theta + a(N_{22} - \phi)\cos(\theta + \Phi) = 0 \tag{G.35}$$

Similarly, for $j = 2$, $\lambda_2 = 1/\phi$ and $\mathbf{c_2} = (\sin\theta, a\sin(\theta + \Phi))$, and the corresponding two equations are

$$(N_{11} - 1/\phi)\sin\theta + aN_{12}\sin(\theta + \Phi) = 0 \tag{G.36}$$

$$N_{21}\sin\theta + a(N_{22} - 1/\phi)\sin(\theta + \Phi) = 0 \tag{G.37}$$

Note that for planar flows, $N_{13} = N_{23} = N_{31} = N_{32} = 0$, and $N_{33} = 1$. Equation (G.34) can be solved to obtain an expression for θ in terms of a, Φ, ϕ, N_{11} and N_{12},

$$\theta = \tan^{-1}\left[\frac{(N_{11} - \phi) + aN_{12}\cos\Phi}{aN_{12}\sin\Phi}\right] \tag{G.38}$$

Here ϕ can be calculated using Eq. (G.30), and a and Φ are also known for any specific lattice. For instance, for a square lattice $a = 1$ and $\Phi = \pi/2$, and for a hexagonal lattice $a = 1$ and $\Phi = \pi/3$. Hence, the only unknowns in Eq. (G.38) are N_{11} and N_{12}. An expression for θ is also obtained from Eq. (G.36) as given below,

$$\theta = \tan^{-1}\left[\frac{-aN_{12}\sin\Phi}{(N_{11} - 1/\phi) + aN_{12}\cos\Phi}\right] \tag{G.39}$$

Now, Eqs. (G.38) and (G.39) can be equated to give

$$\begin{aligned}[(N_{11} - \phi) + aN_{12}\cos\Phi]\,[(N_{11} - 1/\phi) + aN_{12}\cos\Phi] &= \\ -\,[aN_{12}\sin\Phi]\,[aN_{12}\sin\Phi]&\end{aligned} \tag{G.40}$$

Or,

$$\begin{gathered}N_{11}^2 - N_{11}\,(\phi + 1/\phi) + 2a\cos\Phi N_{11}N_{12} + 1 \\ -aN_{12}\cos\Phi\,(\phi + 1/\phi) + a^2N_{12}^2 = 0\end{gathered} \tag{G.41}$$

Note that for planar elongational flows, $\phi + (1/\phi)$ is equal to $N_{11} + N_{22}$ since $N_{33} = 1$. Further, we can write
$m = \lambda_1 + \lambda_2 + \lambda_1\lambda_2 = \phi + (1/\phi) + 1 = N_{11}N_{22} + N_{11} + N_{22} - N_{12}N_{21}$,
or
$N_{11}N_{22} - N_{12}N_{21} = \lambda_1\,\lambda_2 = 1$. As a result, Eq. (G.41) can be simplified to give,

$$-\,N_{21} + a\cos\Phi(N_{11} - N_{22}) + a^2N_{12} = 0 \tag{G.42}$$

We define $\tilde{k} = N_{11} + N_{22}$, and note that $N_{11}N_{22} - N_{12}N_{21} = 1$. This leads to equations for N_{22} and N_{21} in terms of $\tilde{k}$, N_{11} and N_{12}, as given below,

$$N_{22} = \tilde{k} - N_{11} \tag{G.43}$$

$$N_{21} = \frac{N_{11}(\tilde{k} - N_{11}) - 1}{N_{12}} \tag{G.44}$$

Note that $\tilde{k} \geq 2$. Inserting N_{22} and N_{21} into Eq. (G.42) gives an equation in terms of a, Φ, $\tilde{k}$, N_{11} and N_{12},

$$\left(N_{11} - \tilde{k}/2\right)^2 + 2a\cos\Phi\left(N_{11} - \tilde{k}/2\right) N_{12} + a^2 N_{12}^2 = \tilde{k}^2/4 - 1 \tag{G.45}$$

This is also an equation of ellipse centered at $(\tilde{k}/2, 0)$. For a square lattice ($a = 1$, $\Phi = \pi/2$), Eq. (G.45) becomes an equation of circle,

$$\left(N_{11} - \tilde{k}/2\right)^2 + N_{12}^2 = \tilde{k}^2/4 - 1 \tag{G.46}$$

and Eq. (G.42) simplifies to $N_{21} = N_{12}$. Equation (G.46) can be used to find an expression for N_{12} in terms of $\tilde{k}$ and N_{11} as,

$$N_{12} = \pm\sqrt{N_{11}(\tilde{k} - N_{11}) - 1} \tag{G.47}$$

If, for a given N_{11}, we find an integer value of N_{12}, then we have found the solution. As mentioned by Kraynik and Reinelt (1992), the negative root is always considered in Eq. (G.47). It is conventional to simulate PEF with square lattices, which we also do in this work.

G.6 Step-by-Step Implementation for a Square Lattice

Finally, in order to implement all these steps in a computer program, we summarize these results in a step-by-step procedure as follows:

1. First choose an integer value of $\tilde{k}$ ($\tilde{k} = 3, 4, 5, \ldots$).

2. Calculate the eigenvalue ϕ using the following expression

$$\phi = \frac{\tilde{k} + \sqrt{\tilde{k}^2 - 4}}{2} \tag{G.48}$$

3. Estimate the strain period τ_p using the expression $\tau_p = \log(\phi)/\dot{\epsilon}$, where $\dot{\epsilon}$ is the elongational rate.

4. Select a positive integer value of N_{11} such that an integer value of N_{12} is obtained using the following expression

$$N_{12} = -\sqrt{N_{11}(\tilde{k} - N_{11}) - 1} \tag{G.49}$$

5. Calculate the magic angle using the expression

$$\theta = \tan^{-1}\left[\frac{N_{11} - \phi}{N_{12}}\right] \tag{G.50}$$

   Note that this equation is a simplified form of Eq. (G.38) for a square lattice.

Therefore, we know the initial lattice configuration and the time at which the lattice should be deformed and mapped back to its original configuration.

Appendix H

Compatibility Condition in Planar Mixed Flow

Since lattices evolve only in the xy plane for planar flows, one corner of the simulation box can be set at the origin and $\mathbf{r_1}(t)$, $\mathbf{r_2}(t)$ and $\mathbf{r_3}(t)$ can be considered to be the other three corners, as shown in Fig. H.1. The lattice spacing $D(t)$ is in turn expressed as the modulus of the sum of the three vectors $\mathbf{r_1}$, $\mathbf{r_2}$ and $\mathbf{r_3}$ (which are three of the corners in the cell). As indicated by the black dashed lines in Fig. H.1, the trajectories of any cell vectors can be used to derive the minimum length in the direction of the cell vectors. Considering $\mathbf{r} = \mathbf{r_1} + \mathbf{r_2} + \mathbf{r_3}$, expressions for $r_x(t)$ and $r_y(t)$ for PMF are given in Eqs. (7.37) and (7.35) respectively. Note that r_x and r_y are the x and y components of the vector $\mathbf{r}$, respectively. $D(t)$ reaches its minimum value when,

$$\frac{d}{dt}(\mathbf{r}(t) \cdot \mathbf{r}(t)) = 0 \tag{H.1}$$

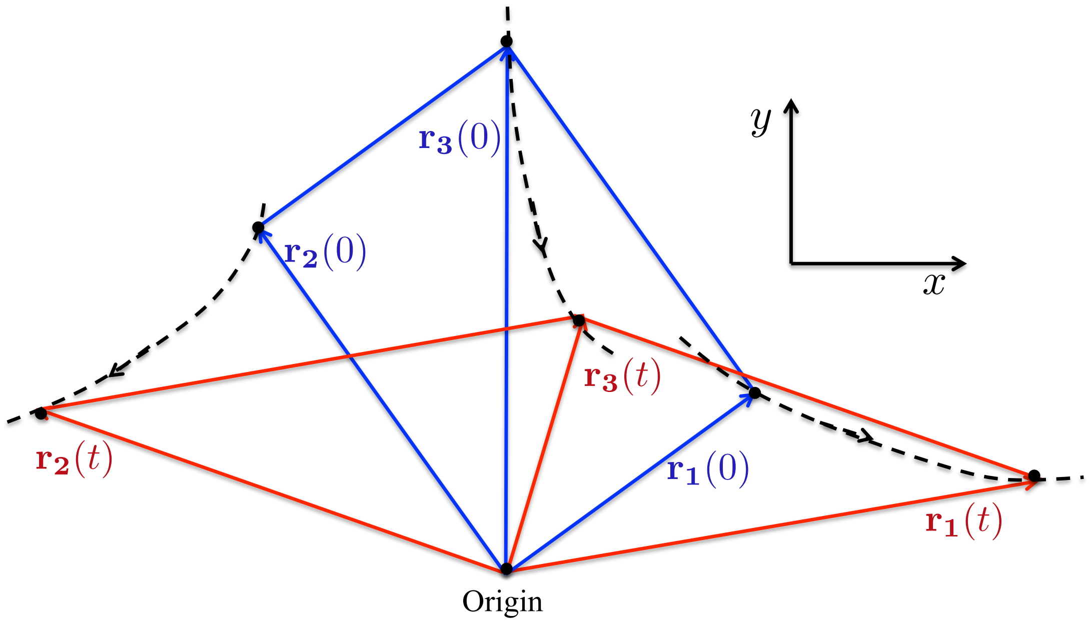


Figure H.1: Schematic illustration of how the corners of a simulation box evolve in the xy plane (reproduced from Hunt et al. (2010)). Blue indicates the initial simulation box, red indicates the simulation box at time t and the black dashed lines represent the trajectories of cell vectors.

This implies,

$$\frac{d}{dt}(\mathbf{r}(t)\cdot\mathbf{r}(t)) = 2r_x(t)\dot{r_x}(t) + 2r_y(t)\dot{r_y}(t) + 2r_z(t)\dot{r_z}(t) = 0 \tag{H.2}$$

Since for planar flows, $\dot{r_z}(t) = 0$, we can write

$$r_x(t)\dot{r_x}(t) + r_y(t)\dot{r_y}(t) = 0 \tag{H.3}$$

Using Eqs. (7.37) and (7.35), the derivatives of $r_x(t)$ and $r_y(t)$ are

$$\dot{r_x}(t) = \dot{\gamma}\, r_y^0 \cosh(\dot{\epsilon}t) + \dot{\epsilon}\, r_x^0 \exp(\dot{\epsilon}t) \tag{H.4}$$

and

$$\dot{r_y}(t) = -\dot{\epsilon}\, r_y^0 \exp\left(-\dot{\epsilon} t\right) \tag{H.5}$$

Putting together $r_x(t)$, $r_y(t)$, $\dot{r_x}(t)$ and $\dot{r_y}(t)$ in Eq. (H.3) leads to

$$\begin{aligned} \left[\frac{\dot{\gamma}}{\dot{\epsilon}} r_y^0 \sinh(\dot{\epsilon} t) + r_x^0 \exp\left(\dot{\epsilon} t\right)\right] \left[\dot{\gamma}\, r_y^0 \cosh(\dot{\epsilon} t) + \dot{\epsilon}\, r_x^0 \exp\left(\dot{\epsilon} t\right)\right] \\ + \left[r_y^0 \exp\left(-\dot{\epsilon} t\right)\right] \left[-\dot{\epsilon}\, r_y^0 \exp\left(-\dot{\epsilon} t\right)\right] = 0 \end{aligned} \tag{H.6}$$

Equation (H.6) can be simplified by using trigonometric formulas and also using the definitions of $\sinh(x)$, $\cosh(x)$ in terms of exponentials $\exp(x)$, to give

$$\exp\left(2\dot{\epsilon} t\right) \left[\frac{\dot{\gamma}^2\, r_y^{0^2}}{4\dot{\epsilon}} + \dot{\gamma}\, r_x^0\, r_y^0 + \dot{\epsilon} r_x^{0^2}\right] = \exp\left(-2\dot{\epsilon} t\right) \left[\frac{\dot{\gamma}^2\, r_y^{0^2}}{4\dot{\epsilon}} + \dot{\epsilon}\, r_y^{0^2}\right] \tag{H.7}$$

Or,

$$\exp\left(4\dot{\epsilon} t\right) = \frac{\dot{\gamma}^2\, r_y^{0^2} + 4\dot{\epsilon}^2\, r_y^{0^2}}{\dot{\gamma}^2\, r_y^{0^2} + 4\dot{\gamma}\, \dot{\epsilon}\, r_x^0\, r_y^0 + 4\dot{\epsilon}^2\, r_x^{0^2}} \tag{H.8}$$

Denoting by $t_{\min}$, the time at which the lattice spacing is a minimum, we can see that,

$$t_{\min} = \frac{1}{4\dot{\epsilon}} \log\left[\frac{\dot{\gamma}^2\, r_y^{0^2} + 4\dot{\epsilon}^2\, r_y^{0^2}}{\dot{\gamma}^2\, r_y^{0^2} + 4\dot{\gamma}\, \dot{\epsilon}\, r_x^0\, r_y^0 + 4\dot{\epsilon}^2\, r_x^{0^2}}\right] \tag{H.9}$$

Therefore, the minimum lattice spacing $D_{\min}$ can be found to be

$$D_{\min} = \sqrt{r_x^2(t_{\min}) + r_y^2(t_{\min})} \tag{H.10}$$

Using Eqs. (7.35), (7.37), (H.9) and (H.10), the minimum lattice spacing can be calculated, and hence a suitable cutoff radius r_c can be chosen to ensure the compatibility condition is satisfied.

Bibliography

Ahlrichs, P. and B. Dünweg, 1999: Simulation of a single polymer chain in solution by combining Lattice Boltzmann and molecular dynamics. *J.Chem.Phys.*, **111**, 8225.

Ahlrichs, P., R. Everaers, and B. Duenweg, 2001: Screening of hydrodynamic interactions in semidilute polymer solutions: A computer simulation study. *Phys.Rev. E*, **64**, 040501.

Allen, M. and D. Tildesley, 1990: *Computer Simulations of Liquids.* Oxford Science, London.

Banchio, A. J. and J. F. Brady, 2003: Accelerated Stokesian dynamics: Brownian motion. *J. Chem. Phys.*, **118**(22), 10323–10332.

Baranyai, A. and P. T. Cummings, 1995: Nonequilibrium molecular dynamics study of shear and shear-free flows in simple fluids. *J.Chem.Phys.*, **103**(23), 10217–10225.

Baranyai, A. and P. T. Cummings, 1999: Steady state simulation of planar elongation flow by nonequilibrium molecular dynamics. *J.Chem.Phys.*, **110**(1), 42–45.

Beenakker, C. W. J., 1986: Ewald sum of the Rotne-Prager tensor. *J.Chem.Phys.*, **85**, 1581–1582.

Bhupathiraju, R., P. T. Cummings, and H. D. Cochran, 1996: An efficient parallel algorithm for non-equilibrium molecular dynamics simulations of very large systems in planar Couette flow. *Mol.Phys.*, **88**(6), 1665–1670.

Binder, K., 1995: *Monte Carlo and Molecular Dynamics Simulations in Polymer Science*. Oxford University Press, USA.

Bird, R. B., C. F. Curtiss, R. C. Armstrong, and O. Hassager, 1987: *Dynamics of Polymeric Liquids*, volume 2. John Wiley and Sons, New York.

Bixon, M., 1976: Polymer dynamics in solution. *Annu. Rev. Phys. Chem.*, **27**(1), 65–84.

Brady, J. F., R. J. Phillips, J. C. Lester, and G. Bossis, 1988: Dynamic simulation of hydrodynamically interacting suspensions. *J. Fluid.Mech.*, **195**, 257–280.

Champeney, D. C., 1973: *Fourier Transforms and Their Physical Applications*. Academic, New York.

Clisby, N., 2010: Accurate estimate of the critical exponent ν for self-avoiding walks via a fast implementation of the pivot algorithm. *Phys. Rev. Lett.*, **104**, 055702.

Cloizeaux, J. and G. Jannink, 2010: *Polymers in Solution: Their Modelling and Structure*. Oxford Classic Texts in the Physical Sciences. OUP Oxford.

Daivis, P. J., M. L. Matin, and B. D. Todd, 2003: Nonlinear shear and elongational rheology of model polymer melts by non-equilibrium molecular dynamics. *J. Non-Newtonian Fluid Mech.*, **111**, 1–18.

Daivis, P. J. and D. N. Pinder, 1990: Application of the blob model to the calculation of cooperative and self-diffusion coefficients in polymer solutions. *Macromolecules*, **23**(25), 5176–5181.

Daoud, M., J. P. Cotton, B. Farnoux, G. Jannink, G. Sarma, H. Benoit, C. Duplessix, C. Picot, and P. G. de Gennes, 1975: Solutions of flexible polymers. neutron experiments and interpretation. *Macromolecules*, **8**(6), 804–818.

de Gennes, P. G., 1976a: Dynamics of entangled polymer solutions. I. the Rouse model. *Macromolecules*, **9**(4), 587–593.

de Gennes, P. G., 1976b: Dynamics of entangled polymer solutions. II. inclusion of hydrodynamic interactions. *Macromolecules*, **9**, 594.

de Gennes, P. G., 1979: *Scaling Concepts in Polymer Physics*. Cornell University, Ithaca, New York.

Deserno, M. and C. Holm, 1998: How to mesh up Ewald sums. i. a theoretical and numerical comparison of various particle mesh routines. *J.Chem.Phys.*, **109**, 7678.

Doi, M. and S. F. Edwards, 1986: *The Theory of Polymer Dynamics*. Clarendon Press, Oxford, New York.

Dua, A. and B. J. Cherayil, 2003: Polymer dynamics in linear mixed flow. *J.Chem.Phys.*, **119**(11), 5696–5700.

Dünweg, B. and A. J. C. Ladd, 2009: Lattice Boltzmann simulations of soft matter systems. *Adv. Polym. Sci.*, **221**, 89.

Edwards, S. F. and M. Muthukumar, 1984: Brownian dynamics of polymer solutions. *Macromolecules*, **17**, 586.

Ewald, P., 1921: . *Ann. Phys.*, **64**, 253.

Ewen, B. and D. Richter, 1997: Neutron spin echo investigations on the segmental dynamics of polymers in melts, networks and solutions. *Adv. Polym. Sci.*, **134**, 1–129.

Fincham, D., 1994: Optimization of the Ewald sum for large systems. *Mol. Sim.*, **13**, 1.

Fixman, M., 1986: Construction of Langevin forces in the simulation of hydrodynamic interaction. *Macromolecules*, **19**, 1204 – 1207.

Flory, P. J., 1953: *Principles of Polymer Chemistry.* Baker lectures 1948. Cornell University Press.

Fredrickson, G. H. and E. Helfand, 1990: Collective dynamics of polymer solutions. *J.Chem.Phys.*, **93**(3), 2048–2061.

Freed, K. F., 1987: *Renormalization Group Theory of Macromolecules.* Wiley, New York.

Freed, K. F. and S. F. Edwards, 1974: Polymer viscosity in concentrated solutions. *J. Chem. Phys.*, **61**(9), 3626–3633.

Frenkel, D. and B. Smit, 2002: *Understanding Molecular Simulation.* Academic Press, London.

Fuller, G. and L. Leal, 1981: The effects of conformation-dependent friction and internal viscosity on the dynamics of the nonlinear dumbbell model for a dilute polymer solution. *Journal of Non-Newtonian Fluid Mechanics*, **8**(3 - 4), 271 – 310.

Gompper, G., T. Ihle, D. Kroll, and R. Winkler, 2009: Multi-particle collision dynamics: A particle-based mesoscale simulation approach to the hydrodynamics of complex fluids. *Adv. Polym. Sci.*, **221**, 1–87.

Graessley, W. W., R. C. Hayward, and G. S. Grest, 1999: Excluded-volume effects in polymer solutions. 2. comparison of experimental results with numerical simulation data. *Macromolecules*, **32**(10), 3510–3517.

Grest, G. S., B. Dünweg, and K. Kremer, 1989: Vectorized link cell fortran code for molecular dynamics simulations for a large number of particles. *Comput. Phys. Commun.*, **55**(3), 269 – 285.

Griebel, M., S. Knapek, and G. Zumbusch, 2007: *Numerical Simulation in Molecular Dynamics.* Springer, Berlin, Heidelberg.

Grosberg, A. Y. and A. R. Khokhlov, 1994: *Statistical physics of macromolecules.* AIP Press, New York.

Hansen, D. P. and D. J. Evans, 1994: A parallel algorithm for nonequilibrium molecular dynamics simulation of shear flow on distributed memory machines. *Mol. Sim.*, **13**(6), 375–393.

Hasimoto, M., 1959: On the periodic fundamental solutions of the Stokes equations and their applications to viscous flow past a cubic array of spheres. *J. Fluid Mech.*, **5**, 317.

Hernández-Ortiz, J. P., J. J. de Pablo, and M. D. Graham, 2007: Fast computation of many-particle hydrodynamic and electrostatic interactions in a confined geometry. *Phys. Rev. Lett.*, **98**, 140602.

Heyes, D., 1985: Molecular dynamics simulations of extensional, shear and unidirectional flow. *Chem. Phys.*, **98**(1), 15 – 27.

Hockney, R. W., S. P. Goel, and J. W. Eastwood, 1973: A 10000 particle molecular dynamics model with long range forces. *Chem. Phys. Lett.*, **21**, 589.

Hoffman, B. D. and E. S. G. Shaqfeh, 2007: The dynamics of the coil-stretch transition for long, flexible polymers in planar mixed flows. *J. Rheol.*, **51**(5), 947–969.

Hounkonnou, M. N., C. Pierleoni, and J.-P. Ryckaert, 1992: Liquid chlorine in shear and elongational flows: A nonequilibrium molecular dynamics study. *J.Chem.Phys.*, **97**(12), 9335–9344.

Huang, C. C., R. G. Winkler, G. Sutmann, and G. Gompper, 2010: Semidilute polymer solutions at equilibrium and under shear flow. *Macromolecules*, **43**(23), 10107–10116.

Hunt, T. A., S. Bernardi, and B. D. Todd, 2010: A new algorithm for extended nonequilibrium molecular dynamics simulations of mixed flow. *J.Chem.Phys.*, **133**(15), 154116.

Hur, J., E. S. G. Shaqfeh, H. P. Babcock, D. E. Smith, and S. Chu, 2001: Dynamics of dilute and semidilute DNA solutions in the start-up of shear flow. *J. Rheol.*, **45**, 421.

Jain, A., B. Dünweg, and J. R. Prakash, 2012: Dynamic crossover scaling in polymer solutions. *Phys. Rev. Lett.*, **109**, 088302.

Jain, A., P. Sunthar, B. Dünweg, and J. R. Prakash, 2012: Optimization of a Brownian-dynamics algorithm for semidilute polymer solutions. *Phys. Rev. E*, **85**, 066703.

Jendrejack, R. M., M. D. Graham, and de Pablo J. J., 2000: Hydrodynamic interactions in long-chain polymers: Application of the Chebyshev polynomial approximation in stochastic simulations. *J.Chem.Phys.*, **113**, 2894.

Kapral, R., 2008: *Multiparticle Collision Dynamics: Simulation of Complex Systems on Mesoscales.* 89–146.

Kirkwood, J. G. and J. Riseman, 1948: The intrinsic viscosities and diffusion constants of flexible macromolecules in solution. *J.Chem.Phys.*, **16**(6), 565–573.

Kozer, N., Y. Y. Kuttner, G. Haran, and G. Schreibe, 2007: Protein-protein association in polymer solutions: From dilute to semidilute to concentrated. *Biophys. J .*, **92**, 2139–2149.

Kraynik, A. M. and D. A. Reinelt, 1992: Extensional motions of spatially periodic lattices. *Int. J. Multiphase Flow*, **18**, 1045.

Kroger, M., A. Alba-Perez, M. Laso, and H. C. Ottinger, 2000: Variance reduced Brownian simulation of a bead-spring chain under steady shear flow considering hydrodynamic interaction effects. *J. Chem. Phys.*, **113**, 4767–4773.

Kumar, K. S. and J. R. Prakash, 2003: Equilibrium swelling and universal ratios in dilute polymer solutions: An exact Brownian dynamics simulations for a delta function excluded volume potential. *Macromolecules*, **36**(20), 7842–7856.

Ladd, A. J. C., R. Kekre, and J. E. Butler, 2009: Comparison of the static and dynamic properties of a semiflexible polymer using Lattice Boltzmann and Brownian-dynamics simulations. *Phys. Rev. E*, **80**, 036704.

Larson, R., 1988: *Constitutive Equations for Polymer Melts and Solutions.* Biotechnology Series. Butterworths (Canada) Limited.

Lees, A. W. and S. F. Edwards, 1972: The computer studies of transport processes under extreme conditions. *J. Phys. C: Solid State Phys.*, **5**, 1921–1929.

Li, B., N. Madras, and A. D. Sokal, 1995: Critical exponents, hyperscaling, and universal amplitude ratios for two- and three-dimensional self-avoiding walks. *J. Stat. Phys.*, **80**(3-4), 661–754.

Luty, B., M. Davis, I. Tironi, and W. van Gunsteren, 1994: A comparison of particle-particle, particle-mesh and Ewald methods for calculating electrostatic interactions in periodic molecular systems. *Mol. Sim.*, **14**, 11–20.

Malevanets, A. and R. Kapral, 1999: Mesoscopic model for solvent dynamics. *J. Chem. Phys.*, **110**(17), 8605–8613.

Mangenota, S., S. Kellera, and J. Radler, 2003: Transport of nucleosome core particles in semidilute DNA solutions. *Biophys. J* ., **85**, 1817–1825.

Marmonier, M. F. and L. Leger, 1985: Reptation and tube renewal in entangled polymer solutions. *Phys. Rev. Lett.*, **55**, 1078–1081.

Miyaki, Y. and H. Fujita, 1981: Excluded-volume effects in dilute polymer solutions. 11. tests of the two-parameter theory for radius of gyration and intrinsic viscosity. *Macromolecules*, **14**(3), 742–746.

Müller, M., K. Binder, and L. Schäfer, 2000: Intra- and interchain correlations in semidilute polymer solutions: Monte Carlo simulations and renormalization group results. *Macromolecules*, **33**(12), 4568–4580.

Muthukumar, M., 1984: Concentration dependent relaxation times of linear polymers in dilute solutions. *Macromolecules*, **17**(4), 971–973.

Muthukumar, M. and S. Edwards, 1982a: Screening concepts in polymer solution dynamics. *Polymer*, **23**(3), 345 – 348.

Muthukumar, M. and S. F. Edwards, 1982b: Extrapolation formulas for polymer solution properties. *J.Chem.Phys.*, **76**(5), 2720–2730.

Muthukumar, M. and S. F. Edwards, 1983: Screening of hydrodynamic interaction in a solution of rodlike macromolecules. *Macromolecules*, **16**(9), 1475–1478.

Nijboer, B. R. A. and F. W. de Wette, 1957: On the calculation of lattice sums. *Physica.*, **23**, 309–321.

Öttinger, H. C., 1989: Gaussian approximation for rouse chains with hydrodynamic interaction. *The Journal of Chemical Physics*, **90**(1), 463–473.

Ottinger, H. C., 1996: *Stochastic Processes in Polymeric Fluids*. Springer, Berlin.

Pan, S., D. A. Nguyen, P. Sunthar, T. Sridhar, and J. R. Prakash Universal solvent quality crossover of the zero shear rate viscosity of semidilute DNA solutions. 2011arXiv1112.3720P.

Perram, J., H. Peterson, and K. De Leeuw, 1988: An algorithm for the simulation of condensed matter which grows as the 3/2 power of the number of particles. *Mol. Phys.*, **65**, 875.

Pham, T. T., U. D. Schiller, J. R. Prakash, and B. Dunweg, 2009: Implicit and explicit solvent models for the simulation of a single polymer chain in solution: Lattice Boltzmann versus Brownian dynamics. *J. Chem. Phys.*, **131**, 164114.

Pouyet, G., F. Candau, and J. Dayantis, 1976: Sedimentation measurements on polystyrene solutions in cyclohexane below the Flory theta temperature. *Die Makromolekulare Chemie*, **177**(10), 2973–2980.

Prabhakar, R., 2005: *Predicting the rheological properties of dilute polymer solutions using bead-spring models: Brownian dynamics simulations and closure approximations*. PhD thesis, Monash University.

Prabhakar, R. and J. R. Prakash, 2004: Multiplicative separation of the influences of excluded volume, hydrodynamic interactions and finite extensibility on the rheological properties of dilute polymer solutions. *J.Non-Newtonian Fluid.Mech.*, **116**, 163.

Prakash, J. R., 1999: The kinetic theory of dilute solutions of flexible polymers: Hydrodynamic interaction. In *Advances in flow and rheology of non-Newtonian fluids*, Siginer, D. A., Kee, D. D., and Chhabra, R. P., editors, Elsevier Science, Rheology Series, Amsterdam, 467–517.

Prakash, J. R., 2001a: The influence of the range of excluded volume interactions on the linear viscoelastic properties of dilute polymer solutions. *Chem. Eng. Sci.*, **56**(19), 5555 – 5564.

Prakash, J. R., 2001b: Rouse chains with excluded volume interactions: Linear viscoelasticity. *Macromolecules*, **34**(10), 3396–3411.

Prakash, J. R., 2002: Rouse chains with excluded volume interactions in steady simple shear flow. *J. Rheol.*, **46**(6), 1353–1380.

Prakash, J. R., 2009: Micro and macro in the dynamics of dilute polymer solutions: Convergence of theory with experiment. *Korea-Australia Rheology Journal*, **21**(4), 245–268.

Prakash, J. R. and H. C. Öttinger, 1999: Viscometric functions for a dilute solution of polymers in a good solvent. *Macromolecules*, **32**(6), 2028–2043.

R. H. Colby, M. Rubinstein, and M. Daoud, 1994: Hydrodynamics of polymer solutions via two-parameter scaling. *J. Phys. II France*, **4**(8), 1299–1310.

Rapaport, D., 2004: *The Art of Molecular Dynamics Simulation.* Cambridge University Press.

Richter, D., K. Binder, B. Ewen, and B. Stuehn, 1984: Screening of hydrodynamic interactions in dense polymer solutions: a phenomenological theory and neutron-scattering investigations. *J. Phys. Chem.*, **88**, 6618.

Rinn, B., K. Zahn, P. Maass, and G. Maret, 1999: Influence of hydrodynamic interactions on the dynamics of long-range interacting colloidal particles. *Europhys. Lett.*, **46**(4), 537.

Rotne, J. and S. Prager, 1969: Variational treatment of hydrodynamic interaction in polymers. *J. Chem. Phys.*, **50**, 4831–4837.

Rubinstein, M. and R. H. Colby, 2003: *Polymer Physics.* Oxford University Press.

Schäfer, L., 1999: *Excluded Volume Effects in Polymer Solutions.* Springer-Verlag, Berlin.

Shiwa, Y., Y. Oono, and P. R. Baldwin, 1988: Dynamics of semidilute polymer solutions: hydrodynamic screening. *Macromolecules*, **21**(1), 208–214.

Sierou, A. and J. F. Brady, 2001: Accelerated Stokesian dynamics simulations. *J. Fluid Mech.*, **448**, 115.

Smith, E. R., I. K. Snook, and W. V. Megen, 1987: Hydrodynamic interactions in Brownian dynamics: 1. periodic boundary conditions for computer simulations. *Physica*, **143A**, 441–467.

Stoltz, C., 2006: *Simulation of dilute polymer and polyelectrolyte solutions: Concentration effects.* PhD thesis, The University of Wisconsin.

Stoltz, C., J. J. de Pablo, and M. D. Graham, 2006: Concentration dependence of shear and extensional rheology of polymer simulations: Brownian dynamics simulations. *J.Rheol.*, **502**, 137.

Sunthar, P. and J. R. Prakash, 2005: Parameter free prediction of DNA conformations in elongational flow by successive fine graining. *Macromolecules*, **38**, 617.

Sunthar, P. and J. R. Prakash, 2006: Dynamic scaling in dilute polymer solutions: The importance of dynamic correlations. *Europhys. Lett.*, **75**, 77.

Thurston, G. B. and A. Peterlin, 1967: Influence of finite number of chain segments, hydrodynamic interaction, and internal viscosity on intrinsic birefringence and viscosity of polymer solutions in an oscillating laminar flow field. *J.Chem.Phys.*, **46**(12), 4881–4885.

Todd, B. and P. J. Daivis, 1999: A new algorithm for unrestricted duration nonequilibrium molecular dynamics simulations of planar elongational flow. *Comput. Phys. Commun.*, **117**(3), 191 – 199.

Todd, B. D. and P. J. Daivis, 1998: Non-equilibrium molecular dynamics simulations of planar elongational flow with spatially and temporally periodic boundary conditions. *Phys. Rev. Lett.*, **81**, 1118.

Todd, B. D. and P. J. Daivis, 2000: The stability of nonequilibrium molecular dynamics simulations of elongational flows. *The Journal of Chemical Physics*, **112**(1), 40–46.

Todd, B. D. and P. J. Daivis, 2007: Homogeneous non-equilibrium molecular dynamics simulations of viscous flow: techniques and applications. *Mol. Sim.*, **33**(3), 189–229.

Toukmaji, A. Y. and J. A. J. Board, 1996: Ewald summation techniques in perspective: A survey. *Comput. Phys. Commun.*, **95**, 73–92.

W. Paul, K. Binder, D. W. Heermann, and K. Kremer, 1991: Crossover scaling in semidilute polymer solutions: a Monte Carlo test. *J. Phys. II France*, **1**(1), 37–60.

Wheeler, D. R., N. G. Fuller, and R. L. Rowley, 1997: Non-equilibrium molecular dynamics simulation of the shear viscosity of liquid methanol: Adaptation of the Ewald sum to Lees Edwards boundary conditions. *Mol. Phys.*, **92**, 55–62.

Wiltzius, P. and D. S. Cannell, 1986: Wave-vector dependence of the initial decay rate of fluctuations in polymer solutions. *Phys. Rev. Lett.*, **56**, 61.

Wiltzius, P., H. R. Haller, D. S. Cannell, and D. W. Schaefer, 1984: Dynamics of long-wavelength concentration fluctuations in solutions of linear polymers. *Phys. Rev. Lett.*, **53**, 834–837.

Woo, N. J. and E. S. G. Shaqfeh, 2003: The configurational phase transitions of flexible polymers in planar mixed flows near simple shear. *J.Chem.Phys.*, **119**(5), 2908–2914.

Yamakawa, H., 1970: Transport properties of polymer chains in dilute solution: Hydrodynamic interaction. *J. Chem. Phys.*, **53**(1), 436–443.

Yamakawa, H., 1971: *Modern theory of polymer solutions.* Harper's chemistry series. Harper & Row.

Yamakov, V., A. Milchev, and K. Binder, 1997: Inter-chain structure factors of flexible polymers in solutions: A Monte Carlo investigation. *J. Phys. II France*, **7**(8), 1123–1139.

Zhang, K. J., M. E. Briggs, R. W. Gammon, J. V. Sengers, and J. F. Douglas, 1999: Thermal and mass diffusion in a semidilute good solvent-polymer solution. *J. Chem. Phys.*, **111**, 2270.

Zhou, T. and S. B. Chen, 2006: Computer simulations of diffusion and dynamics of short-chain polyelectrolytes. *The Journal of Chemical Physics*, **124**(3), 034904.

Zimm, B. H., 1956: Dynamics of polymer molecules in dilute solution: Viscoelasticity, flow birefringence and dielectric loss. *J.Chem.Phys.*, **24**(2), 269–278.