FAST ALGORITHMS FOR BROWNIAN DYNAMICS WITH HYDRODYNAMIC INTERACTIONS

Wenhua Guan

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics in the College of Arts and Sciences.

Chapel Hill
2016

Approved by:

Jingfang Huang

Gary A. Huber

Laura Miller

Katie Newhall

Jan F. Prins

ABSTRACT

Wenhua Guan: Fast Algorithms for Brownian Dynamics with
Hydrodynamic Interactions
(Under the direction of Jingfang Huang)

In this dissertation, we contribute on three fundamental parts of Brownian dynamics simulations with hydrodynamic interactions. The first part of the dissertation is to derive the formulas for computing the electric field gradients by the new version of fast multipole method(FMM) [1] and to implement them as new functions for existing FMM solvers. In the second part of the dissertation, we discuss how to decompose the far-field Rotne-Prager-Yamakawa potential into four far-field Laplace FMM calls including electrostatic potential, electric field and field gradient terms. A parallelized Rotne-Prager-Yamakawa solver based on the new version of fast multipole method has been developed with tunable accuracy. The solver makes it computationally viable for large-scale, long-time Brownian dynamic simulations with hydrodynamic interactions. In the third part, a model is built toward an accurate description of hydrodynamic effects on the translational and rotational dynamics of complex, rigid macromolecules with arbitrary shape in suspension. The grand diffusion matrix is calculated by employing the bead-shell model for describing the shape and structure of macromolecules in the many-body system. Two fast algorithms based on block conjugate gradient method and the Schur complement method are developed for computing the translational and angular velocities, as well as the displacements and orientations in order to track the trajectories of the macromolecules in the complex structured biological systems.

To my family!

ACKNOWLEDGEMENTS

Many thanks to the members of the Department of Mathematics of the University of North Carolina at Chapel Hill for their instruction, support, and encouragement.

I wish to single out my advisor, Dr. Jingfang Huang, for his exceptional patience, valuable guidance and generous support during my graduate study. Dr. Huang gave me a lot of freedom and allowed me to explore on my own, which greatly improved my ability to work independently and benefited me for my future career. His pursuit of optimal numerical results and dedication to research has a big influence on me and stimulated me. I am grateful to him for holding me to a high research standard which guaranteed the quality of work presented here.

I also wish to thank my committee, including Dr. Gary Huber, Dr. Laura Miller, Dr. Katie Newhall and Dr. Jan Prins. I am grateful to Dr. Jan Prins for providing me intelligent advice for this dissertation, for teaching me a lot of knowledge about parallel computing, and for giving me many suggestions about the job hunting. Discussion with him every time was full of pleasure and knowledge. His professionalism, dedication and kindness has greatly inspired me and encouraged me. I am thankful to my collaborator, Dr. Gary Huber for broadening my knowledge in biochemistry and biology, for reading and commenting on this dissertation. I also wish to thank Dr. Laura Miller for introducing me job opportunities, for writing references, and for helping me along the way. I thank Dr. Katie Newhall for her time and enthusiasm.

I thank Dr. Gregory Forest for supporting me in the final summer and taking every effort to help me finish my degree at UNC. I thank Dr. Xiaolin Chen for providing wonderful suggestions on my oral slides.

I would like to express my deep gratitude and appreciation to Miss Laurie Strabue for offering me a lot of friendly help. Her support and care helped me overcome setbacks and focus on my graduate study. I greatly value the friendship and the beautiful memories we have made together. I am grateful to Dr. Betty Davidson for spending her time to improve my communication skills, for providing wonderful suggestions to my dissertation. This dissertation has been greatly improved through the careful grammatical editing by Miss Strabue and Dr. Davidson. I would also thank the joyful moments from the lunch group(Dan, Feifei, Tim, Yan, Yuan, Wayne etc), as well as their consideration and support. Thank my office mates Yuan and Zeliha. Thank all the other friends who have helped me and encouraged me.

My sincere appreciation also goes to my master advisor, Dr. Bo Yu, for his constant support and encouragement all the time.

Most importantly, none of this would happen without the love and encouragement from the family. Thank my parents who raised me to value knowledge and supported me in all of my life endeavors. Their love and parenting helped me foster a solid work ethic and shaped the person I have become. I would like to thank my husband, Ming, who has made all the difference in my life, for believing me, loving me, encouraging me. His wisdom and knowledge enlightened me.

TABLE OF CONTENTS

LIST OF FIGURES . . . . **ix**

LIST OF TABLES . . . . **x**

CHAPTER 1: INTRODUCTION **1**

1.1 Background 1

1.2 Ermak-McCammon Model for Spherical Particles 3

1.3 Brownian Dynamics Model for Rigid Macromolecules of Arbitrary Shape 7

1.4 Outline of the Dissertation 10

CHAPTER 2: COMPUTATION OF ELECTRIC FIELD GRADIENTS BY FAST MULTIPOLE METHOD **13**

2.1 Adaptive Tree Structure for the Spatial Domain 16

2.2 Approximation and Translation 20

2.2.1 Mathematical Preliminaries 20

2.2.2 Approximation Operators 23

2.2.3 Translation Operators 24

2.3 Local Expansions of the Fields and Electric Field Gradients 29

2.3.1 Electrostatic Fields 29

2.3.2 The far-field Electric Field Gradients 32

2.4 Algorithm for Adaptive FMM 37

2.5 Numerical Results 41

2.5.1 Cube 42

2.5.2 Surface of A Unit Sphere 45

2.5.3 First Octant of the Unit Sphere 47

CHAPTER 3: ADAPTIVE ROTNE-PRAGER-YAMAKAWA SOLVER BY MULTIPOLE METHODS 49

3.1 Rotne-Prager-Yamakawa Tensor 52

3.1.1 Far-field Rotne-Prager-Yamakawa Evaluation with Multipole Methods 53

3.1.2 Near-field Rotne-Prager-Yamakawa Potential Evaluation 55

3.2 Adaptive Rotne-Prager-Yamakawa Algorithm with Multipole Methods 56

3.3 Numerical Results 57

3.3.1 Cylinder 57

3.3.2 Eight Surface 58

CHAPTER 4: HYDRODYNAMIC INTERACTIONS OF COMPLEX, RIGID, BIOLOGICAL MACROMOLECULES IN SUSPENSION 63

4.1 Hydrodynamic Interactions 65

4.2 Bead-Shell Model 65

4.3 Mathematical Formulation of the many-body System in Suspensions with Bead Model 66

4.4 Fast Algorithms for the many-body System 70

4.4.1 Block Conjugate Gradient Method 71

4.4.2 Numerical Results of the Block Conjugate Gradient Method 71

4.4.3 Schur Complement Method 73

4.4.4 Numerical Results of Schur Complement Method 75

CHAPTER 5: CONCLUSION 78

REFERENCES 81

LIST OF FIGURES

2.1 Blue boxes are colleagues of box b and white boxes are the interaction list of b. . . . 12

2.2 Box b and its associated lists in two dimension 13

2.3 The uplist of Box b [2] 14

2.4 Charges randomly distributed in a cube 33

2.5 Charges randomly distributed on the surface of the sphere 36

2.6 Charges randomly distributed on the surface of the first octant 38

3.1 Particles randomly distributed on the surface of a cylinder 46

3.3 Particles randomly distributed on an eight surface 48

3.2 Speedup of particles randomly distributed on the surface of the cylinder 49

3.4 Speedup of particles randomly distributed on the eight surface 51

4.1 Complex biological macromolecule structures [4] 53

4.2 (A) Atomistic DNA model (B) Bead Model with each DNA residue modeled with a single pseudoatom (C) Bead-shell model with surfaces covered with spherical elements(By Maciej Dlugosz [5]) 54

4.3 Bead-shell model of the two body system 59

4.4 Iteration number of the block conjugate gradient method for the hydrodynamics of the two-body system 60

4.5 Bead-shell model of the two body system with different sizes 62

4.6 Iteration number of Schur complement method for the hydrodynamics of the two-body system with different sizes 62

4.7 Comparison of iteration number of Schur complement method with and without preconditioner for the hydrodynamics of the two-body system with different sizes . . 63

4.8 The bead-shell model of the many-body system with different sizes 63

4.9 Comparison of iteration number of Schur complement method with and without preconditioner for the hydrodynamics of the two-body system with different sizes . . 64

LIST OF TABLES

2.1 Error Results of the new version of FMM for 3-digit accuracy with charges randomly distributed in a cube 33

2.2 Error results of the new version of FMM for 6-digit accuracy with charges randomly distributed in a cube 34

2.3 Error results of the new version of FMM for 9-digit accuracy with charges randomly distributed in a cube 34

2.4 Timing results of the new version of FMM with charges randomly distributed in a cube 35

2.5 Error results of the new version of FMM for 3-digit accuracy with charges randomly distributed on the surface of a sphere 36

2.6 Error results of the new version of FMM for 6-digit accuracy with charges randomly distributed on the surface of a sphere 37

2.7 Error results of the new version of FMM for 9-digit accuracy with charges randomly distributed on the surface of a sphere 37

2.8 Timing results of the new version of FMM with charges randomly distributed on the surface of a sphere 37

2.9 Error results of the new version of FMM for 3-digit accuracy with charges randomly distributed on the surface of the first octant 38

2.10 Error results of the new version of FMM for 6-digit accuracy with charges randomly distributed on the surface of the first octant 38

2.11 Error results of the new version of FMM for 9-digit accuracy with charges randomly distributed on the surface of the first octant 39

2.12 Timing results of the new version of FMM with charges randomly distributed on the surface of the first octant 39

3.1 Numerical results of RPY solver for particles randomly distributed on the surface of the cylinder(12 threads) 47

3.2 Running time of RPY solver of 6-digit accuracy for particles randomly distributed on the surface of the cylinder 47

3.3 Speedup of RPY solver of 6-digit accuracy for particles randomly distributed on the surface of the cylinder 48

3.4 Numerical Results of RPY solver for particles randomly distributed on the eight surface (12threads) 49

3.5 Running time of RPY solver of 6-digit accuracy using different threads for particles randomly distributed on the eight surface 50

3.6 Speedup of RPY solver of 6-digit accuracy using different threads for particles randomly distributed on the surface of the eight surface 50

CHAPTER 1

Introduction

1.1 Background

The interiors of all living cells are crowded with biological macromolecules, such as proteins, nuclear acids (RNA, DNA), carbohydrates, polymeric lipids and so on. Macromolecules are constantly moving around by diffusion in biological systems. Diffusion is basically the random motion of molecules which is driven by thermal energy. It is one of the fundamental questions that is being pursued by scientists in molecular biology, cellar biology and beyond. Molecules, macromolecules and nanoparticles, under diffusive transport in a fluid medium, is of interest in many biological applications, such as drug delivery and separation processes [6]. Moreover, a detailed study of the transport, regulation and dynamics of molecules inside cells can help us to understand how healthy cells change to disease states and provide important information to the developers of therapeutic drugs. Micro and nano drug delivery systems are developed to deliver the drug to the desired tissue in the human body so that it increases the efficiency and effectiveness of the treatment and minimizes the side effects associated with the drug [7]. Therefore, the understanding of these mechanisms will also help researchers to develop preemptive treatment, monitored by a wrist-worn device for detecting diseases. Recently, Google has been working on a nanoparticle pill that can identify cancers, heart attacks and other diseases before they become a problem.

There are mainly two classes of methods for simulating the diffusion in biological systems. Traditional one-scale continuum models, such as the Navier-Stokes equation for Newtonian fluids, do not incorporate the size of the biomolecules explicitly, nor do they incorporate the shape changes of the molecules. They are far from capturing the multiscale-multiphysics mechanics in a biological systems. The other class of methods treats the diffusing molecules individually, for example, molecular dynamics and Brownian dynamics methods. Molecular dynamics [8] is a computational method which describes equilibrium and dynamics properties of a biological system, and computes the motions of individual molecules. Hence, the detailed information on the dynamic behaviors of both solute and solvent molecules are included in molecular dynamics at an atomic level of description. The molecular dynamics is more accurate compared with the other coarse-grained models. However, accounting for the effect of the solvent molecules on the solute will require a prohibitive computational cost for long-time simulations [9].

Brownian dynamics is a mesoscopic method for simulating the diffusive behavior of particles which undergoes Brownian motion in fluids. In Brownian dynamics, the solvent is treated as a continuum and the impact of the explicit solvent molecules to the solute molecules are replaced instead by an instantaneous friction force and a fluctuating force. The reason why the Brownian dynamics methods allow one to simulate much larger time scales than in molecular dynamics simulations is because Brownian dynamics techniques are able to coarse-grain out the fast modes of the solvent molecules [10]. The Brownian dynamics method takes the advantage of the large separation in time scales between the rapid motion of solvent molecules and the slow motion of the Brownian particles, the mass and size of which are larger than those of the host medium particles [11]. The core of the Brownian dynamics simulation is a first or second order stochastic differential equation in time for the position of the solute molecules which allows for studying the temporal evolution and dynamics of complex fluids. Brownian dynamics techniques have been widely used in biology, biochemistry, chemical engineering and materials sciences [12, 13]. It has many applications, such as protein folding [14, 15], DNA bending and supercoiling [16, 17, 18], steering in enzyme/substrate encounter [19, 20, 21, 22], and diffusion in crowded cells [23, 24].

An important effect that helps Brownian dynamics algorithms to capture correctly the dynamics of the Brownian particles in dilute solution is the *hydrodynamic interaction*. Hydrodynamic interaction is the dominant factor in this intracellular diffusion and crucial to macromolecular motion in vivo [25]. A particles moving in a viscous fluid will induce a local flow field that will affects the movements of other particles. As a result, the other particles will feel this local flow field and experience the corresponding movements. The movements of the other particles are said to be resulted from hydrodynamic interactions of the original particle. These long-range, nonlinear many-body interactions, mediated by the solvent, are commonly called hydrodynamic interactions (HIs) which depend on the configuration of all particles in fluids. Traditional Brownian dynamics simulations often neglect or simplify the hydrodynamic effects which describe the dynamic correlations between the Brownian particles mediated by fast momentum transfer through the solvent since it will result in high computational cost. It may cause unphysical results as soon as the time scale exceeds a few collision times. Proper treatment of hydrodynamic interactions is essential in simulation studies of Brownian particles in the flow.

Given the ability of Brownian dynamics techniques in bridging different time and length scales, it becomes essential for developing fast algorithms in Brwonian dynamics simulations with hydrodynamic interactions to study the dynamics of macromolecules. Current algorithms for dynamical modeling fail to perform a simulation long enough to observe a large-scale conformational change. The massive number of parameters and iterations required to accurately model a reasonably-sized many-body system of macromolecules at proper resolution are beyond even the processing abilities of supercomputers. Understanding the interplay between different mechanisms at various scales presents formidable challenges to conventional mathematical modeling and solution techniques, and thus requires the development of accurate and robust multiscale models and efficient algorithms for simulations. Below, the Brownian dynamics model with hydrodynamic interactions will be presented for describing the motions of spherically symmetric particles and the associated computation challenges. The rotations of these particles can be neglected due to their symmetric geometry. Based on the model for spherical particles, the Brownian dynamics translational-rotational model for rigid macromolecules of arbitrary shape is described and the corresponding diffusion matrix has been calculated through the deformation of the macromolecules by bead-shell model.

The goal of this dissertation is to derive state-of-the-art fundamental algorithms to advance Brownian dynamics simulations with hydrodynamic interactions and implement the fast algorithms as solvers to accelerate the numerical simulations. These simulations will be suitable for a broad class of flows and physical scales and allowing for further simulating notable complex-structured macromolecules, as well as forcing terms.

1.2 Ermak-McCammon Model for Spherical Particles

Consider a suspension of N identical spherical particles of radius a, immersed in an incompressible Newtonian fluid of viscosity η at low Reynold number. When the inertial relaxation times are short compared to the timescale of interest, it is often possible to ignore inertia in the governing equation.

A Langevin description of the N-particle system is presented below and a random force representing the action of the thermal motion is added to Newton's equation. Rotations of the particles will not be accounted for since they are spherically symmetric. For particle $i, i = 1, \ldots, N$ with translational velocity $\mathbf{V}_i$, from Newton's Second Law, there is

$$m_i \frac{d\mathbf{V}_i}{dt} = \mathbf{F}_i^t = \mathbf{F}_i^h + \mathbf{F}_i^B + \mathbf{F}_i^{nh}. \tag{1.1}$$

Here, m_i is the generalized mass of particle i. The total force $\mathbf{F}_i^t$ on particle i is composed of three parts: the frictional force $\mathbf{F}_i^h$ from the particle moving through the viscous solvent, a Brownian force $\mathbf{F}_i^B$ due to the random successive collision of the solvent molecules with the Brownian particle, and all deterministic non-hydrodynamic force $\mathbf{F}_i^{nh}$ for particle i. The non-hydrodynamic force is also called *systematic force* which includes any external body force and force due to the interaction potential energy between Brownian particles, like the electrostatic interactions and Vander Waals interactions.

The hydrodynamic force $\mathbf{F}_i^h$ acting on particle i tends to decrease the energy and depends on the appropriate components of the configuration-dependent tensor $\boldsymbol{\zeta}$, which is called *hydrodynamic friction tensor*. Due to the linearity of the Stokes equations, the forces exerted by the fluid on the particles depend linearly on the translational velocities of the particles. This relation defines

$$\mathbf{F}_i^h = -\sum_{j=1}^{N} \boldsymbol{\zeta}_{ij} \frac{d\mathbf{x}_j}{dt}. \tag{1.2}$$

Let $\mathbf{F}^h = [\mathbf{F}_1^h, \ldots, \mathbf{F}_N^h]^T$, $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ be the hydrodyanmic forces and locations of all particles. The above equation can be written as

$$\mathbf{F}^h = -\boldsymbol{\zeta} \frac{d\mathbf{x}}{dt}. \tag{1.3}$$

According to the Stokes-Einstein relation, the N-particle diffusion tensor $\mathbf{D}$ and the friction tensor $\boldsymbol{\zeta}$ has relation

$$\mathbf{D} = k_B T \boldsymbol{\zeta}^{-1}, \tag{1.4}$$

where k_B is the Boltzman constant and T is the temperature. The diffusion tensor, a $3N \times 3N$ dense matrix describing the hydrodynamic interactions, is usually modeled by Rotne-Prager-Yamakawa tensor or the Oseen tensor. The difference between these two tensors are elaborated in chapter 3. Notably, both the Roten-Prager-Yamakawa tensor and the Oseen tensor are divergence-free, which considerably simplifies the computation of Brownian dynamics models.

$$\sum_j \frac{\partial \mathbf{D}_{ij}}{\partial \mathbf{r}_j} \equiv 0. \tag{1.5}$$

The Brownian force $\mathbf{F}_i^B(t)$ arising from the thermal fluctuations in the fluid and tending to increase the energy of the particle is characterized by [26]

$$\begin{aligned} < \mathbf{F}_i^B(t) > &= 0 \\ < \mathbf{F}_i^B(t)\mathbf{F}_j^B(s) > &= 2k_BT\boldsymbol{\zeta}_{ij}\delta(t-s). \end{aligned} \tag{1.6}$$

The angle brackets in Equation (1.6) denote an ensemble average. δ is the Dirac's delta function. The amplitude of the correlation between the Brownian forces at t and s results from the fluctuation-dissipation theorem for the N-body system. The time scale Δt researchers are interested in is longer than $\tau = m/6\pi\eta a$ (momentum relax time of the particle after a Brownian impluse) but smaller than the time the configuration changes. The fluctuating forces are considered instantaneous.

The evolution equation for the particles is described by following Ermak-McCammon model [11] by integrating Equation (1.1) over time step Δt. The displacement vector $\mathbf{\Delta x}$ for all small rigid spherical particles during time step Δt due to the non-hydrodynamic force $\mathbf{F}^{nh}$ and the Brownian force $\mathbf{F}^B$ is given by

$$\Delta\mathbf{x}(\Delta t) = \frac{\Delta t}{k_BT}\mathbf{D}\cdot\mathbf{F}^{nh} + \mathbf{R}(\Delta t) + \nabla\cdot\mathbf{D}\Delta t \tag{1.7}$$

$$< R(\Delta t) >= 0, < \mathbf{R}(\Delta t)\mathbf{R}(\Delta t) >= 2\mathbf{D}\Delta t. \tag{1.8}$$

Equation (1.7) can also be obtained from Fokker-Planck or Smolochowski equation for the N-particle probability distribution function. Equation (1.7) is the heart of the Brownian dynamic simulations which describes N particles suspended in unbounded flow interacting through hydrodynamic, interparticle, external, and Brownian forces. It simply states that the displacement of a particle includes three parts. There is a deterministic contribution due to the non-hydrodynamic or systematic force, and two contributions from Brownian motion: a random displacement which makes the fluctuation-dissipation theorem satisfied and a displacement due to the configuration-space divergence of the N-particle diffusivity $\nabla \cdot \mathbf{D}\Delta t$. This term becomes zero when the diffusion matrix $\mathbf{D}$ is modeled by Oseen tensor and Roten-Prager-Yamakawa tensor because they are divergence-free.

For rigid, spherical particles without considering rotations, there are three computational challenges for the Brownian dynamic simulations with hydrodynamic interactions :

1. The electrostatic field around the target molecule is only computed once at the initial step of the simulation due to prohibitive computational cost using existing methods. Such temporal approximation presents major limitations in accuracy of the simulations, especially in the situations when the ligand is not small in size, or the concentration of ions is high (as in real biological ranges), or when boundaries or interfaces are present. The Adaptive Fast Multipole Poisson-Boltzmann (AFMPB) solver [27] is an open-source software developed by Huang and the collaborators for solving the linearized Poisson-Boltzmann (LPB) equation which models the electrostatic interactions in biomolecular systems. AFMPB has significant improvements in computational efficiency and numerical stability over other existing LPB solvers.

2. The first term on the right side of Equation (1.7) is mainly a matrix-vector multiplication between the diffusion matrix and the non-hydrodynamic force. Direct evaluation of it will result in $O(N^2)$ complexity, where N is the number of spherical particles. Additionally, since the diffusion matrix $\mathbf{D}$ for spherical Brownian particles is configuration-dependent and dense, explicit construction of it will result in high memory usage. For example, according to Liu and Chow [28], the memory storage for the diffusion matrix $\mathbf{D}$ of $10K$ particles will be as huge as 32 GB, which is impracticable for large-scale and long-time simulation. To reduce the complexity for diffusion-force matrix-vector evaluation, various fast algorithms have been developed, for example, Ewald summation [29], Particle mesh techniques including the Particle Mesh Ewald (PME) method [30] and the Smooth Particle Mesh Ewald (SPME) method [31] with complexity $O(NlogN)$. The other methods include Method of local corrections [32], multigrid methods [33], panel clustering method and precorrected-FFT [34], and fast multipole method [35].

3. The third challenge for Brownian dynamic simulations is the random process generation. A common method for computing the Brownian displacement is to apply the Cholesky factorization which scales $O(N^3)$ and limits the Brownian dynamics simulation with hydrodynamic

interactions for small scales. Matrix factorization techniques $\mathbf{D} = \mathbf{B} \cdot \mathbf{B}^T$ and $\mathbf{D} = \sqrt{\mathbf{D}} \cdot \sqrt{\mathbf{D}}$ (Davis, 1987 [36]) require forming the diffusion matrix $\mathbf{D}$ explicitly and result in high memory usage. Iterative methods based on Krylov subspace methods and Chebyshev approximations $\sqrt{\mathbf{D}} \cdot F \approx p(\mathbf{D})F$ are practical for large-scale time simulation if the fast evaluation of the matrix-vector multiplication $\mathbf{D} \cdot \mathbf{F}$ is performed properly. Fundamentally, $\mathbf{D} \cdot \mathbf{F}$ will be accelerated tto reduce the overall computation cost for each time step of the Brownian dynamic simulations with hydrodynamic interactions.

In this dissertation, how to apply the new version of the fast multipole method [1] to compute the electric field gradients will be firstly presented. The fast evaluation of the electric field gradients is very important in many areas, such as nuclear quadrupole resonance. Then, for rigid, spherical particles with a small radius, a fast algorithm based on the new version of FMM for computing the matrix-vector multiplication between the diffusion matrix and the non-hydrodynamic force will be described. The Rotne-Prager-Yamakawa tensor is applied to describe the diffusion of the particles with hydrodynamic interactions. The far-field part of the Rotne-Prager-Yamakawa potential is decomposed into four Laplace fast multipole calls combined with the elastic potential term, fields terms and electric field gradients by the developed new version of the fast multipole solver in Chapter 1. The near field of the Rotne-Prager-Yamakawa potential is computed directly. Based on this, the random process generation issue can be resolved by the iterative Krylov subspace methods combining the Chebyshev approximations efficiently.

1.3 Brownian Dynamics Model for Rigid Macromolecules of Arbitrary Shape

Macromolecules in fluids exhibit a variety of complex translational, rotational and bending motions and phase behavior which can illuminate many fundamental issues in statistical mechanics. For macromolecules with complex, ubiquitous structures, the deformation, size and shape are crucial to model the anisotropic diffusivity of the particles in solution. Measurement of the relation time constants associated with the motions is essential for characterizing the molecules and predicting their macroscopic behavior. The rotations of the macromolecules cannot be neglected if the shapes of them are not spherically symmetric. In addition to the technological and biological importance, modeling the translational and rotational dynamics of the macromolecules is an indispensable step toward the large-scale long-time Brownian dynamics simulation. Before describing the translational-rotational displacement of the many-body system, we will start with how to compute the grand diffusion(mobility) matrix, through which the translational and angular velocities of particles can be obtained. Further the translational and angular velocities can be applied to compute the displacement and orientation of the macromolecules in the many-body system.

Here, the hydrodynamic interactions of the macromolecules which can be viewed as rigid bodies will be studied. Many biological macromolecules and their associations are rigid-like molecules. Such examples can be proteins(such as collagen, spectrin, tubulin, myosin, actin, and keratin), polysaccharides(such as cellulose and xanthan gum), viruses(like Pfl and fd bacteriophages and tobacco mosaic viruses), and short duplex DNAs [37].

The Brownian trajectories followed by a single rigid body is associated by a 6 × 6 *grand diffusion tensor* **D** containing terms related to the translational and rotational diffusivities, where **D** can be calculated from the shape of the rigid particle. The rigid macromolecule in dilute solutions is treated as a particle immersed in a hydrodynamic continuum in many theories for the rotational and translational diffusion coefficients. The computation of the grand diffusion matrix for rigid particles begins with simple, symmetric shapes such as revolution ellipsoids [38, 39, 40] or cylinders [41, 42]. To enable the computation for macromolecules of arbitrary shape, boundary element methods [43, 44, 45] and bead modeling techniques [46, 47, 48, 49, 50, 51, 52] have been widely used. There is pioneering work on rotational Brownian motion using Euler angles [53, 54], oriented rotation angles [55], and other representations [56, 57]. These representations either involve singularities or redundancies, or complex analytical expressions with trigonometric functions. Additionally, most of the prior work about rotational diffusion has the assumption that the grand diffusion matrix is independent of the configuration of the body distribution [53, 54, 58].

From Equation (1.2), the translational and angular velocity of the particle due to the non-hydrodynamic force/torque can be computed as

$$\begin{pmatrix} \mathbf{V} \\ \mathbf{\Omega} \end{pmatrix} = \frac{1}{k_B T}\mathbf{D}\begin{pmatrix} \mathbf{F}^{nh} \\ \mathbf{T}^{nh} \end{pmatrix} = \begin{pmatrix} \mathbf{D}_{tt} & \mathbf{D}_{tr} \\ \mathbf{D}_{rt} & \mathbf{D}_{rr} \end{pmatrix}\begin{pmatrix} \mathbf{F}^{nh} \\ \mathbf{T}^{nh} \end{pmatrix}, \tag{1.9}$$

where $\mathbf{F}^{nh}$ and $\mathbf{T}^{nh}$ are the non-hydrodynamic force and torque. The subscripts t and r mean translation and rotation respectively. $\mathbf{D}_{tt}, \mathbf{D}_{tr} = \mathbf{D}_{rt}$, and $\mathbf{D}_{rr}$ are 3×3 matrix.

For a many-body system in suspension, the associated grand diffusion matrix $\vec{\mathbf{D}}$ describing the hydrodynamic interactions among the macromolecules is a $6m \times 6m$ matrix, where m is the number of the bodies. Finding the grand diffusion matrix is the fundamental problem in constructing the numerical algorithms for tracing the motion of macromolecules in viscous fluids. The bead-shell model along with the mechanism is applied to compute the intermolecular mobility matrix and to build a model for the relationship between the translational and rotational velocities and the general forces/torques exerted on the macromolecules. Actually, in the complex, rigid many-body system, the grand diffusion matrix is $\vec{\mathbf{D}} = (Q\mathbf{D}Q^T)^{-1}$, where $\mathbf{D}$ is the diffusion matrix among all beads of the m bodies. Since the beads are spherically symmetric, the rotations of the beads are not considered. And matrix Q is a transformation matrix with size $6m \times 3N$ and N is the total number of beads for the m-body system. The Rotne-Prager-Yamakawa tensor is employed to describe the matrix $\mathbf{D}$. The grand diffusion matrix $\vec{\mathbf{D}}$ is still symmetric positive definite according to our analysis. Rather than computing a matrix-vector multiplication, readers will see in chapter 4, a linear equation is solved by two fast algorithms to obtain the general velocities including the translational and angular velocities.

Similarly, after using Euler-Maruyama scheme, the m-body translational-rotational displacement can be described by

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\boldsymbol{\varphi} \end{pmatrix} = \frac{\Delta t}{k_B T}\vec{\mathbf{D}} \begin{pmatrix} \mathbf{F}^{nh} \\ \mathbf{T}^{nh} \end{pmatrix} + \sqrt{2\Delta t}\mathbf{B}\mathbf{f} + \nabla \cdot \vec{\mathbf{D}}\Delta t, \tag{1.10}$$

where $\mathbf{x}$ is the vector position of the center of the macromolecules, and $\boldsymbol{\varphi}$ describes their orientations. $\mathbf{F}$ and $\mathbf{T}$ are forces and torques which are evaluated at the beginning of every time step. It is specific to the field or external agent, for example, from polarization. $\vec{\mathbf{D}}$ is the $6m \times 6m$ grand diffusion matrix between the macromolecules, where m is the number of the macromolecules or colloidals. Still, the last term of the right side in Equation (1.10) is zero due to Equation (1.5) for the Rotne-Prager-Yamakawa tensor or the Oseen tensor.

From the Equation (1.10), each translational-rotational displacement can be decomposed into two parts. One part is the displacement due to the non-hydrodynamic force and the other part of displacement is due to the Brownian force. By multiplying the general velocities with Δt, the increase of the displacement for the macromolecules due to non-hydrodynamic forces/torques will be obtained. This displacement is basically the first term of the Brownian dynamics for macromolecules. Moreover, the rigid characteristics of the macromolecules can be used to construct a specific preconditioner to accelerate the convergence rate of the iterative methods for solving the linear equation Equation (1.9).

Different from the diffusion matrix $\mathbf{D}$ for rigid small spherical particles, the grand diffusion matrix of the many-body macromolecules will become $\vec{\mathbf{D}} = (Q\mathbf{D}^{-1}Q^T)^{-1}$ ($\mathbf{D}$ is the diffusion matrix among all beads of the macromolecules). Instead of evaluating a matrix-vector multiplication in the Brownian dynamics model for spherical particles,

$$\begin{pmatrix} \Delta \mathbf{x}^{nh} \\ \Delta \boldsymbol{\varphi}^{nh} \end{pmatrix} = \frac{\Delta t}{k_B T} \vec{\mathbf{D}} \begin{pmatrix} \mathbf{F}^{nh} \\ \mathbf{T}^{nh} \end{pmatrix}, \tag{1.11}$$

the displacements and orientations of the macromolecules due to the systematic force will be obtained through solving a linear equation

$$(Q\mathbf{D}^{-1}Q^T) \begin{pmatrix} \Delta \mathbf{x}^{nh} \\ \Delta \boldsymbol{\varphi}^{nh} \end{pmatrix} = \frac{\Delta t}{k_B T} \begin{pmatrix} \mathbf{F}^{nh} \\ \mathbf{T}^{nh} \end{pmatrix}. \tag{1.12}$$

Also, it will result in corresponding changes for computing the displacement or orientation due to the Brownian force. From these, it shows that the algorithms for simulating the Brownian dynamics with hydrodynamic interactions of the macromolecules are quite different from the ones for spherical particles which are treated as point charges with a small radius.

1.4 Outline of the Dissertation

There are mainly three chapters in this dissertation. The outline of the dissertation is as follows.

In chapter 2, the data structures and the mathematical fundamentals of the new version of the fast multipole method [1] will be summarized firstly. Then how to apply the fast multipole method to compute the field, as well as the electric field gradients will be presented. The electric field gradients are quite important in applications of solid-state physics(ionic crystals), biochemistry, and are also main factors contributing to depolarization of the nerve fiber in MRI scans. Our contribution in this chapter is that the formulas for computing the electric gradient fields using the new version of the fast multipole method [1] have been derived. The computation of field gradients has been added as a function to existing FMM solver. Through utilizing a combination of the local expansion coefficients of the potential, the calculation of the electric field gradients by the new version of the fast multipole method is shown to be efficient with tunable accuracy. The pseudo code for the adaptive algorithm will be presented, as well as the numerical performance for particles with different kinds of distributions.

In chapter 3, the Rotne-Prager-Yamakawa tensor is employed to approximate the mobility of the spherically symmetric particles in fluids. The hydrodynamic forces mediated by the fluids is assumed to act on the center of the particles. A fast algorithm based on the new version of the fast multipole method is proposed for evaluating a matrix-vector multiplication between the Rotne-Prager-Yamakawa tensor and the force vector of all the particles. Efficient evaluation of this matrix-vector multiplication is essential for the simulation of Brownian dynamics with hydrodynamic interactions. The diffusion matrix represented by Rotne-Prager-Yamakawa tensor is configuration-dependent.The vector is the deterministic non-hydrodynamic forces exerted on the particles. The Rotne-Prager-Yamakawa(RPY) solver with tunable accuracy is parallelized on the multicore systems and numerical results have demonstrated its efficiency. The RPY solver we have developed makes it computationally viable for large-scale and long-time Brownian dynamic simulations.

In chapter 4, the hydrodynamic interactions of the macromolecules of arbitrary shape will be considered. The bead-shell model is applied to represent the rigid, complex-structured macromolecules in which the diffusion tensor of the beads is approximated by the Rotne-Prager-Yamakawa tensor. The diffusion tensor of the biological macromolecules of arbitrary shape in stokes fluid of the many-body system has been computed. A concise model has been developed to describe the relationship between the translational and rotational dynamics and the forces or torques of the macromolecules. Two algorithms based on block conjugate gradient method and the Schur complement method are presented in this chapter to solve the model and compute the translational and rotational dynamics in complex systems. A preconditioner has been devised by employing the rigid structure of the macromolecules to accelerate the convergence of the iterative method. Numerical experiments of these two methods have been carried out and the results show that these methods have the advantages in both speed efficiency and memory saving. The displacements and the orientations can be obtained immediately for tracking the trajectories of the macromolecules. The hydrodynamic theory makes it potentially attractive for modeling more complex structured large-scale biomolecular systems and can provide guidelines for computational tools of the bead modeling.

CHAPTER 2

Computation of Electric Field Gradients by Fast Multipole Method

Many applications in computational physics, molecular dynamics, and celestial mechanics require the rapid evaluation of pairwise interactions between all particles. These interactions could be Coulombic or gravitational. It will cause substantial computational cost for evaluating the interactions of large-scale ensembles of particles.

Given a system of N particles, each particle with a charge q_i (or mass m_i) at location $\mathbf{x_i} = (x_i, y_i, z_i)$, the electrostatic(or gravitational) *potential* requires to evaluate

$$\Phi(\mathbf{x_i}) = \sum_{j \neq i} \frac{q_j}{||\mathbf{x}_i - \mathbf{x}_j||}, \tag{2.1}$$

which could be written as a matrix-vector multiplication for N particles. The matrix has zeros on the diagonal and $\{\frac{1}{||\mathbf{x}_i - \mathbf{x}_j||}\}$ on the off diagonals with the corresponding vector described by all charges $\{q_i\}$, $i = 1, \ldots, N$.

The *electrostatic field* is the negative of the derivatives of the potential function and described by the expression

$$E(\mathbf{x_i}) = -\nabla \Phi = \sum_{j \neq i} q_j \frac{\mathbf{x}_i - \mathbf{x_j}}{||\mathbf{x}_i - \mathbf{x}_j||^3}. \tag{2.2}$$

The *Electric Field Gradients* (EFG) is defined as the gradient of the fields, or the negative of the second derivative of the electrostatic potential. It is a ground state property of solids in physics. It is also used to measure the rate of change of the electric field at an atomic nucleus generated by the electronic charge distribution and the other nuclei. The EFG at the nuclear center is caused by the surrounding charge distribution (for example atoms, bonds, electrons, molecular structure, etc). Thus, it is very sensitive to even the most subtle changes in structure, bonding, and dynamics. The

EFG is described by a second-rank tensor

$$\ddot{\mathbf{\Phi}} = \nabla E = - \begin{bmatrix} \Phi_{xx} & \Phi_{xy} & \Phi_{xz} \\ \Phi_{yx} & \Phi_{yy} & \Phi_{yz} \\ \Phi_{zx} & \Phi_{zy} & \Phi_{zz} \end{bmatrix}. \tag{2.3}$$

For the position of each particle $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}) = (x, y, z)$, the EFG of a set of N charges in rectangular coordinates is [59]

$$\ddot{\mathbf{\Phi}}_{ij}(\mathbf{x}) = \sum_{k=1, x_k \neq x}^{N} q_k \left(\frac{\delta_{ij}}{r_k^3} - \frac{3\mathbf{r}_k^{(i)}\mathbf{r}_k^{(j)}}{r_k^5} \right), i, j = 1, 2, 3. \tag{2.4}$$

Here, q_k is the charge of particle k. $\mathbf{r} = \mathbf{x} - \mathbf{x}_k$ is the relative position between the current particle and the particle k. The distance between the current particle and particle k is $r_k = \|\mathbf{r_k}\|_2$. Each of the nine components describes the gradient of the electric field vector components with respect to position. The tensor is traceless because $\frac{1}{r}$ is the kernel of the Laplace equation.

Direct evaluation of (2.1) for N particles will result in $O(N^2)$ operations which will be computationally intensive and time-consuming when N is large. To make large-scale problems tractable, there are many fast summation methods which reduce the complexity of the matrix-vector multiplication (2.1) from $O(N^2)$ to a lower order. The well-known Fast Fourier Transform(FFT) method and the related algorithms are based on a translation invariant and applicable for uniform spatial grids with complexity $O(NlogN)$. Methods like pre-corrected FFT [60, 61], Particle Mesh methods, Particle Mesh Ewald methods[30], and the hierarchical SVD methods [62] all belong to this category.

Another class of the fast summation algorithms is the tree-code algorithms by Appel [63] and Barnes and Hut [64]. The key idea in tree-code algorithms is treating the distant particles as a single large particle centered at the center of clusters and using low order spherical harmonics to compute multipole coefficients that are used to evaluate the potentials from distant particles. The potential from the nearby particles is computed individually and the overall complexity is $O(NlogN)$.

The third category of the fast summation algorithms is the fast multipole method(FMM) which was originally introduced by Greengard and Rokhlin in 1987 [65, 66]. Utilizing the far-field expansions of the Barnes-Hut algorithm, the fast multipole method takes advantage of near-field or local expansions. The influence of a cluster of source points on a cluster of target points could be evaluated in an efficient way by the fast multipole method, while in Barnes-hut algorithm the influence of a cluster of sources is computed on a single target point. With local and multipole expansions, upward and downward passes, the FMM reduces the complexity to $O(N)$ and has been applied to many applications in computational electromagnetics, molecular dynamics, computational fluids and solid mechanics.

Though the fast multipole method [65, 66] is highly successful in two dimensions, the three-dimensional version of the original method has been shown less efficient compared to Barnes-Hut tree algorithms and the FFT-related algorithms due to the huge prefactor in front of $O(N)$. The major obstacle to achieving high efficiency at high accuracy in the original FMM is the cost of the multipole to local translation operator since this translation operator requires $189p^4$ operations per box. In 1997, a new version of FMM was introduced by Greengard and Rokhlin[1] and a scheme has been introduced to reduce the cost of the multipole-to-local expansions by applying exponential translation operators. This scheme utilizes an intermediate "plane-wave" representation to diagonalize the expensive translation operator and apply a "merge-and-shift" technique to reduce the number of the translations. The new FMM is significantly faster than the previous implementation at any desired level of precision, especially for three-dimensional applications and a break-even point of approximately 600 for 6 digits precision is numerically observed.

In this chapter, the data structures and the mathematical fundamentals of the new version of the fast multipole method will be summarized in the first two sections. Next, how to compute the field, as well as the electric field gradients by the fast multipole method will be presented. Through utilizing a combination of the local expansion coefficients of the potential, the calculation is shown to be efficient and with tunable accuracy. The electric gradient fields are quite important in applications of solid-state physics(ionic crystals), biochemistry (electrical polarizabilities of a molecule), and they are also main factors contributing to depolarization of the nerve fiber in MRI scans. Previously, people tried to use perturbation theory [67], finite difference approximation[68, 69], Ewald summation[70], point charge model[59] and other models [71] to approximately evaluate it. However, to our best knowledge, there have been no FMM-related method for efficiently evaluating EFG. Applying FMM to compute EPG will also benefit the fast Rotne-Prager-Yamakawa solver which will be described in Chapter 2. In the last section of this chapter, some numerical results about evaluating potential, field, and electric field gradients using the new version of FMM will be presented.

2.1 Adaptive Tree Structure for the Spatial Domain

In this section, how to construct the adaptive oct-tree data structure of the fast multipole method will be introduced, as well as some related notations. The fast multipole method decomposes the domain space hierarchically, yielding an adaptive oct-tree for three-dimensional cases. In this section, how the oct-tree has been adaptively built according to the distribution of the particles will be described, as well as the corresponding definitions and data structures. Readers can also refer to the original description in [2].

Given a set of N particles distributed randomly in $\mathbb{R}^3$, the computational domain will be defined as the smallest cube which contains all the source particles. This single box corresponding to the entire domain will be viewed as the box at the refinement level 0. Starting from level 0, the computational domain will be partitioned hierarchically with a tree structure until the number of points in each leaf box is less than a prescribed constant s. More specifically, the refinement at level $\ell + 1$ is obtained recursively from partitioning each box at level ℓ into eight cubic boxes with equal size if the number of particles in each box is larger than s.

To begin with, a couple of related definitions for the data structure will be introduced. A box b is said to be the *parent* of box c if box c is obtained through a single subdivision of box b. And box c is called the *child* of box b. The parent box contains more than s particles, while the *childless* box or *leaf* box will have particles less than or equal to s. Boxes at the same level of refinement and sharing at least a boundary point are called *colleagues*. And a box is considered to be a colleague of itself. In three dimension, a box could have 27 colleagues at most.

Mathematically, in [66], two sets $\{\mathbf{x}_i\}_{i=1}^m$ and $\{\mathbf{y}_i\}_{i=1}^n$ are said to be *well separated* if there exists $\mathbf{x}_0, \mathbf{y}_0 \in \mathbb{R}^3$ and a real number $r > 0$ such that

$$\begin{aligned}
|\mathbf{x}_i - \mathbf{x}_0| &< r \text{ for all } i = 1, \cdots, m, \\
|\mathbf{y}_j - \mathbf{y}_0| &< r \text{ for all } j = 1, \cdots, n, and \\
|\mathbf{x}_0 - \mathbf{y}_0| &> cr,
\end{aligned}$$

where $c > c_0$, $c_0 > 2$ is a constant. In the Oct-tree structure, two boxes at the same level of refinement but not colleagues of each other is said to be *well separated*. Or they are well separated if they are at least one box of the same size apart. The well-separated boxes have the low-rank property. Information on the two well-separated boxes can be condensed analytically using some basis functions, such as spherical harmonics for the Laplace kernel.

While traversing down the tree structure, a box c can inherit from its parent box b about the condensed information contributed from boxes that are all well-separated from b. The *interaction list* of a box b is composed of boxes well-separated from b which are also the children of the colleagues of b's parent. In three-dimensional space, the interaction list of each box has 189 boxes at most. Then the information from all well-separated boxes of b are from two parts: one is inherited from b's parent and the other one is from the interaction list of box b. Figure 2.1 shows the interaction list in two dimension. For each box b at a given level, it will associate with five lists of boxes with different sizes. The definitions of these lists are as below:

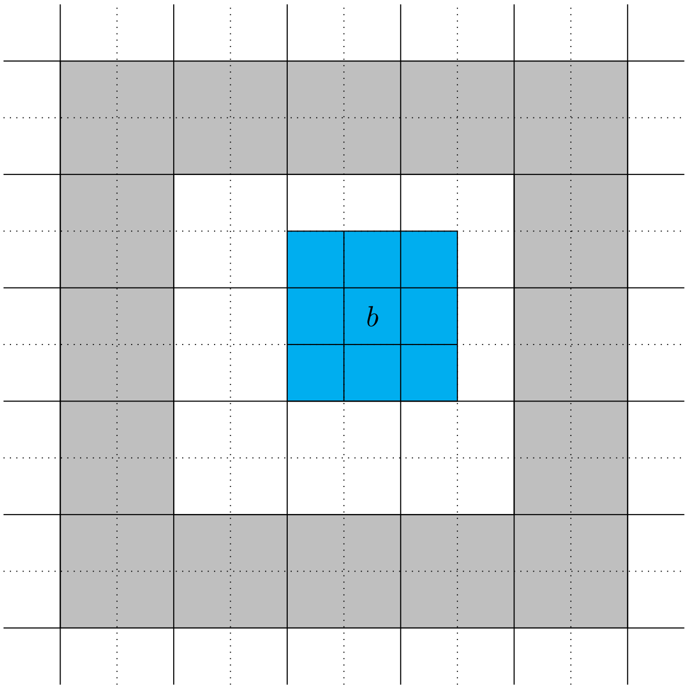

Figure 2.1: Blue boxes are colleagues of box b and white boxes are the interaction list of b.

List 1 of box b denoted by U_b:

- If b is a parent box, U_b is empty;

- If b is a leaf box, U_b is composed of all the childless boxes adjacent to b and box b.

List 2 of box b denoted by V_b:

- It consists of all boxes in the interaction list. i.e. all the children of the colleagues of b's parent that are well separated from b.

List 3 of box b denoted by W_b:

- If b is a parent box, W_b is empty;
- If b is a leaf box, W_b consists of all descendant of b's colleagues whose parents are adjacent to b but not adjacent to b themselves. Each box w in W_b is separated from b by a distance no less than the length of the side of w.

List 4 of box b denoted by X_b

- It is composed of all boxes c such that $b \in W_c$.
- All boxes in X_b are childless and larger than b.

List 5 of box b denoted by Y_b

- It is formed by all boxes that are well-separated from b's parent.

Figure 2.1 shows the associated five lists of box b. To reduce the huge factor in front of $O(N)$ of the original FMM, the new version of FMM introduced plane wave based translation operators to diagonalize the multipole-to-local translations from box b to its interaction list boxes. The interaction list of box b will be partitioned into six lists associated with the six coordinate directions $(+z, -z, +y, -y, +x, -x)$ in three-dimensional space. Corresponding directions and the definitions of the directional lists are as below and Figure 2.1 shows the uplist of b.

- $+z$ direction is referred to as *up*. The *Uplist* for a box b is formed of the boxes in the interaction list which lie above b and are separated by at least one box in the $+z$ direction.
- $-z$ direction is referred to as *down*. The *Downlist* for a box b consists of the boxes in the interaction list which lie below b and are separated by at least one box in the $-z$ direction.

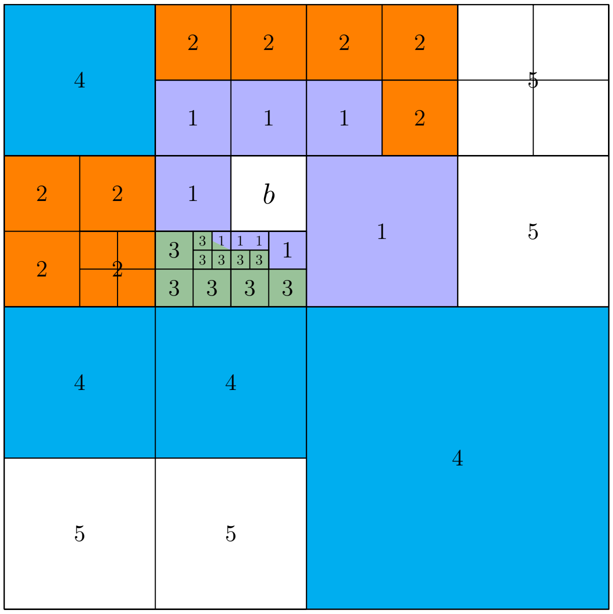

Figure 2.2: Box b and its associated lists in two dimension

- $+y$ direction is referred to as *north*. The *Northlist* for a box b consists of the boxes in the interaction list which lie north of b. They are separated by at least one box in the $+y$ direction, and are not contained in the Up, Down lists.

- $-y$ direction is referred to as *south*. The *Southlist* for a box b consists of the boxes in the interaction list which lie south of b. They are separated by at least one box in the $-y$ direction, and are not contained in the Up, Down lists.

- $+x$ direction is referred to as *east*. The *Eastlist* for a box b consists of the boxes in the interaction list which lie east of b. They are separated by at least one box in the $+x$ direction and are not contained in the Up, Down, South, North lists.

- $-x$ direction is referred to as *west*. The *Westlist* for a box b consists of the boxes in the interaction list which lie south of b. They are separated by at least one box in the $-x$ direction and are not contained in the Up, Down, South, North lists.

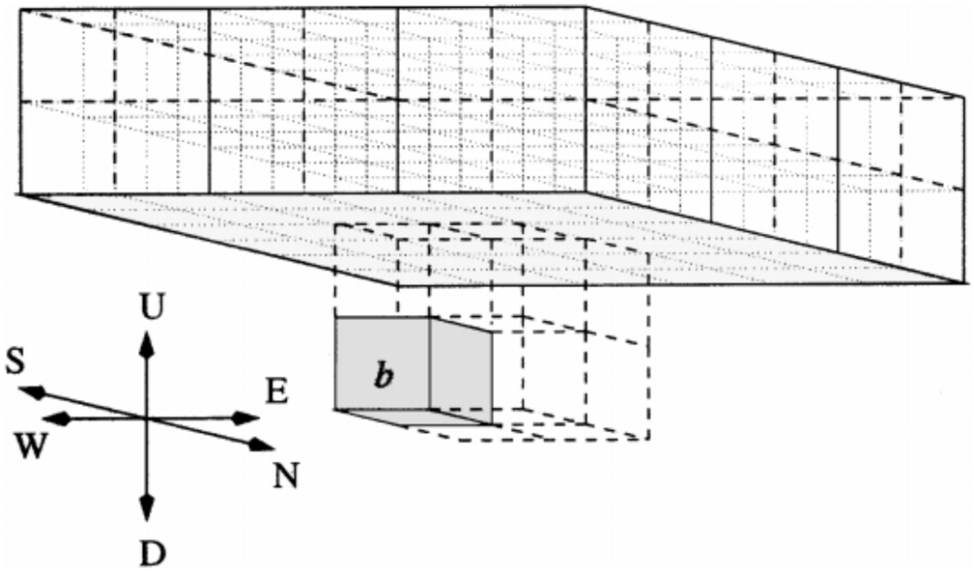

Figure 2.3: The uplist of Box b [2]

2.2 Approximation and Translation

2.2.1 Mathematical Preliminaries

The spherical harmonics $Y_n^m(\theta, \phi)$ are the angular portions of the solution to Laplace equation in spherical coordinates. The definition of spherical harmonics of degree n and order m in [1] is as below;

$$Y_n^m(\theta, \varphi) = \sqrt{\frac{2n+1}{4\pi}} \sqrt{\frac{(n-|m|)}{(n+|m|)}} P_n^{|m|}(\cos\theta) e^{im\varphi}, \forall n \geq 0, |m| \leq n \tag{2.5}$$

Here, P_n^m are the associated Legendre functions which can be defined by the Rodrigues's formula

$$P_n^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_n(x), \tag{2.6}$$

where $P_n(x)$ denotes the Legendre polynomial of degree n and satisfies

$$P_n^{-m} = (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m.$$

All together this gives $Y_n^{-m} = (-1)^m (Y_n^m)^*, \forall n \geq 0, 0 \leq m \leq n$.

A different definition of spherical harmonics in quantum physics is expressed as :

$$\bar{Y}_n^m(\theta, \varphi) = \sqrt{\frac{2n+1}{4\pi}} \sqrt{\frac{(n-m)}{(n+m)}} P_n^m(\cos\theta) e^{im\varphi}, \forall n \geq 0, 0 \leq m \leq n. \tag{2.7}$$

When $0 \leq m \leq n$, there is $\bar{Y}_n^{-m} = (\bar{Y}_n^m)^*$.

For a given value of n, there are $2n+1$ independent solutions of this form, one for each integer m with $-n \leq m \leq n$. The restrictions of n and $|m|$ to non-negative integers with $|m| \leq n$ is a consequence of the requirement that P_n^m should be non-singular at $\cos\theta = \pm 1$.

The Lemma 1 will be used for the computation of the field and the electric field gradient.

Lemma 1. *(Properties of Spherical harmonics) For any $n \geq 0, 0 \leq m \leq n$, there are*

$$\begin{aligned} Y_n^m &= \bar{Y}_n^m \quad &(2.8) \\ Y_n^{-m} &= (-1)^m \bar{Y}_n^{-m}. \quad &(2.9) \end{aligned}$$

Derivatives As in [72], define the sphercal tensor operators

$$\nabla_0 = \partial_z, \nabla_+ = -\frac{1}{\sqrt{2}}(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}), \nabla_- = \frac{1}{\sqrt{2}}(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}) \tag{2.10}$$

there are

$$\begin{aligned} \frac{\partial}{\partial x} &= -\frac{1}{\sqrt{2}}(\nabla_+ - \nabla_-), \quad &(2.11) \\ \frac{\partial}{\partial y} &= \frac{i}{\sqrt{2}}(\nabla_+ + \nabla_-), \quad &(2.12) \\ \frac{\partial}{\partial z} &= \nabla_0. \quad &(2.13) \end{aligned}$$

Lemma 2. *If $f(r)$ is a function about r, and $\bar{Y}_n^m$ is the spherical harmonic function defined by 2.7, $\forall n \geq 1, |m| \leq n$, and define*

$$\alpha(n,m) = \sqrt{\frac{(n+m)(n-m)}{(2n+1)(2n-1)}}, \beta_\pm(n,m) = \sqrt{\frac{(n \mp m - 1)(n \mp m)}{2(2n-1)(2n+1)}}, \tag{2.14}$$

there are

$$\begin{aligned}\nabla_0[f(r)\bar{Y}_n^m] = \alpha(n+1,m)(\frac{df}{dr} - \frac{n}{r}f)\bar{Y}_{n+1}^m \\ + \alpha(n,m)(\frac{df}{dr} + \frac{n+1}{r}f)\bar{Y}_{n-1}^m,\end{aligned} \tag{2.15}$$

$$\begin{aligned}\nabla_\pm[f(r)\bar{Y}_n^m] = \beta_\mp(n+2,m)(\frac{df}{dr} - \frac{n}{r}f)\bar{Y}_{n+1}^{m\pm1} \\ + \beta_\pm(n,m)(\frac{df}{dr} + \frac{n+1}{r}f)\bar{Y}_{n-1}^{m\pm1}.\end{aligned} \tag{2.16}$$

Proof. The details of the proof can be reached at [72]. □

Lemma 3. *(Properties) Suppose for any $n \geq 1, |m| \leq n$, $\alpha(n,m), \beta(n,m)$ are defined as 2.14, there are*

$$\alpha(n,m) = \alpha(n,-m), \tag{2.17}$$

$$\beta_+(n,m) = \beta_-(n,-m). \tag{2.18}$$

Apply Theorem 2 to $f(r) = r^n$, there are

Corollary 1. *If $f(r) = r^n, n \geq 1$, Y_n^m is defined by Equation (2.5), there are*

$$\nabla_0[r^n Y_n^m] = (2n+1)\alpha(n,m)r^{n-1}Y_{n-1}^m, \forall n \geq 1, |m| \leq n, \tag{2.19}$$

$$\nabla_\pm[r^n Y_n^m] = -(2n+1)\beta_\pm(n,m)r^{n-1}Y_{n-1}^{m\pm1}, \forall n \geq 1, |m| \leq n. \tag{2.20}$$

Lemma 4. *Let f, g be real functions whose derivative exists at every point, there is*

$$(\nabla_+)(f+ig) = -[\nabla_-(f-ig)]^*. \tag{2.21}$$

For spherical harmonics defined by Equation (2.5), combine Lemma 3 and Lemma 4, the following theorem for spherical harmonics can be derived.

Theorem 2.2.1. *The corresponding derivatives of spherical harmonics defined by Equation (2.5) are as below:*

$$\nabla_0[r^n Y_n^m] = (2n+1)\alpha(n,m)r^{n-1}Y_{n-1}^m \forall n \geq 0, |m| \leq n; \tag{2.22}$$

$$\nabla_+[r^n Y_n^m] = \begin{cases} -(2n+1)\beta_+(n,m) r^{n-1} Y_{n-1}^{m+1} & \forall n \geq 1, 0 \leq m \leq n; \\ (2n+1)\beta_+(n,m) r^{n-1} Y_{n-1}^{m+1} & \forall n \geq 1, -n \leq m \leq -1; \end{cases} \tag{2.23}$$

$$\nabla_-[r^n Y_n^m] = \begin{cases} -(2n+1)\beta_-(n,m) r^{n-1} Y_{n-1}^{m-1} & \forall n \geq 0, 1 \leq m \leq n; \\ (2n+1)\beta_-(n,m) r^{n-1} Y_{n-1}^{m-1} & \forall n \geq 0, -n \leq m \leq 0. \end{cases} \tag{2.24}$$

2.2.2 Approximation Operators

All hierarchical N-body algorithms are based on the idea of evaluating combined effect of a set of distant particles instead of treating them individually. In this chapter, the Laplace kernel will be used as an example to present how the fast multipole methods extract, condense and transmit information on the adaptive Oct-tree structures. Two expansions operators including the multipole expansion and the local expansion, are fundamental in FMM. The multipole expansions allow one to group a cluster of particles that lie close together and treat them as if they are a single source. The local expansions transfer information from long range to near range. The proof of the theorems will be neglected. Interested readers can refer to [2] [66] for details.

Theorem 2.2.2 (Multipole Expansion)**.** *Suppose N particles with charges $\{q_i\}_{i=1}^N$ and positions $\{\mathbf{x}_i = (\rho_i, \alpha_i, \beta_i)\}_{i=1}^N$ are located in a box centered at the origin. Then for any point $\mathbf{x} = (r, \theta, \phi) \in \mathbb{R}^3$ outside the box, the potential $\Phi(\mathbf{x})$ is given by*

$$\Phi(\mathbf{x}) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{M_n^m}{r^{n+1}} \cdot Y_n^m(\theta, \phi), \tag{2.25}$$

where the multipole expansion coefficients M_n^m are given by

$$M_n^m = \sum_{i=1}^{N} q_i \cdot \rho_i^n \cdot Y_n^{-m}(\alpha_i, \beta_i). \tag{2.26}$$

Furthermore, for any $p \geq 1$,

$$\left| \Phi(\mathbf{x}) - \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{M_n^m}{r^{n+1}} \cdot Y_n^m(\theta, \phi) \right| \leq \left(\frac{\sum_{i=1}^N |q_i|}{r - a} \right) \left(\frac{a}{r} \right)^{p+1}.$$

where a is the radius of the smallest sphere enclosing the box.

The above theorem describes an efficient representation of the far field potential due to a collection of sources. $\mathcal{T}_{SM}$ means source-to-multipole operator and $\mathcal{T}_{LT}$ is the local-to-target operator.

Theorem 2.2.3 (Local Expansion)**.** *Suppose N particles with charges $\{q_i\}_{i=1}^N$ and positions $\{\mathbf{x}_i = (\rho_i, \alpha_i, \beta_i)\}_{i=1}^N$ are located outside the sphere S_a of radius a centered at the origin. Then for any point with coordinates $\mathbf{x} = (r, \theta, \phi) \in S_a$ outside the box, the potential $\Phi(\mathbf{x})$ generated by the charges $q_1, q_2, \dots, q_N$ is described by the local expansion*

$$\Phi(\mathbf{x}) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j, \tag{2.27}$$

where the local expansion coefficients L_j^k are given by

$$L_j^k = \sum_{l=1}^{N} q_l \cdot \frac{Y_j^{-k}(\alpha_l, \beta_l)}{\rho_l^{j+1}}. \tag{2.28}$$

Furthermore, for any $p \geq 1$,

$$\left| \Phi(\mathbf{x}) - \sum_{j=0}^{p} \sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^{j+1} \right| \leq \left(\frac{\sum_{i=1}^{N} |q_i|}{a - r} \right) \left(\frac{r}{a} \right)^{p+1}.$$

2.2.3 Translation Operators

To form multipole expansions for all level of boxes, the FMM applies a divide-and-conquer strategy to collect the compressed information through merging and shifting its children's multipole expansions for each parent box level by level. The following multipole-to-multipole translation operator $\mathcal{T}_{MM}$ is applied in an upward pass for forming all the multipole expansions.

Theorem 2.2.4 (Translation of a Multipole Expansion $\mathcal{T}_{MM}$)**.** *Suppose N particles with charges $\{q_i\}_{i=1}^N$ and positions $\{\mathbf{x}_i = (\rho_i, \alpha_i, \beta_i)\}_{i=1}^N$ are located in a sphere D centered at $\mathbf{x}_0 = (\rho, \alpha, \beta)$. Then for any point $\mathbf{x} = (r, \theta, \phi) \in \mathbb{R}^3 \setminus D$, the potential generated by the charges $\{q_i\}_{i=1}^N$ is given by*

$$\Phi(\mathbf{x}) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{O_n^m}{r'^{n+1}} \cdot Y_n^m(\theta', \phi'), \tag{2.29}$$

where (r', θ', ϕ') *are the spherical coordinates of the vector* $\mathbf{x} - \mathbf{x_0}$.

Then for any point $\mathbf{x} = (\rho, \theta, \phi)$ *outside a sphere* D_1 *with radius* $a + \rho$ *centered at the origin,*

$$\Phi(\mathbf{x}) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} \frac{M_j^k}{r^{j+1}} \cdot Y_j^k(\theta, \phi),$$

where

$$M_j^k = \sum_{n=0}^{j} \sum_{m=-n}^{n} \frac{O_{j-n}^{k-m} \cdot i^{|k|-|m|-|k-m|} \cdot A_n^m \cdot A_{j-n}^{k-m} \cdot \rho^n \cdot Y_n^{-m}(\alpha, \beta)}{A_j^k}, \tag{2.30}$$

with A_n^m *defined by the formula*

$$A_n^m = \frac{(-1)^n}{\sqrt{(n-m)! \cdot (n+m)!}}. \tag{2.31}$$

Furthermore for any $p \geq 1$,

$$\left| \Phi(\mathbf{x}) - \sum_{j=0}^{p} \sum_{k=-j}^{j} \frac{M_j^k}{r^{j+1}} \cdot Y_j^k(\theta, \phi) \right| \leq \left(\frac{\sum_{i=1}^{N} |q_i|}{r - (a+\rho)} \right) \left(\frac{a+\rho}{r} \right)^{p+1}.$$

The linear operator defined by Equation (2.29) converts the old multipole expansion coefficients O_j^k into the new multipole expansion M_j^k and will denoted by $\mathcal{T}_{MM}$. The above theorem can be used to obtain all multipole expansions from the finest level of boxes to the coarsest level in the upward pass.

In the downward pass from the coarsest refinement level to the finest level, local-to-local translation operator $\mathcal{T}_{LL}$ is carried out firstly to shift the parent box's local expansion which contains the far-field particle contributions into the center of the children's boxes.

Theorem 2.2.5 (Translation of a Local expansion $\mathcal{T}_{LL}$). *Consider a point* $\mathbf{x} = (\rho, \theta, \phi)$ *in a box centered at* $\mathbf{x}_0 = (\rho, \alpha, \beta)$. *The spherical coordinates of the vector* $\mathbf{x} - \mathbf{x_0}$ *is* (ρ', θ', ϕ'). *The pth order local expansion of its parent box is described by*

$$\Phi(\mathbf{x}) = \sum_{n=0}^{p} \sum_{m=-n}^{n} O_n^m \cdot Y_n^m(\theta', \phi') \cdot r'^n. \tag{2.32}$$

Then the local expansion of that box is

$$\Phi(\mathbf{x}) = \sum_{j=0}^{p} \sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j, \tag{2.33}$$

where

$$L_j^k = \sum_{n=j}^{p} \sum_{m=-n}^{n} \frac{O_n^m \cdot i^{|m|-|m-k|-|k|} \cdot A_{n-j}^{m-k} \cdot A_j^k \cdot Y_{n-j}^{m-k}(\alpha, \beta) \cdot \rho^{n-j}}{(-1)^{n+j} A_n^m}, \tag{2.34}$$

with A_n^m defined by (2.31).

After the local-to-local expansions in the downward pass, a box c has inherited information from its parent box b. The second step for box c is trying to collect information from its interaction list. Since all boxes in the interaction list are well-separated from c, a multipole-to-local expansion operator $\mathcal{T}_{ML}$ is applied to approximate the contribution from the interaction list of c instead of communicating with each particle in this region.

Theorem 2.2.6 (Conversion of a Multipole Expansion into a Local expansion $\mathcal{T}_{ML}$)**.** *Suppose ℓ charges of strengths $\{q_i\}_{i=1}^{\ell}$ are located inside a sphere $D_{\mathbf{x_0}}$ centered at $\mathbf{x_0} = (\rho, \alpha, \beta)$ with radius a, and $\rho > (c+1)a$ for some constant $c > 1$. Then the corresponding multipole expansion given by (2.25) converges inside another sphere D_0 with radius a centered at the origin. For any point $\mathbf{x} \in D_0$ with spherical coordinates (r, θ, ϕ), its potential due to charges inside $D_{\mathbf{x_0}}$ is described by*

$$\Phi(\mathbf{x}) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j, \tag{2.35}$$

where

$$L_j^k = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{O_n^m \cdot i^{|k-m|-|k|-|m|} \cdot A_n^m \cdot A_j^k \cdot Y_{n+j}^{m-k}(\alpha, \beta)}{(-1)^n A_{n+j}^{m-k} \cdot \rho^{j+n+1}}. \tag{2.36}$$

Furthermore, for any $p \geq 1$,

$$\left| \Phi(\mathbf{x}) - \sum_{j=0}^{p} \sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^{j+1} \right| \leq \left(\frac{\sum_{i=1}^{N} |q_i|}{ca - a} \right) \left(\frac{1}{c} \right)^{p+1}. \tag{2.37}$$

The last step for the downward pass is to evaluate the local expansion at each particle of all leaf boxes and combine the near-field interactions.

In all of the above expansions, the multipole-to-local expansion is the bottleneck and the complexity is $O(p^4)$ if the expansions are approximated with p^2 terms for each box. In three dimensions, a given box could have up to 189 boxes in its interaction list. As a result, in the original FMM method, the multipole-to-local expansion for a given box could cost $189p^4$ operations which makes it not practical especially for large-scale problems.

To make the original fast multipole method more promising, a new version of FMM [1] based on a new diagonal form for translation operators was introduced to produce high accuracy at an acceptable computation cost. Interested readers can also refer to [73],[74], and [75] for other version translation operators. Here, the plane-wave expansion with exponential basis [2] will be presented to diagonalize the $\mathcal{T}_{ML}$ operator. In three dimension, six plane-wave expansions for direction $(+z, -z, +y, -y, +x, -x)$ are introduced and the multipole-to-local translator has been decomposed into $\mathcal{T}_{ML} = \mathcal{T}_{EL} \circ \mathcal{T}_{EE} \circ \mathcal{T}_{ME}$, where $\mathcal{T}_{ME}$ is the multipole-to-exponential operator, $\mathcal{T}_{EE}$ represented the exponential-to-exponential operator, and $\mathcal{T}_{EL}$ denoted as the exponential-to-local operator. This decomposition will reduce the cost from $189p^4$ to $6p^3 + 189p^2$ for each box. If the merge and shift strategy is applied, the cost for multipole to local expansion will be $6p^3 + 40p^2$ and this result could be further improved to $15p^2 logp + 40p^2$ by applying FFT.

In the new version of FMM, the interaction list of a given box c has been subdivided into Uplist, Downlist, Eastlist, Westlist, Northlist, and Southlist. These six lists are corresponding to directions $(+z, -z, +y, -y, +x, -x)$. For any box c in the six interaction lists of b, the multipole expansion centered in b will be firstly translated into an exponential expansion about a center of a box, then shift it using exponential-to-exponential expansion to a new center of an interaction list box. Once the exponential expansions from the interaction list boxes have been collected, the exponential expansions will be translated to a local expansion in box c using exponential-to-local operator.

The translations of upward $(+z)$ direction are illustrated here and translations for the other directions can be processed similarly.

Theorem 2.2.7 (Multipole Expansion to Exponential expansion $\mathcal{T}_{ME}$)**.** *Suppose box b containing N charges $\{q_i, i = 1, \ldots, N\}$ is centered at the origin and with unit volume. The charges are located at points $\{\mathbf{x}_i = (x_i, y_i, z_i)\}$. Let x be the location of a particle in box c and $c \in$ Uplist(b). Given the* *multipole expansion of* $\Phi(\mathbf{x})$ *as*

$$\Phi(\mathbf{x}) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{M_n^m}{r^{n+1}} \cdot Y_n^m(\theta, \phi). \tag{2.38}$$

Then

$$\left| \Phi(\mathbf{x}) - \sum_{k=1}^{s(\epsilon)} \sum_{j=1}^{M(k)} W(k,j) e^{-\lambda_k z} e^{i\lambda_k (x \cos\alpha_j + y \sin\alpha_j)} \right| < A\epsilon, \tag{2.39}$$

and

$$W(k,j) = \frac{\omega_k}{M(k)} \sum_{m=-\infty}^{\infty} (-i)^{|m|} e^{ima_j} \sum_{n=|m|}^{\infty} \frac{M_n^m}{\sqrt{(n-m)!(n+m)!}} \lambda_k^n, \tag{2.40}$$

where $A = \sum_{i=1}^{N} |q_i|$, $\{\lambda_k, \omega_k\}$ *are the quadrature nodes and weights returned by the generalized Gaussian quadrature.* $s(\epsilon)$ *denotes the number of* λ_k*'s for accuracy requirement* ϵ. *For each* λ_k, $M(k)$ *denotes the number of quadrature needed for the trapezoid rule. And* $\alpha_j = \frac{2\pi j}{M_k}$.

$\mathcal{T}_{ME}$ maps the multipole expansion coefficients $\{M_n^m)\}, n = 0, \ldots, p, m = -n, \ldots, n$ to the corresponding exponential expansion coefficients $\{W(k,j)\}$.

Theorem 2.2.8 (Exponential to Exponential expansion $\mathcal{T}_{EE}$)**.** *Suppose box* b *containing* N *charges* $\{q_i, i = 1, \ldots, N\}$ *is centered at the origin and with unit volume. The charges are located at points* $\{\mathbf{x}_i = (x_i, y_i, z_i)\}$. *Let* x *be the location of a particle in box* c. c *is a box in Uplist(b) and centered at* (x_1, y_1, z_1). *Given the exponential expansion centered at the origin of* $\Phi(\mathbf{x})$ *as*

$$\Phi(\mathbf{x}) = \sum_{k=1}^{s(\epsilon)} \sum_{j=1}^{M(k)} W(k,j) e^{-\lambda_k z} e^{i\lambda_k (x \cos\alpha_j + y \sin\alpha_j)} + O(\epsilon), \tag{2.41}$$

then

$$\Phi(\mathbf{x}) = \sum_{k=1}^{s(\epsilon)} \sum_{j=1}^{M(k)} V(k,j) e^{-\lambda_k (z - z_1)} e^{i\lambda_k ((x-x_1) \cos\alpha_j + (y-y_1) \sin\alpha_j)} + O(\epsilon), \tag{2.42}$$

where

$$V(k,j) = W(k,j) e^{-\lambda_k z_1} e^{i\lambda_k (x_1 \cos\alpha_j + y_1 \sin\alpha_j)}. \tag{2.43}$$

The exponential-to-exponential operator maps the original set of exponential expansion coefficients $\{W(k,j)\}$ to the shifted exponential expansion coefficients $\{V(k,j)\}$, where (x_1, y_1, z_1) is the vector from the center of box b to the center of box c.

Theorem 2.2.9 (Exponential expansion to Local expansion $\mathcal{T}_{EL}$). *Suppose box b containing N charges $\{q_i, i = 1, \ldots, N\}$ is centered at the origin and with unit volume. The charges are located at points $\{\mathbf{x}_i = (x_i, y_i, z_i)\}$. Let x be the location of a particle in box c and $c \in$ UPlist(b). Given the exponential expansion of $\Phi(\mathbf{x})$ as*

$$\left| \Phi(\mathbf{x}) - \sum_{k=1}^{s(\epsilon)} \sum_{j=1}^{M(k)} W(k,j) e^{-\lambda_k z} e^{i\lambda_k (x\cos\alpha_j + y \sin\alpha_j)} \right| < A\epsilon, \tag{2.44}$$

where $A = \sum_{i=1}^{N} |q_i|$, $\{\lambda_k, \omega_k\}$. Then

$$\left| \Phi(\mathbf{x}) - \sum_{n=0}^{\infty} \sum_{m=-n}^{n} L_n^m \cdot Y_n^m(\theta, \phi) \cdot r^n \right| < A\epsilon, \tag{2.45}$$

where

$$L_n^m = \frac{(-i)^{|m|}}{\sqrt{(n-m)!(n+m)!}} \sum_{k=1}^{s(\epsilon)} (-\lambda_k)^n \sum_{j=1}^{M(k)} W(k,j) e^{im\alpha_j}. \tag{2.46}$$

Translation operator $\mathcal{T}_{EL}$ maps the shifted set of exponential expansion coefficients $\{W(k,j)\}$ to the corresponding truncated harmonic expansion coefficients $\{L_n^m\}, n = 0, \ldots, p, m = -n, \ldots, n$.

2.3 Local Expansions of the Fields and Electric Field Gradients

This section will start with how to use the local expansion coefficients of the far-field potential to derive the far-field part of the fields and the electric field gradients. Next the function of computing the electric field gradients using FMM is added for existing FMM solver. The near-field calculations will be directly evaluated and the details will be omitted here.

2.3.1 Electrostatic Fields

Lemma 5. *Consider a point $\mathbf{x} = (r, \theta, \phi)$ outside a box centered at the origin. There are N particles with charges $\{q_j\}_{j=1}^N$ and locations $\{\mathbf{x}_j = (\rho_j, \alpha_j, \beta_j)\}_{j=1}^N$ in the box, the pth order local expansion of the potential Φ at $\mathbf{x}$ is described by*

$$\Phi(\mathbf{x}) = \sum_{\ell=0}^{p} \sum_{m=-\ell}^{\ell} L_\ell^m \cdot Y_\ell^m(\theta, \phi) \cdot r^\ell. \tag{2.47}$$

Here, L_ℓ^m is the local expansion coefficients given by

$$L_\ell^m = \sum_{j=1}^{N} q_j \cdot \frac{Y_\ell^{-m}(\alpha_j, \beta_j)}{\rho_j^{\ell+1}}. \tag{2.48}$$

So $L_\ell^{-m} = (L_\ell^m)^*$. Define the notations below for all $0 \leq \ell \leq p-1, |m| \leq \ell$,

$$\begin{aligned} D_0(\ell, m) &= (2\ell+3)\alpha(\ell+1, m) L_{\ell+1}^m Y_\ell^m r^\ell, \\ D_p(\ell, m) &= (2\ell+3)\beta_+(\ell+1, m-1) L_{\ell+1}^{m-1} Y_l^m r^\ell, \\ D_m(\ell, m) &= (2\ell+3)\beta_-(\ell+1, m+1) L_{\ell+1}^{m+1} Y_l^m r^\ell, \end{aligned} \tag{2.49}$$

the pth order local expansion of the potential Φ under spherical tensor operators at $\mathbf{x}$ is described by

$$\nabla_0(\phi) = \sum_{\ell=0}^{p-1} D_0(\ell, 0) + \sum_{\ell=1}^{p-1} \sum_{m=1}^{\ell} 2\Re(D_0(\ell, m)), \tag{2.50}$$

$$\nabla_+[\phi] = \sum_{\ell=0}^{p-1} \left[\sum_{m=-\ell}^{0} - \sum_{m=1}^{\ell} \right] D_p(\ell, m), \tag{2.51}$$

$$\nabla_-[\phi] = \sum_{\ell=0}^{p-1} \left[\sum_{m=-\ell}^{-1} - \sum_{m=0}^{\ell} \right] D_m(\ell, m). \tag{2.52}$$

Proof.

$$\begin{aligned} \nabla_0(\phi) &= \sum_{\ell=0}^{p} \sum_{m=-\ell}^{\ell} L_\ell^m \nabla_0(r^\ell Y_\ell^m) \\ &= \sum_{\ell=1}^{p} \sum_{m=-(\ell-1)}^{\ell-1} L_\ell^m \alpha(\ell, m)(2\ell+1) r^{\ell-1} Y_{\ell-1}^m. \Leftarrow 0 \leq |m| \leq \ell - 1 \end{aligned} \tag{2.53}$$

Let $\ell' = \ell - 1$ and change back the index from ℓ' to ℓ.

$$\begin{aligned} \nabla_0(\phi) &= \sum_{\ell=0}^{p-1} \sum_{m=-\ell}^{\ell} L_{\ell+1}^m \alpha(\ell+1, m)(2\ell+3) r^\ell Y_\ell^m \\ &= \sum_{\ell=0}^{p-1} (2\ell+3)\alpha(\ell+1, 0) L_{\ell+1}^0 Y_\ell^0 r^\ell \\ &+ \sum_{\ell=1}^{p-1} \sum_{m=1}^{\ell} 2(2\ell+3)\alpha(\ell+1, m) \Re(L_{\ell+1}^m Y_\ell^m) r^\ell. \end{aligned} \tag{2.54}$$

$$
\begin{aligned}
\nabla_+(\phi) &= \sum_{\ell=0}^{p}\sum_{m=-\ell}^{\ell} L_\ell^m \nabla_+(r^\ell Y_\ell^m) \\
&= \sum_{\ell=1}^{p}\sum_{m=-(\ell-1)}^{-1} L_\ell^m \beta_+(\ell,m)(2\ell+1)r^{\ell-1}Y_{\ell-1}^{m+1} \\
&+ \sum_{\ell=1}^{p}\sum_{m=0}^{\ell-1} L_\ell^m(-\beta_+(\ell,m))(2\ell+1)r^{\ell-1}Y_{\ell-1}^{m+1}.
\end{aligned} \tag{2.55}
$$

Let $\ell' = \ell - 1, m' = m + 1$, and change back index to ℓ, m, the theorem can be derived easily. Similar procedure could be applied to $\nabla_-(\phi)$. □

Theorem 2.3.1. *[Electronic Fields of the far-field Part] The pth order local expansion of electrostatic fields* $(-\frac{\partial\phi}{\partial x}, -\frac{\partial\phi}{\partial y}, -\frac{\partial\phi}{\partial z})$ *due to the charges located in boxes well separated from box b can be evaluated as below*

$$\frac{\partial\phi}{\partial x} = \sqrt{2}\left[\sum_{\ell=1}^{p-1}\sum_{m=1}^{\ell}\Re(D_p(\ell,m)) - \sum_{\ell=0}^{p-1}\sum_{m=0}^{\ell}\Re(D_m(\ell,m)\right], \tag{2.56}$$

$$\frac{\partial\phi}{\partial y} = \sqrt{2}\left[\sum_{\ell=1}^{p-1}\sum_{m=1}^{\ell}\Im(D_p(\ell,m)) - \sum_{\ell=0}^{p-1}\sum_{m=0}^{\ell}\Im(D_m(\ell,m)\right], \tag{2.57}$$

$$\frac{\partial\phi}{\partial z} = \sum_{\ell=0}^{p-1} D_0(\ell,0) + \sum_{\ell=1}^{p-1}\sum_{m=1}^{\ell} 2\Re(D_0(\ell,m)). \tag{2.58}$$

Proof. Note that

$$D_p(\ell,-m) = D_m(\ell,m)^*, \tag{2.59}$$

$$D_0(\ell,-m) = D_0(\ell,m)^*. \tag{2.60}$$

According to Lemma 5, the *pth* order of local expansions of the partial derivatives of potential ϕ can be derived from

$$
\begin{aligned}
\frac{\partial\phi}{\partial x} &= -\frac{1}{\sqrt{2}}\sum_{\ell=0}^{p-1}\left[\sum_{m=0}^{\ell}(D_p(\ell,-m)+D_m(\ell,m)) - \sum_{m=1}^{\ell}(D_p(\ell,m)+D_m(\ell,-m))\right] \\
&= \frac{1}{\sqrt{2}}\sum_{\ell=0}^{p-1}\left[-\sum_{m=0}^{\ell}2\Re(D_m(\ell,m)) + \sum_{m=1}^{\ell}2\Re(D_p(\ell,m))\right].
\end{aligned} \tag{2.61}
$$

Follow the same procedure to get the proof of Equation (2.57) and Equation (2.58). □

2.3.2 The far-field Electric Field Gradients

Corollary 2. *For* $\forall \ell \geq 1, |m| \leq \ell$, *there are*

$$\begin{aligned}
\nabla_+^2[r^l Y_\ell^m] &= (2l-1)(2l+1)\beta_+(\ell,m)\beta_+(\ell-1,m+1)r^{\ell-2}Y_{\ell-2}^{m+2}, \\
\nabla_-^2[r^l Y_\ell^m] &= (2l-1)(2l+1)\beta_-(\ell,m)\beta_-(\ell-1,m-1)r^{\ell-2}Y_{\ell-2}^{m-2}, \\
\nabla_0^2[r^l Y_\ell^m] &= (2l-1)(2l+1)\alpha(\ell,m)\alpha(\ell-1,m)r^{\ell-2}Y_{\ell-2}^{m}, \\
\nabla_+\nabla_-[r^l Y_\ell^m] &= (2l-1)(2l+1)\beta_+(\ell,m)\beta_-(\ell-1,m+1)r^{\ell-2}Y_{\ell-2}^{m}, \\
\nabla_+\nabla_0[r^l Y_\ell^m] &= -(2l-1)(2l+1)\alpha(\ell,m)\beta_+(\ell-1,m)r^{\ell-2}Y_{\ell-2}^{m+1}, \\
\nabla_-\nabla_0[r^l Y_\ell^m] &= -(2l-1)(2l+1)\alpha(\ell,m)\beta_-(\ell-1,m)r^{\ell-2}Y_{\ell-2}^{m-1}. \qquad (2.62)
\end{aligned}$$

Proof. Apply Theorem 2.2.1 repeatedly to get the conclusions. □

Theorem 2.3.2. *Consider a point* $\mathbf{x} = (r, \theta, \phi)$ *outside a box centered at the origin. There are* N *particles with charges* $\{q_j\}_{j=1}^N$ *and locations* $\{\mathbf{x}_j = (\rho_j, \alpha_j, \beta_j)\}_{j=1}^N$ *in the box, the pth order local expansion of the potential* ϕ *at* $\mathbf{x}$ *is described by*

$$\Phi(\mathbf{x}) = \sum_{\ell=0}^{p} \sum_{m=-\ell}^{\ell} L_\ell^m \cdot Y_\ell^m(\theta, \phi) \cdot r^\ell. \qquad (2.63)$$

Here, L_ℓ^m *is the local expansion coefficients given by*

$$L_\ell^m = \sum_{j=1}^{N} q_j \cdot \frac{Y_\ell^{-m}(\alpha_j, \beta_j)}{\rho_j^{\ell+1}}. \qquad (2.64)$$

Define the notations below:

$$
\begin{aligned}
D_{pp}(\ell,m) &= (2\ell+3)(2\ell+5)\beta_+(\ell+2,m-2)\beta_+(\ell+1,m-1)r^\ell L_{\ell+2}^{m-2}Y_\ell^m,\\
D_{mm}(\ell,m) &= (2\ell+3)(2\ell+5)\beta_-(\ell+2,m+2)\beta_-(\ell+1,m+1)r^\ell L_{\ell+2}^{m+2}Y_\ell^m,\\
D_{pm}(\ell,m) &= (2\ell+3)(2\ell+5)\beta_+(\ell+2,m)\beta_-(\ell+1,m+1)r^\ell L_{\ell+2}^{m}Y_\ell^m,\\
D_{mp}(\ell,m) &= (2\ell+3)(2\ell+5)\beta_-(\ell+2,m)\beta_+(\ell+1,m-1)r^\ell L_{\ell+2}^{m}Y_\ell^m,\\
D_{00}(\ell,m) &= (2\ell+3)(2\ell+5)\alpha(\ell+2,m)\alpha(\ell+1,m)r^\ell L_{\ell+2}^{m}Y_\ell^m,\\
D_{p0}(\ell,m) &= (2\ell+3)(2\ell+5)\alpha(\ell+2,m-1)\beta_+(\ell+1,m-1)r^\ell L_{\ell+2}^{m-1}Y_\ell^m,\\
D_{m0}(\ell,m) &= (2\ell+3)(2\ell+5)\alpha(\ell+2,m+1)\beta_-(\ell+1,m+1)r^\ell L_{\ell+2}^{m+1}Y_\ell^m.
\end{aligned}
$$

Then the pth order local expansions of the second order derivatives of the potential ϕ at $\mathbf{x}$ is

$$
\begin{aligned}
\frac{\partial^2\phi}{\partial x^2} &= \sum_{\ell=0}^{p-2}\left[\sum_{m=2}^{\ell}\Re(D_{pp}(\ell,m))+\sum_{m=0}^{\ell}\Re(D_{mm}(\ell,m))-\Re(D_{pp}(\ell,1))\right]\\
&\quad-\sum_{\ell=0}^{p-2}\left[\sum_{m=0}^{\ell}D_{pm}(\ell,m)+\sum_{m=1}^{\ell}D_{mp}(\ell,m)\right],
\end{aligned}
\tag{2.65}
$$

$$
\begin{aligned}
\frac{\partial^2\phi}{\partial y^2} &= -\sum_{\ell=0}^{p-2}\left[\sum_{m=2}^{\ell}\Re(D_{pp}(\ell,m))+\sum_{m=0}^{\ell}\Re(D_{mm}(\ell,m))-\Re(D_{pp}(\ell,1))\right]\\
&\quad-\sum_{\ell=0}^{p-2}\left[\sum_{m=0}^{\ell}D_{pm}(\ell,m)+\sum_{m=1}^{\ell}D_{mp}(\ell,m)\right],
\end{aligned}
\tag{2.66}
$$

$$
\frac{\partial^2\phi}{\partial z^2} = \sum_{\ell=0}^{p-2}\left[\sum_{m=1}^{\ell}2\Re(D_{00}(\ell,m))+D_{00}(\ell,0)\right],
\tag{2.67}
$$

$$
\frac{\partial^2\phi}{\partial x\partial y} = \sum_{\ell=0}^{p-2}\left[\sum_{m=2}^{\ell}\Im(D_{pp}(\ell,m))-\sum_{m=0}^{\ell}\Im(D_{mm}(\ell,m))-\Im(D_{pp}(\ell,1))\right],
\tag{2.68}
$$

$$
\frac{\partial^2\phi}{\partial x\partial z} = \sqrt{2}\sum_{\ell=0}^{p-2}\left[\sum_{m=1}^{\ell}\Re(D_{p0}(\ell,m))-\sum_{m=0}^{\ell}\Re(D_{m0}(\ell,m))\right],
\tag{2.69}
$$

$$
\frac{\partial^2\phi}{\partial y\partial z} = \sqrt{2}\sum_{\ell=0}^{p-2}\left[\sum_{m=1}^{\ell}\Im(D_{p0}(\ell,m))-\sum_{m=0}^{\ell}\Im(D_{m0}(\ell,m))\right].
\tag{2.70}
$$

Proof. First, note that

$$\begin{aligned} D_{pp}(\ell,m) &= D_{mm}(\ell,-m)^*, & (2.71)\\ D_{pm}(\ell,m) &= D_{mp}(\ell,-m)^*, & (2.72)\\ D_{p0}(\ell,m) &= D_{m0}(\ell,-m)^*, & (2.73) \end{aligned}$$

So

$$\begin{aligned} D_{pp}(\ell,m)+D_{mm}(\ell,-m) &= 2\Re(D_{pp}(\ell,m)),\\ D_{pp}(\ell,m)-D_{mm}(\ell,-m) &= 2i\Im(D_{pp}(\ell,m)),\\ D_{p0}(\ell,m)+D_{m0}(\ell,-m) &= 2\Re(D_{p0}(\ell,m)),\\ D_{p0}(\ell,m)-D_{m0}(\ell,-m) &= 2i\Im(D_{p0}(\ell,m)). \end{aligned}$$

Since $|m+2| \leq |\ell-2|$, there are $-\ell \leq m \leq \ell-4$ and the *pth* order local expansion of the potential ϕ under opterator ∇_+^2 at $\mathbf{x}$ is described by

$$\begin{aligned} \nabla_+^2[\phi] &= \sum_{\ell=0}^{p}\sum_{m=-\ell}^{\ell} L_\ell^m \nabla_+^2\left[r^\ell Y_\ell^m\right]\\ &= \sum_{\ell=2}^{p}[(\sum_{m=0}^{\ell-4}+\sum_{m=-\ell}^{-2})(2\ell+1)(2\ell-1)L_\ell^m\beta_+(\ell,m)\beta_+(\ell-1,m+1)r^{\ell-2}Y_{\ell-2}^{m+2}\\ &\quad -(2\ell+1)(2\ell-1)L_\ell^{-1}\beta_+(\ell,-1)\beta_+(\ell-1,0)r^{\ell-2}Y_{\ell-2}^{1}]. \end{aligned}$$

Let $m' = m+2, \ell' = \ell-2$, and change back the index into m, ℓ. The formula below can be derived.

$$\nabla_+^2[\phi] = \sum_{\ell=0}^{p-2}\left[\left(\sum_{m=2}^{\ell}+\sum_{m=-\ell}^{0}\right)D_{pp}(\ell,m)-D_{pp}(\ell,1)\right].$$

Similarly, according to Theorem 2, the *pth* order local expansion of the potential ϕ under operator ∇_-^2 at $\mathbf{x}$ can be obtained

$$\nabla_-^2[\phi] = \sum_{\ell=0}^{p-2}\left[\left(\sum_{m=0}^{\ell}+\sum_{m=-\ell}^{-2}\right)D_{mm}(\ell,m)-D_{mm}(\ell,-1)\right].$$

$$\begin{aligned}
\nabla_-\nabla_+[\phi] &= \nabla_- \left[\sum_{\ell=0}^{p-1}(\sum_{m=-\ell}^{0} - \sum_{m=1}^{\ell})D_p(\ell,m)\right] \\
&= \sum_{\ell=0}^{p-1}\left[(\sum_{m=-\ell}^{0} - \sum_{m=1}^{\ell})(2\ell+3)\beta_+(\ell+1,m-1)L_{\ell+1}^{m-1}\nabla_-[r^\ell Y_\ell^m]\right] \\
&= \sum_{\ell=1}^{p-1}\left[\sum_{m=-\ell}^{\ell}(2\ell+3)\beta_+(\ell+1,m-1)L_{\ell+1}^{m-1}(2\ell+1)\beta_-(\ell,m)r^{\ell-1}Y_{\ell-1}^{m-1}\right].
\end{aligned}$$

Let $\ell' = \ell - 1, m' = m - 1$ and change back the indexes, the formula below can be derived

$$\nabla_-\nabla_+[\phi] = \sum_{\ell=0}^{p-2}\sum_{m=-\ell}^{\ell} D_{pm}(\ell,m).$$

$$\begin{aligned}
\nabla_+\nabla_-[\phi] &= \nabla_+ \left[\sum_{\ell=0}^{p-1}(\sum_{m=-\ell}^{-1} - \sum_{m=0}^{\ell})D_m(\ell,m)\right] \\
&= \sum_{\ell=0}^{p-1}\left[(\sum_{m=-\ell}^{-1} - \sum_{m=0}^{\ell})(2\ell+3)\beta_-(\ell+1,m+1)L_{\ell+1}^{m+1}\nabla_+[r^\ell Y_\ell^m]\right] \\
&= \sum_{\ell=1}^{p-1}\left[\sum_{m=-\ell}^{\ell}(2\ell+3)\beta_-(\ell+1,m+1)L_{\ell+1}^{m+1}(2\ell+1)\beta_+(\ell,m)r^{\ell-1}Y_{\ell-1}^{m+1}\right].
\end{aligned}$$

Let $\ell' = \ell - 1, m' = m + 1$ and change back the indexes, one could obtain

$$\nabla_+\nabla_-[\phi] = \sum_{\ell=0}^{p-2}\sum_{m=-\ell}^{\ell} D_{mp}(\ell,m). \tag{2.74}$$

From the proof of Theorem 2, there is

$$\begin{aligned}
\frac{\partial^2\phi}{\partial z^2} &= \nabla_0^2\phi = \nabla_0\left[\sum_{\ell=0}^{p-1}\sum_{m=-\ell}^{\ell}(2\ell+3)\alpha(\ell+1,m)L_{\ell+1}^m r^l Y_\ell^m\right] \\
&= \sum_{\ell=1}^{p-1}\sum_{m=-\ell}^{\ell}(2\ell+3)\alpha(\ell+1,m)L_{\ell+1}^m(2\ell+1)\alpha(\ell,m)r^{\ell-1}Y_{\ell-1}^m \\
&= \sum_{\ell=0}^{p-2}\sum_{m=-\ell}^{\ell} D_{00}(\ell,m).
\end{aligned}$$

$$\nabla_+\nabla_0[\phi] = \sum_{\ell=0}^{p-2}\left[\sum_{m=1}^{\ell} -D_{p0}(\ell,m) + \sum_{m=-\ell}^{0} D_{p0}(\ell,m)\right],$$

$$\nabla_- \nabla_0[\phi] = \sum_{\ell=0}^{p-2} \left[\sum_{m=0}^{\ell} -D_{m0}(\ell,m) + \sum_{m=-\ell}^{-1} D_{m0}(\ell,m) \right].$$

$$\begin{aligned}
& \frac{\partial^2 \phi}{\partial x^2} = \frac{1}{2}(\nabla_+ - \nabla_-)^2 = \frac{1}{2}(\nabla_+^2 - 2\nabla_+\nabla_- + \nabla_-^2) \\
= \; & \frac{1}{2}\sum_{\ell=0}^{p-2} \left[\sum_{m=2}^{\ell} (D_{pp}(\ell,m) + D_{mm}(\ell,-m)) + \sum_{m=0}^{\ell} (D_{mm}(\ell,m) + D_{pp}(\ell,-m)) \right. \\
& \left. - (D_{pp}(\ell,1) + D_{mm}(\ell,-1)) - 2\sum_{m=-\ell}^{\ell} (D_{mp}(\ell,m) + D_{pm}(\ell,m)) \right] \\
= \; & \frac{1}{2}\sum_{\ell=0}^{p-2} \left[\sum_{m=2}^{\ell} (D_{pp}(\ell,m) + D_{pp}(\ell,m)^*) + \sum_{m=0}^{\ell} (D_{mm}(\ell,m) + D_{mm}(\ell,m)^*) \right. \\
& \left. - (D_{pp}(\ell,1) + D_{pp}(\ell,1)^*) - 2\sum_{m=-\ell}^{\ell} (D_{mp}(\ell,m) + D_{pm}(\ell,m)) \right] \\
= \; & \sum_{\ell=0}^{p-2} \left[\sum_{m=2}^{\ell} \Re(D_{pp}(\ell,m)) + \sum_{m=0}^{\ell} \Re(D_{mm}(\ell,m)) - \Re(D_{pp}(\ell,1)) \right. \\
& \left. - \sum_{m=-\ell}^{\ell} (D_{mp}(\ell,m) + D_{pm}(\ell,m)) \right].
\end{aligned}$$

Similarly derivation could be obtained as:

$$\begin{aligned}
\frac{\partial^2 \phi}{\partial y^2} &= -\frac{1}{2}(\nabla_+ - \nabla_-)^2 = \frac{1}{2}(\nabla_+^2 + 2\nabla_+\nabla_- + \nabla_-^2) \\
&= -\sum_{\ell=0}^{p-2} \left[\sum_{m=2}^{\ell} \Re(D_{pp}(\ell,m)) + \sum_{m=0}^{\ell} \Re(D_{mm}(\ell,m)) - \Re(D_{pp}(\ell,1)) \right. \\
& \quad \left. + \sum_{m=-\ell}^{\ell} (D_{mp}(\ell,m) + D_{pm}(\ell,m)) \right],
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \phi}{\partial x \partial y} &= -\frac{i}{2}(\nabla_+^2 - \nabla_-^2) \\
&= -\frac{i}{2}\sum_{\ell=0}^{p-2} \left[\sum_{m=2}^{\ell} (D_{pp}(\ell,m) - D_{mm}(\ell,-m)) - \sum_{m=0}^{\ell} (D_{pp}(\ell,-m) - D_{mm}(\ell,m))) \right. \\
& \quad - (D_{pp}(\ell,1) - D_{mm}(\ell,-1))] \\
&= \sum_{\ell=0}^{p-2} \left[\sum_{m=2}^{\ell} \Im(D_{pp}(\ell,m)) - \sum_{m=0}^{\ell} \Im(D_{mm}(\ell,m)) - \Im(D_{pp}(\ell,1)) \right],
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial^2\phi}{\partial x\partial z} &= -\frac{1}{\sqrt{2}}(\nabla_+\nabla_0 - \nabla_-\nabla_0) \\
&= -\frac{1}{\sqrt{2}}\sum_{\ell=0}^{p-2}\left[\sum_{m=1}^{\ell}(-D_{p0}(\ell,m) - D_{m0}(\ell,-m)) + \sum_{m=0}^{\ell}(D_{p0}(\ell,-m) + D_{m0}(\ell,m))\right] \\
&= \sqrt{2}\sum_{\ell=0}^{p-2}\left[\sum_{m=1}^{\ell}\Re(D_{p0}(\ell,m)) - \sum_{m=0}^{\ell}\Re(D_{m0}(\ell,m))\right],
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2\phi}{\partial y\partial z} &= -\frac{i}{\sqrt{2}}(\nabla_+\nabla_0 + \nabla_-\nabla_0) \\
&= \frac{i}{\sqrt{2}}\sum_{\ell=0}^{p-2}\left[\sum_{m=1}^{\ell}(-D_{p0}(\ell,m) + D_{m0}(\ell,-m)) + \sum_{m=0}^{\ell}(D_{p0}(\ell,-m) - D_{m0}(\ell,m))\right] \\
&= \sqrt{2}\sum_{\ell=0}^{p-2}\left[\sum_{m=1}^{\ell}\Im(D_{p0}(\ell,m)) + \sum_{m=0}^{\ell}\Im(D_{m0}(\ell,m))\right].
\end{aligned}
$$

□

2.4 Algorithm for Adaptive FMM

In this section, a pseudo-code of adaptive FMM for computing the electrostatic potential, fields and field gradients will be presented. An adaptive tree data structure will be built recursively until the number of charges in each box is fewer than s, i.e. to be a childless box. If a box contains more than s particles, it will be subdivided into eight child boxes. And this box is considered as a parent box. Before the discussion, the following notations will be reviewed or introduced.

- B_0 is denoted as the computational domain with all particles. B_l is denoted as all nonempty boxes at the refinement level l. A box without charges or particles (empty box) will not be further considered.

- For each box b at the refinement level ℓ, it will associate with five lists denoted by U_b, V_b, W_b, X_b, Y_b.

- For each box b, the interaction list V_b will be further subdivided into $Uplist(b)$, $Downlist(b)$, $Northlist(b)$, $Southlist(b)$, $Eastlist(b)$, $Westlist(b)$ for the exponential expansions in $\mathcal{T}_{ML}$.

- Each box b is associated with fourteen expansions:

1. A *multipole expansion* $\Phi(b)$ described by (2.25) representing the potential generated by charges inside b which is valid in $\mathbb{R}^3 \setminus \{U_b \cup V_b\}$.
2. A *local expansion* $\Psi(b)$ described by (2.27) representing the potential generated by charges outside $U_b \cup W_b$.
3. Six *outgoing exponential expansions* W_b^{Up}, W_b^{Down}, W_b^{North}, W_b^{South}, W_b^{East}, W_b^{West} of the form Equation (2.38) representing the potential generated by charges located in b and valid in the $Uplist(b)$, $Downlist(b)$, $Northlist(b)$, $Southlist(b)$, $Eastlist(b)$.
4. Six *ingoing exponential expansions* V_b^{Up}, V_b^{Down}, V_b^{North},V_b^{South}, V_b^{East}, V_b^{West} of the form 2.42 representing the potential inside b generated by charges located in b and valid in the $Uplist(b)$, $Downlist(b)$, $Northlist(b)$, $Southlist(b)$, $Eastlist(b)$.

The algorithm below is the pseudo code for new version of the adaptive fast multipole method.

Algorithm 1 ADAPTIVE FMM

Input : Particle locations $\{\mathbf{x}_i\}_{i=1}^N$, charges $\{q_i\}_{i=1}^N$, precision ϵ, the maximum number of particles in leaf box s

Output : Potential $\Phi(\mathbf{x}_i)$, field $E(\mathbf{x}_i)$, and electric field gradients $\Phi_{xx}(\mathbf{x}_i)$, $\Phi_{yy}(\mathbf{x}_i)$, $\Phi_{zz}(\mathbf{x}_i)$, $\Phi_{xy}(\mathbf{x}_i)$, $\Phi_{yz}(\mathbf{x}_i)$, $\Phi_{xz}(\mathbf{x}_i)$.

Initialization

```
Choose $pth$ order of the multipole expansion according to $\epsilon$.
```

Step 1 : Generating Adaptive Oct-Tree Structure

```
for $l = 0, 1, \dots, l_{max}$ do                    ▷ $l_{max}$ - the maximum refinement level
    for each box $b \in B_l$ do
        if $b$ contains more than $s$ charges then
            Divided $b$ into eight child boxes. Add the nonempty child boxes to $B_{l+1}$.
        end if
    end for
end for                                              ▷ $nbox$ - Total number of boxes created
for each box $b_i, i = 1, \cdots, nbox$ do
    Create Lists $U_{b_i}, V_{b_i}, W_{b_i}, X_{b_i}$
    Subdivide $V_{b_i}$ into $Up, Down, North, South, East, West$ lists.
end for
```

Step 2 : Upward Pass - Multipole Expansion $\mathcal{T}_{SM}$

```
▷ A $pth$ order multipole expansion is formed for each box $b$ about its center.
for each box $b_i, i = 1, \cdots, nbox$ do
    if $b_i$ is childless then
        Use Theorem 2.2.2 to form $p$th order multipole expansion $\Phi_{b_i}$
    else
        Use Theorem 2.2.4 to merge multipole expansion from its children into $\Phi_{b_i}$.
    end if
end for
```

Step 3 : Downward Pass - Local Expansion $\mathcal{T}_{LT}$

```
▷ For each box $b$, add the contribution from particles in $X_b$ to its local expansion
for each box $b_i, i = 1, \cdots, nbox$ do
    for each box $c \in X_{b_i}$ do
        if the number of particles in $b_i$ is less than $p^2$ then
            Compute potential, field, field derivatives at each point of $\Phi_{b_i}$ directly from particle in $c$
        else
            Generate a local expansion at $b_i$'s center due to charges in $c$ using Theorem 2.2.3 and add it to $\Psi_{b_i}$.
        end if
    end for
end for
```

Step 4 : Downward Pass - Multipole-to-Local Translation $\mathcal{T}_{ML}$

▷ Multipole to Local translations:

1. For each box b on level $l, l = 2, \ldots, l_{max}$, create the outgoing exponential expansion W_b^{Dir} from the multipole expansion of b at every direction in V_b using Theorem 2.2.7.
2. Then translate W_b^{Dir} to the center of each box $c \in Dirlist(b)$ using Theorem 2.2.8 and add

the translated exponential expansion to its incoming exponential expansion V_c^{Dir}.

3. Convert V_c^{Dir} into a local expansion using Theorem 2.2.9 and add it to Ψ_c.

```
for  $l = 2, \dots, l_{max}$ do
    for $Dir = Up, Down, North, South, East, West$ do
        for each box $b \in B_l$ do
            for each box $c \in X_{b_i}$ do
                Convert $\Phi(b)$ into $W_b^{Dir}$ by $\mathcal{T}_{ME}$ in Theorem 2.2.7.
                Add the translated expansion to $V_c^{Dir}$
            end for
        end for
        for each box $c \in B_l$ do
            Convert $V_c^{Dir}$ into a local expansion using Theorem 2.2.8 and add it to $\Psi(c)$.
        end for
    end for
end for
```

Step 5 : Downward Pass - Local-to-Local Translation $\mathcal{T}_{LL}$

```
▷ Shift the center of the local expansion from each parent box $b$ to its children.
for each box $b_i, i = 1, \cdots, nbox$ do
    if $b_i$ is a parent box then
        Shift the local expansion $\Psi_{b_i}$ into the center of its children using $\mathcal{T}_{LL}$
        Add the translated expansions to children's local expansion.
    end if
end for
```

Step 6 : Evaluation of Potentials, Fields, Electric Field Gradients at leaf nodes

```
for each box $b_i, i = 1, \cdots, nbox$ do
    if $b_i$ is childless  then
        Use the local expansion $\Psi_{b_i}$ to compute the potential, field, electric field gradients at each
    charge of $b_i$.
    end if
end for
```

Step 7 : Evaluation of Potentials, Fields, Electric Field Gradients by W_b

```
▷ For each childless box $b$, evaluate the potentials, fields, electric field gradients contributed by particles in $W_b$.
for each box $b_i, i = 1, \cdots, nbox$ do
    if $b_i$ is childless then
        for each box $c \in W_{b_i}$ do
            if the number of charges in $c$ no more than $p^2$ then
                Compute the potential, fields, electric field gradients at each charge of $b_i$ directly from particles in $c$.
            else
                Compute the potential, fields, electric field gradients at each charge of $b_i$ from multipole expansion $\Phi_c$.
            end if
        end for
    end if
end for
```

Step 8:Evaluation of Potentials, Fields, Electric Field Gradients-Near-field Interaction

```
for each box $b_i, i = 1, \cdots, nbox$ do
    if $b_i$is childless then
        Compute the potential, fields, electric field gradients at each charge of $b_i$ directly for all charges in $U_{b_i}$.
    end if
end for
```

2.5 Numerical Results

The algorithm described in the previous section has been implemented as solvers. The numerical experiments are performed on nodes of killdevil, each node with 12-core, 2.93 GHz Intel processor. The numerical results for a variety of charge distributions are described in this section: particles randomly distributed in a cube, particles randomly distributed on the surface of a sphere, and particles randomly distributed on the surface of the first octant. For each distribution, the numerical tests were performed with 3-digit, 6-digit, and 9-digit accuracy. The maximum number of charges allowed in each leaf box is set to be 80. To satisfy the corresponding accuracy, the approximation terms p has been selected as 9 for 3-digit accuracy, 18 for 6-digit accuracy, and 27 for 9-digit accuracy. The L_2 norm of the relative error in the FMM approximation is computed as

$$Error = \sqrt{\frac{\sum_{i=1}^{N} |\Phi(\mathbf{x}_i) - \Phi_i|^2}{\sum_{i=1}^{N} |\Phi(\mathbf{x}_i)|^2}} \tag{2.75}$$

Here, assume $\Phi(\mathbf{x}_i)$ is the potentials of particles computed by direct evaluation for particle i and Φ_i is the numerical approximation by the new version of FMM for particle i. Similarly, the error for the fields and electric field gradients can be defined. Below are some notations for the numerical results.

- N - the number of the particles.
- $Levels$ - the number of levels used in the hierarchy tree structure.
- $Boxes$ - the number of the multipole expansion used.
- p - the number of the approximation terms for different accuracy. Here, $p = 9, 18, 27$ for 3-digit, 6-digit, 9-digit accuracy correspondingly.
- $T_{fmm-idigit}$ - the time for ith digit accuracy using FMM, $i = 3, 6, 9$.
- T_{dir} - the time for direct evaluation of all particles.

2.5.1 Cube

In the first set of the numerical experiments, the charges were distributed randomly in a $[-5, 5] \times [-5, 5] \times [-5, 5]$ cube. The numerical error results with 3-digit, 6-digit and 9-digit accuracy with 12 threads are presented in Table 2.1, Table 2.2, and Table 2.3.

In Table 2.4, the running-time results produced by the new version of adaptive FMM with different accuracy are compared to the direct evaluation of the potential, fields, and the electric field gradients for all particles.

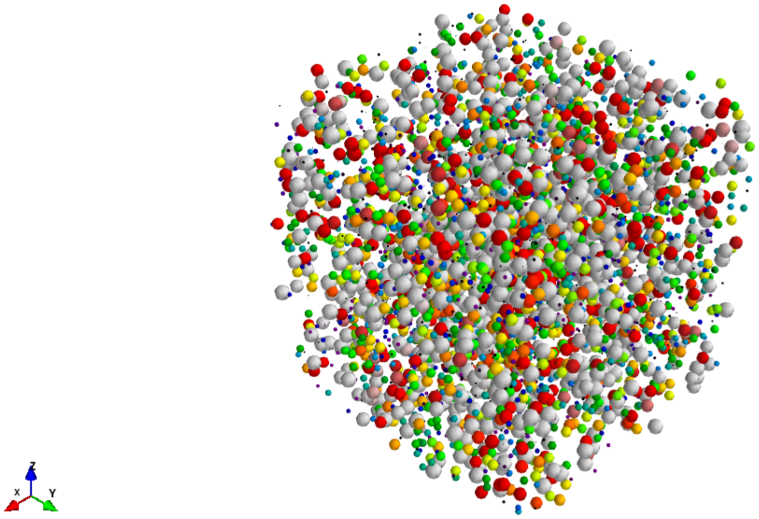

Figure 2.4: Charges randomly distributed in a cube

Table 2.1: Error Results of the new version of FMM for 3-digit accuracy with charges randomly distributed in a cube

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	4.9e-5	2.4e-4	1.8e-4	4.4e-4	2.6e-6	1.5e-6	3.0e-6	3.0e-6	5.8e-6	5.4e-6
20000	4.9e-5	2.9e-4	1.8e-4	3.2e-4	1.4e-6	1.3e-6	3.9e-6	2.9e-6	5.2e-6	1.1e-6
50000	5.2e-5	2.9e-4	3.9e-4	4.2e-4	8.9e-6	3.6e-6	4.4e-6	3.2e-6	1.7e-6	3.0e-5
100000	5.2e-5	2.9e-4	3.7e-4	5.2e-4	4.2e-6	3.9e-6	3.6e-6	1.1e-6	1.8e-6	2.3e-6
200000	5.2e-5	3.5e-4	4.5e-4	5.8e-4	7.7e-6	7.9e-6	7.9e-6	2.8e-6	4.4e-5	5.6e-6
500000	5.3e-5	3.6e-4	4.8e-4	6.0e-4	8.6e-6	7.1e-6	6.7e-6	8.5e-6	6.0e-5	8.8e-6
1000000	5.3e-5	3.6e-4	4.7e-4	6.1e-4	2.8e-6	2.2e-6	5.9e-6	2.2e-6	3.9e-6	2.0e-6

Table 2.2: Error results of the new version of FMM for 6-digit accuracy with charges randomly distributed in a cube

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	3.5e-8	1.6e-7	1.0e-7	2.0e-6	6.6e-9	3.7e-9	7.2e-9	6.6e-9	1.3e-8	1.1e-8
20000	3.5e-8	1.7e-7	8.0e-8	1.2e-7	3.1e-9	2.5e-9	8.1e-9	5.0e-9	1.0e-8	1.8e-9
50000	4.0e-7	1.8e-7	1.7e-7	1.6e-7	2.3e-8	9.0e-9	1.0e-8	7.1e-9	3.8e-9	6.4e-9
100000	4.0e-7	1.6e-7	1.4e-7	1.7e-7	9.6e-9	8.3e-9	7.8e-9	2.2e-9	3.5e-9	4.1e-9
200000	4.0e-7	1.7e-7	1.5e-7	1.7e-7	1.5e-8	1.4e-8	1.4e-8	4.5e-9	7.2e-9	8.4e-9
500000	4.1e-8	1.8e-7	1.6e-7	1.7e-7	2.1e-8	1.6e-8	1.5e-8	1.7e-8	1.2e-8	1.6e-8
1000000	4.1e-8	1.7e-7	1.4e-7	1.7e-7	5.7e-9	4.2e-9	1.2e-8	3.6e-9	6.8e-9	3.2e-9

Table 2.3: Error results of the new version of FMM for 9-digit accuracy with charges randomly distributed in a cube

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	3.8e-11	1.2e-10	6.8e-11	1.5e-10	9.0e-12	4.4e-12	9.1e-12	8.4e-12	1.7e-11	1.3e-11
20000	3.8e-11	1.1e-10	5.4e-11	8.8e-11	3.9e-12	3.3e-12	1.0e-11	6.9e-12	1.2e-11	2.2e-12
50000	4.3e-11	1.1e-10	1.1e-10	1.2e-10	3.1e-11	1.2e-11	1.4e-11	9.6e-12	4.8e-12	7.8e-12
100000	4.3e-11	8.1e-11	8.6e-11	1.2e-10	1.3e-11	1.1e-11	9.7e-12	2.9e-12	4.4e-12	4.9e-12
200000	4.3e-11	7.3e-11	7.9e-11	1.2e-10	1.8e-11	1.7e-11	1.7e-11	5.5e-12	8.4e-12	9.5e-12
500000	4.5e-11	7.5e-11	8.2e-11	1.2e-10	2.7e-11	2.1e-11	1.9e-11	2.2e-11	1.5e-11	2.0e-11
1000000	4.5e-11	5.7e-11	6.3e-11	1.1e-10	6.9e-12	5.1e-12	1.4e-11	4.4e-12	8.0e-11	3.6e-12

Table 2.4: Timing results of the new version of FMM with charges randomly distributed in a cube

N	Levels	Boxes	p	s	$T_{fmm-3digit}$	$T_{fmm-6digit}$	$T_{fmm-9digit}$	T_{dir}
10000	3	585	9-18-27	80	0.05s	0.08s	0.16s	0.63s
20000	3	585	9-18-27	80	0.14s	0.21s	0.31s	2.50s
50000	4	4537	9-18-27	80	0.26s	0.45s	0.91s	15.6s
100000	4	4681	9-18-27	80	0.52s	0.86s	1.52s	62.51s
200000	4	4681	9-18-27	80	1.76s	2.27s	3.31s	250.00s
500000	5	37449	9-18-27	80	2.21s	4.30s	8.43s	1565s
1000000	5	37449	9-18-27	80	6.32s	9.62s	15.6s	6134s

If the electric field gradients are computed, there will be six more variables to be evaluated comparing the original code with one potential variable plus three variables for the fields. Thus, the running time of the near-field part should be around 2.5 times of the code without computing second order derivatives. The efficiency of the far-field approximation in the original FMM has been improved by using the exponential expansion in the new version of FMM. The approximation expansions terms are respectively 9,18, and 27 for 3-digit, 6-digit, and 9-digit accuracy. The far-field running time is supposed to be 1 : 4 : 9. After adding the function for computing the electric field gradients, the far-field running time only dominates for larger digit accuracy. Therefore, the ratio between the near-field running time and the far-field running time will be different comparing the code without computing the electric field gradients or with the original FMM.

2.5.2 Surface of A Unit Sphere

The second set of numerical experiments is for charges distributed randomly on the surface of the unit sphere centered at the origin. Table 2.5, Table 2.6, and Table 2.7 are the L_2 norm numerical error results of potential, fields and electric field gradients computed by the new version of FMM. In Table 2.8, the running-time results produced by the new version of adaptive FMM with different accuracy are compared to the direct evaluation of the potential, fields, and the electric field gradients for all particles randomly distributed on the surface of the sphere.

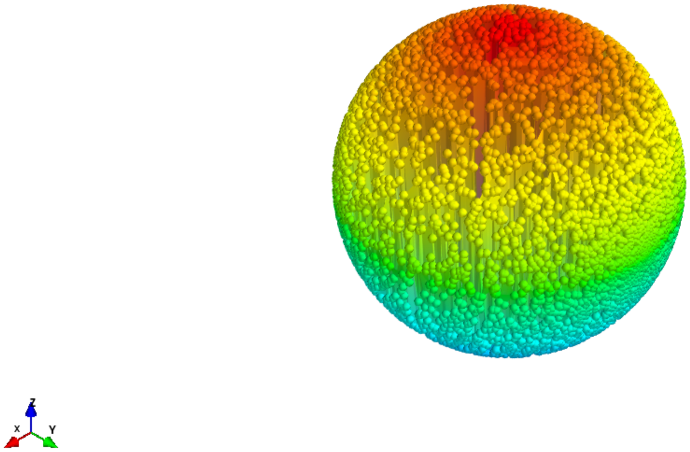

Figure 2.5: Charges randomly distributed on the surface of the sphere

Table 2.5: Error results of the new version of FMM for 3-digit accuracy with charges randomly distributed on the surface of a sphere

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	1.2e-4	1.3e-6	4.8e-6	2.4e-5	8.9e-9	2.6e-8	2.1e-7	2.4e-8	2.3e-7	8.0e-7
20000	1.2e-4	9.3e-7	1.2e-6	1.0e-5	5.8e-9	1.5e-8	4.6e-9	2.9e-9	3.0e-8	4.3e-8
50000	1.3e-4	1.4e-6	1.1e-6	4.6e-6	1.4e-8	6.4e-9	7.6e-9	8.3e-9	3.9e-8	2.7e-8
100000	1.3e-4	7.8e-7	1.3e-6	1.8e-5	5.2e-9	1.1e-8	6.8e-9	8.0e-8	4.8e-7	7.3e-7
200000	1.3e-4	1.3e-6	6.9e-7	1.8e-5	3.8e-9	2.1e-9	3.2e-9	3.3e-9	2.3e-7	4.9e-8
500000	1.5e-4	6.6e-7	4.8e-7	4.6e-6	2.2e-8	2.7e-9	2.4e-9	1.9e-9	2.9e-8	5.5e-8
1000000	1.6e-4	2.2e-6	1.9e-6	2.4e-6	4.2e-8	1.9e-8	2.4e-8	1.3e-8	1.1e-8	1.8e-8

Table 2.6: Error results of the new version of FMM for 6-digit accuracy with charges randomly distributed on the surface of a sphere

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	7.1e-8	1.6e-9	5.1e-9	2.6e-8	2.2e-11	6.9e-11	6.1e-11	5.4e-11	4.1e-10	1.3e-9
20000	7.1e-8	1.1e-9	1.8e-9	1.3e-8	1.8e-11	6.9e-11	2.1e-11	5.8e-12	6.7e-11	1.2e-10
50000	9.7e-8	2.3e-9	2.2e-9	6.0e-9	6.5e-11	3.2e-11	3.8e-11	1.7e-11	6.9e-11	1.4e-10
100000	8.8e-8	1.3e-9	2.3e-9	1.5e-8	2.2e-11	5.1e-11	3.7e-11	1.4e-11	1.1e-9	2.0e-9
200000	8.4e-8	1.8e-9	1.0e-9	1.0e-8	1.1e-11	6.2e-12	1.2e-11	7.4e-12	3.7e-10	1.1e-10
500000	8.9e-8	9.2e-10	6.2e-10	2.8e-9	6.2e-11	7.7e-12	7.5e-12	3.0e-12	7.4e-11	1.3e-10
1000000	9.9e-8	2.6e-9	1.9e-9	1.5e-9	1.1e-10	4.7e-11	5.7e-11	2.2e-11	2.5e-11	5.8e-11

Table 2.7: Error results of the new version of FMM for 9-digit accuracy with charges randomly distributed on the surface of a sphere

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	4.5e-11	1.5e-12	4.3e-12	2.0e-11	3.1e-14	8.7e-14	7.3e-14	7.8e-14	6.6e-13	2.1e-12
20000	4.4e-11	9.3e-13	1.5e-12	1.3e-11	2.8e-14	8.6e-14	2.5e-14	1.0e-14	1.3e-13	2.3e-13
50000	5.3e-11	2.3e-12	2.0e-12	6.8e-12	7.9e-14	6.3e-14	7.2e-14	3.6e-14	3.1e-13	2.5e-13
100000	6.1e-11	1.6e-12	2.8e-12	4.3e-11	3.9e-14	8.8e-14	5.3e-14	5.0e-13	7.0e-11	1.1e-11
200000	5.7e-11	2.0e-12	1.5e-12	3.5e-11	2.8e-14	2.0e-14	2.1e-14	3.0e-14	2.4e-12	6.5e-13
500000	6.0e-11	1.1e-12	1.1e-12	1.1e-11	2.7e-13	2.5e-14	2.1e-14	1.9e-14	5.0e-13	1.1e-12
1000000	7.8e-11	3.6e-12	3.2e-12	6.1e-12	2.3e-13	1.0e-13	8.6e-14	1.2e-13	8.7e-14	3.8e-13

Table 2.8: Timing results of the new version of FMM with charges randomly distributed on the surface of a sphere

N	Levels	Boxes	p	s	$T_{fmm-3digit}$	$T_{fmm-6digit}$	$T_{fmm-9digit}$	T_{dir}
10000	6	427	9-18-27	80	0.06s	0.11s	0.19s	0.63s
20000	7	890	9-18-27	80	0.19s	0.19s	0.31s	2.50s
50000	8	2496	9-18-27	80	0.32s	0.49s	0.82s	15.62s
100000	10	4037	9-18-27	80	0.61s	0.95s	1.54s	62.5s
200000	11	9355	9-18-27	80	1.26s	2.00s	3.28s	250.0s
500000	12	20741	9-18-27	80	2.99s	4.70s	7.72	1565s
1000000	13	46209	9-18-27	80	5.86s	9.42s	15.8s	6135s

2.5.3 First Octant of the Unit Sphere

The third set of numerical experiments is for charges distributed randomly on the surface of the first octant of the unit sphere centered at the origin. Table 2.9, Table 2.10, and Table 2.11 are for the L_2 norm numerical error results of potential, fields and electric field gradients computed by the new version of FMM. In Table 2.12, the timing results produced by the new version of adaptive FMM with different accuracy are compared to the direct evaluation of the potential, fields, and the electric field gradients for all particles randomly distributed on the surface of the first octant of the unit sphere.

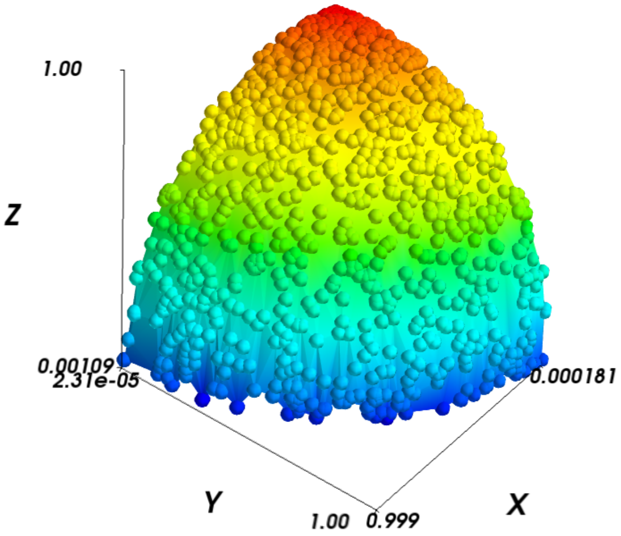

Figure 2.6: Charges randomly distributed on the surface of the first octant

Table 2.9: Error results of the new version of FMM for 3-digit accuracy with charges randomly distributed on the surface of the first octant

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	1.0e-4	1.8e-6	3.3e-6	1.3e-5	8.8e-9	1.4e-8	1.6e-8	2.5e-8	1.6e-7	2.7e-7
20000	1.2e-4	1.7e-6	1.5e-6	5.9e-6	3.9e-8	1.1e-8	1.3e-8	6.8e-9	1.4e-7	5.5e-8
50000	1.3e-4	1.9e-6	2.7e-6	4.3e-6	2.7e-8	5.3e-8	2.2e-8	1.9e-8	3.2e-8	1.0e-7
100000	1.2e-4	3.3e-7	9.7e-7	1.0e-5	1.0e-9	2.7e-9	1.7e-9	2.3e-9	2.1e-7	2.0e-7
200000	1.1e-4	1.4e-7	1.1e-6	2.9e-6	3.2e-10	7.8e-10	5.8e-10	1.4e-9	3.9e-8	2.5e-8
500000	1.5e-4	7.6e-7	1.4e-6	2.2e-6	3.7e-9	7.2e-9	6.4e-9	1.2e-8	5.8e-9	1.8e-8
1000000	1.5e-4	6.4e-7	5.2e-7	7.9e-7	4.7e-9	1.8e-9	2.3e-9	1.4e-9	2.0e-9	2.6e-9

Table 2.10: Error results of the new version of FMM for 6-digit accuracy with charges randomly distributed on the surface of the first octant

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	6.5e-8	2.4e-9	4.4e-9	2.1e-8	2.2e-11	3.5e-11	4.7e-11	4.8e-11	4.5e-11	6.7e-10
20000	7.4e-8	2.3e-9	1.9e-9	9.2e-9	6.7e-11	2.5e-11	2.9e-11	1.5e-11	3.3e-10	1.2e-10
50000	8.1e-8	2.3e-9	3.2e-9	4.7e-9	7.3e-11	1.4e-10	7.6e-11	4.3e-11	9.1e-11	2.3e-10
100000	8.0e-8	3.8e-10	1.3e-9	1.1e-8	3.4e-12	7.1e-12	5.3e-12	4.3e-12	4.3e-10	6.1e-10
200000	7.1e-8	2.3e-10	1.8e-9	2.7e-9	1.2e-12	3.1e-12	2.9e-12	3.5e-12	8.5e-11	6.9e-11
500000	9.6e-8	1.1e-9	1.8e-9	1.9e-9	1.3e-11	1.8e-11	2.6e-11	2.2e-11	1.3e-11	3.9e-11
1000000	9.8e-8	1.1e-9	7.7e-10	5.5e-10	2.1e-11	5.4e-12	1.1e-11	4.1e-12	4.0e-12	6.0e-12

Table 2.11: Error results of the new version of FMM for 9-digit accuracy with charges randomly distributed on the surface of the first octant

N	Φ	Φ_x	Φ_y	Φ_z	Φ_{xx}	Φ_{yy}	Φ_{zz}	Φ_{xy}	Φ_{xz}	Φ_{yz}
10000	4.2e-11	3.0e-12	4.3e-12	2.2e-11	3.5e-14	7.9e-14	8.7e-14	9.2e-14	1.2e-12	1.2e-12
20000	4.1e-11	3.2e-12	1.9e-12	9.6e-12	2.5e-13	8.1e-14	3.7e-14	3.7e-14	1.3e-12	3.0e-13
50000	4.5e-11	3.3e-12	3.2e-12	8.0e-12	2.8e-13	4.2e-13	2.0e-13	1.4e-13	3.9e-13	1.0e-12
100000	4.4e-11	7.4e-13	1.6e-12	2.4e-11	1.2e-14	1.5e-14	1.7e-14	1.8e-14	4.5e-12	4.3e-12
200000	4.1e-11	3.0e-13	2.1e-12	6.5e-12	5.6e-15	9.0e-15	9.0e-15	1.5e-14	5.0e-13	5.9e-13
500000	5.4e-11	1.2e-12	2.1e-12	4.8e-12	2.2e-14	3.6e-14	4.5e-14	5.8e-14	5.6e-14	1.4e-13
1000000	5.3e-11	1.0e-12	1.0e-12	1.7e-12	5.9e-14	1.6e-14	2.3e-14	1.9e-14	2.3e-14	3.7e-14

Table 2.12: Timing results of the new version of FMM with charges randomly distributed on the surface of the first octant

N	Levels	Boxes	p	s	$T_{fmm-3digit}$	$T_{fmm-6digit}$	$T_{fmm-9digit}$	T_{dir}
10000	7	428	9-18-27	80	0.06s	0.10s	0.16s	0.63s
20000	8	861	9-18-27	80	0.11s	0.18s	0.30s	2.50s
50000	9	2345	9-18-27	80	0.38s	0.51s	0.84s	15.7s
100000	10	4170	9-18-27	80	0.57s	0.92s	1.52s	62.5s
200000	11	9645	9-18-27	80	1.25s	1.90s	3.16s	250.1s
500000	12	20620	9-18-27	80	3.06s	4.75s	7.77s	1565s
1000000	14	46158	9-18-27	80	6.15s	9.54s	15.9s	6134.9s

CHAPTER 3

Adaptive Rotne-Prager-Yamakawa Solver by Multipole Methods

In this chapter, a fast algorithm will be presented for computing the matrix-vector multiplication, which is essential for the simulation of Brownian dynamics with hydrodynamic interactions. The algorithm is based on the development of the new version of the FMM solver with new functions of computing the field gradients as shown in the last chapter. The diffusion matrix is represented by the Rotne-Prager-Yamakawa tensor which is configuration-dependent. The Brownian particles with a small radius are spherically symmetric. The hydrodynamic forces mediated by the fluid are assumed to act on the center of the particles.

To describe hydrodynamic interactions, consider an ensemble of N particles suspended in a viscous fluid with a low Reynolds number. The motion of a particle in the fluid will induce a local flow field, which will be felt by the other particles. As a result, the other particles will experience a force mediated by the fluid. By generating and reacting to a local fluid velocity, particles experience interactions with each other mediated by the solvent, which is called the *hydrodynamic interactions* between the particles.

The hydrodynamic interactions are long-range, many-body interactions. Due to the multi-body and nonlinear characteristics of the hydrodynamic interaction, it is hard to take the full complexity into consideration. Various approximations have been introduced to make the computation more tractable. Usually, the hydrodynamic interactions are described by a position-dependent inter-particle diffusion tensor. The Rotne-Prager-Yamakawa approximation [76] is one of the approximations that has been widely used to study the mobility of spherical particles with a finite size in an unbounded domain. It is very popular in modeling the hydrodynamic interactions in colloidal suspensions and polymer solutions. There are mainly two reasons why RPY approximation is so popular. Firstly, it includes all long-range force terms up to R^{-3}, where R is the distance between the two particles. Secondly, the diffusion matrix described by RPY tensor is positive definite when the two particles are not overlapping (the distance of the center of the two particles is larger than the diameter of the particles). This characteristic ensures the fluctuation-dissipation balance [76] and is very important in Brownian Dynamic simulations. When generating the Brownian forces, the square root or the Cholesky decomposition of the covariance matrix has to be computed to make sure the Brownian forces satisfy the fluctuation-dissipation theorem. Without the symmetric positive definite property, the results of the Brownian dynamic simulations will result in a nonphysical behavior. Something needed to be pointed out is that the lubrication is not included in the RPY tensor. If the particles get very close (less than the diameter of the particle), corrections have to be added to make sure the diffusion matrix is positive-definite. Only the pairwise approximation (two-body hydrodynamic interactions) is included in the RPY tensor and therefore it is not a good approximation to study solutions with a high particle concentration [77].

Assume $\mathbf{D}$ is the diffusion matrix describing the hydrodynamic interactions of the particles with dimension $3N \times 3N$ and $\mathbf{F}$ is the deterministic non-hydrodynamic force vector exerted on the particles with dimension $3N$. The non-hydrodynamic forces are forces due to the interaction potential energy between Brownian particles or external forces exerted on the particles. In Brownian dynamics simulations, $\mathbf{D} \cdot \mathbf{F}$ is constantly required to be evaluated to study the motion of the particles, as well as the displacement of the particles due to the systematic forces $\mathbf{F}$ exerted on the spherical particles. Since $\mathbf{D}$ involves the hydrodynamic interactions between all particles, it is a dense matrix. Direct evaluation of $\mathbf{D} \cdot \mathbf{F}$ will be $O(N^2)$ complexity. Also, the memory requirement is expensive if the matrix $\mathbf{D}$ is generated explicitly. According to Liu and Chow [28], for 10K particles, storage for $\mathbf{D}$ is as huge as 32 GB.

To overcome the $O(N^2)$ cost of direct evaluation, various fast algorithms have been developed to reduce the complexity to $O(NlogN)$ or $O(N)$. In 1986, Beenakker provided [29] Ewald summation for the RPY mobility tensor and variant. Later on, Particle mesh techniques including the Particle Mesh Ewald (PME) method [30] and the Smooth Particle Mesh Ewald (SPME) method [31] have been extensively used for calculating hydrodynamic interactions with $O(NlogN)$ computation scaling. Other methods include methods of local corrections [32], multigrid methods [33], panel clustering methods, precorrected-FFT [34] and so on. In 2013, Liang decomposed the RPY tensor into harmonic potentials and its fields, which required four harmonic FMM calls and the harmonic potentials are evaluated by kernel-independent FMM [78]. In this chapter, the RPY tensor is decomposed into four harmonic FMM calls with different charges. The decomposition here required evaluating potentials, electric fields, and the field gradients. The algorithm for evaluation those terms is described in Chapter 2. Another difference is the harmonic potentials are evaluated by the new version of the fast multipole method [1] which works efficient in three-dimensional simulations. The fast RPY solver has been parallelized on the multi-core and multi-processor systems. The numerical results show that the method shown below has a better performance. Specifically, there is going to be a three-dimensional force vector corresponding to each particle. To apply the new version of fast multipole methods, the far-field RPY potential is decomposed into four Laplace FMM calls, with each particle corresponding to one scalar component related to the force. This scalar component is sometimes marked as "charge" in this chapter to match the description of the electrostatic potential in Chapter 2. They are not physical charges. The adaptive, fast RPY solver developed in this dissertation computationally enables the large-scale complex Brownian dynamics simulations viable and has a direct application to the problem of how to deal with the long-range hydrodynamic interactions in computer simulations of macromolecular solutions.

In the first section of this chapter, the definition of the RPY tensor and the formulation of the RPY potential will be introduced. Here, the force exerted on each particle are treated as charges to keep consistent with the description in Chapter 2. The matrix-vector multiplication between the diffusion matrix and the force vector is called potentials. In the second section, it will show how to decompose the RPY tensor into four harmonic potentials, electric fields, and field gradients and apply the Laplace FMM subroutines to collect the far-field contributions of RPY potential. In the third section, a pseudocode for the fast RPY algorithm with multipole methods will be presented. The numerical results in the last section are to illustrate the performance of the fast RPY solver developed in this thesis.

3.1 Rotne-Prager-Yamakawa Tensor

Given N particles with charges $\{q_i\}_{i=1}^N$ and locations $\{\mathbf{x}_i\}_{i=1}^N$, the definition of the RPY tensor is as below:

$$\mathbf{D}(\mathbf{x}_m, \mathbf{x}_n) = \begin{cases} \frac{k_BT}{6\pi\eta a}\mathbf{I} & m = n \\ \frac{k_BT}{8\pi\eta r_{mn}}\left[(\mathbf{I} + \frac{\mathbf{r_{mn}}\otimes\mathbf{r_{mn}}}{r_{mn}^2}) + \frac{2a^2}{3r_{mn}^2}(\mathbf{I} - 3\frac{\mathbf{r_{mn}}\otimes\mathbf{r_{mn}}}{r_{mn}^2})\right] & m \neq n \text{ and } r_{mn} > 2a, \\ \frac{k_BT}{6\pi\eta a}\left[(1 - \frac{9}{32}\frac{r_{mn}}{a})\mathbf{I} + \frac{3}{32a}\frac{\mathbf{r_{mn}}\otimes\mathbf{r_{mn}}}{r_{mn}}\right] & m \neq n \text{ and } r_{mn} \leq 2a. \end{cases} \quad (3.1)$$

Here, k_B is the Boltzmann constant. T is the absolute temperature and η is the solvent viscosity. a is the radius of each bead. m, n represents the indices of the particles. r_{mn} is the distance between the particle i and particle j. $\mathbf{r_{mn}} = \mathbf{x_m} - \mathbf{x_n}$. $\mathbf{I}$ is the 3×3 identity matrix. $\otimes$ is the outer product and $\mathbf{D}(\mathbf{x}_m, \mathbf{x}_n)$ is a 3×3 matrix.

An important property of the RPY tensor is

$$\sum_n \frac{\partial \mathbf{D}(\mathbf{x}_m, \mathbf{x}_n)}{\partial r_n} = 0. \quad (3.2)$$

This property greatly simplifies the computation of the displacements for particles since it is not necessary to compute the gradient of the diffusion tensor. For each particle $m, m = 1, \ldots, N$, one requires to evaluate

$$\mathbf{U}(\mathbf{x}_m) = \sum_{n=1}^{N} \mathbf{D}(\mathbf{x}_m, \mathbf{x}_n)\mathbf{F}_n, \quad (3.3)$$

where $\mathbf{U}(\mathbf{x_m}) = (\mathbf{U}_m^{(1)}, \mathbf{U}_m^{(2)}, \mathbf{U}_m^{(3)})^T$ is the RPY potential at $\mathbf{x}_m$. $\mathbf{D}(\mathbf{x}_m, \mathbf{x}_n)$ satisfies Equation (3.1) and $\mathbf{F}_n$ is the systematic force exerted on particle n.

To rapidly evaluate Equation (3.3), the same adaptive oct-tree data structure as was used in fast multipole method will be applied. The strategy is to cluster particles at various spatial lengths and compute the interactions with other clusters that are sufficiently far away. The parallelized new version of the fast multipole method for the three-dimensional Laplace equation [1] has been employed to compute the far-field part. Interactions with nearby particles will be handled directly.

$$\begin{aligned}\mathbf{U}(\mathbf{x}_m) &= \sum_{\mathbf{x_n}\in\Omega_{near}} \mathbf{D}(\mathbf{x_m},\mathbf{x_n})\mathbf{F_n} + \sum_{\mathbf{x_n}\in\Omega_{far}} \mathbf{D}(\mathbf{x_m},\mathbf{x_n})\mathbf{F_n} \\ &= \mathbf{U}_{near}(\mathbf{x}_m) + \mathbf{U}_{far}(\mathbf{x}_m) \end{aligned} \tag{3.4}$$

Here, if $\mathbf{x}_m$ is in a box b, Ω_{near} means the list U_b and Ω_{far} are the remaining four lists V_b, W_b, X_b, Y_b which are well-seperated from box b. Assume the spherical particles have a small radius. For two particles m and m in well-separated boxes(or no overlapping), the relative position between particle m and n $r_{mn} = |\mathbf{x_m} - \mathbf{x_n}| > 2a$ and how to evaluate the far-field RPY potential will be focused on in the next section.

3.1.1 Far-field Rotne-Prager-Yamakawa Evaluation with Multipole Methods

For any two particles m and n, when $r_{mn} = |\mathbf{x_m} - \mathbf{x_n}| > 2a$, for $i,j = 1,2,3$, rewrite the formulation of $\mathbf{D}(\mathbf{x}_m,\mathbf{x}_n)$. Here, $\mathbf{D}_{ij}(\mathbf{x}_m,\mathbf{x}_n)$ means the element on the ith row and jth floor of the 3×3 matrix $\mathbf{D}(\mathbf{x}_m,\mathbf{x}_n)$. Before started, two formulas are shown as below:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{x}_m^{(i)}}\left(\frac{1}{r_{mn}}\right) &= -\frac{\mathbf{r}_{mn}^{(i)}}{r_{mn}^3}, \\ \frac{\partial^2}{\partial \mathbf{x}_m^{(i)}\partial \mathbf{x}_m^{(j)}}\left(\frac{1}{r_{mn}}\right) &= -\left(\frac{\delta_{ij}}{r_{mn}^3} - \frac{3\mathbf{r}_{mn}^{(i)}\mathbf{r}_{mn}^{(j)}}{r_{mn}^5}\right), i,j = 1,2,3.\end{aligned}$$

These two formulas will be applied in the rewritten process.

$$\begin{aligned}\mathbf{D}_{ij}(\mathbf{x}_m,\mathbf{x}_n) &= C_1\left(\frac{\delta_{ij}}{r_{mn}} + \frac{\mathbf{r}_{mn}^{(i)}\mathbf{r}_{mn}^{(j)}}{r_{mn}^3}\right) + C_2\left(\frac{\delta_{ij}}{r_{mn}^3} - 3\frac{\mathbf{r}_{mn}^{(i)}\mathbf{r}_{mn}^{(j)}}{r_{mn}^5}\right) \\ &= C_1\left(\frac{\delta_{ij}}{r_{mn}} - \mathbf{r}_{mn}^{(j)}\frac{\partial}{\partial \mathbf{x}_m^{(i)}}\frac{1}{r_{mn}}\right) + C_2\frac{\partial}{\partial \mathbf{x}_m^{(i)}}\frac{\mathbf{r}_{mn}^{(j)}}{r_{mn}^3} \\ &= C_1\left(\frac{\delta_{ij}}{r_{mn}} - \mathbf{x}_m^{(j)}\frac{\partial}{\partial \mathbf{x}_m^{(i)}}\frac{1}{r_{mn}} + \frac{\partial}{\partial \mathbf{x}_m^{(i)}}\frac{\mathbf{x}_n^{(j)}}{r_{mn}}\right) - C_2\frac{\partial^2}{\partial \mathbf{x}_m^{(i)}\partial \mathbf{x}_m^{(j)}}\frac{1}{r_{mn}}.\end{aligned}$$

Here, $C_1 = \frac{k_BT}{8\pi\eta}, C_2 = \frac{k_BTa^2}{12\pi\eta}$. $\mathbf{D}_{ij}(\mathbf{x}_m,\mathbf{x}_n)$ is the element on ith row and jth column of the 3×3 matrix $\mathbf{D}(\mathbf{x}_m,\mathbf{x}_n)$, $i,j = 1,2,3, m,n = 1,\ldots,N$.

According to Equation (3.3) and Equation3.13, the ith element of the far-field part of the

Rotne-Prager-Yamakawa potential for the particle m is

$$\begin{aligned}\mathbf{U}^{(i)}_{far}(\mathbf{x}_m) &= \sum_{n\in\Omega_{far}} \sum_{j=1}^{3} \mathbf{D}_{ij}(\mathbf{x_m}, \mathbf{x_n})\mathbf{F}^{(j)}_n, \\ &= C_1 \sum_{\mathbf{x_n}\in\Omega_{far}} \frac{\mathbf{F}^{(i)}_n}{r_{mn}} - C_1 \sum_{j=1}^{3} \mathbf{x}^{(\mathbf{j})}_{\mathbf{m}} \frac{\partial}{\partial \mathbf{x}^{(\mathbf{i})}_{\mathbf{m}}} \left(\sum_{\mathbf{x_n}\in\Omega_{far}} \frac{\mathbf{F}^{(j)}_n}{r_{mn}} \right) \\ &+ \frac{\partial}{\partial \mathbf{x}^{(\mathbf{i})}_{\mathbf{m}}} \left(\sum_{\mathbf{x_n}\in\Omega_{far}} \frac{C_1(\mathbf{x_n}\cdot\mathbf{F_n})}{r_{mn}} \right) - C_2 \sum_{j=1}^{3} \frac{\partial^2}{\partial \mathbf{x}^{(\mathbf{i})}_{\mathbf{m}} \partial \mathbf{x}^{(\mathbf{j})}_{\mathbf{m}}} \left(\sum_{\mathbf{x_n}\in\Omega_{far}} \frac{\mathbf{F}^{(j)}_n}{r_{mn}} \right).\end{aligned}$$

In the following, define the far-field harmonic potential

$$\Phi_{far}(\mathbf{x}_m, q_m) = \sum_{\mathbf{x}_n\in\Omega_{far}} \frac{q_n}{r_{mn}}, \tag{3.5}$$

the far-field part of the electric field as

$$E_{far}(\mathbf{x}_m, q_m) = -\sum_{\mathbf{x}_n\in\Omega_{far}} q_n \frac{\mathbf{r}_{mn}}{r^3_{mn}}, \tag{3.6}$$

and the far-field part of the electric field gradient for particle m as

$$\ddot{\mathbf{\Phi}}_{far}(\mathbf{x}_m, \mathbf{x}_n) = -\begin{bmatrix} \frac{\partial^2\Phi_{far}}{\partial^2 x_1} & \frac{\partial^2\Phi_{far}}{\partial x_1 \partial x_2} & \frac{\partial^2\Phi_{far}}{\partial x_1 \partial x_3} \\ \frac{\partial^2\Phi_{far}}{\partial x_2 \partial x_1} & \frac{\partial^2\Phi_{far}}{\partial^2 x_2} & \frac{\partial^2\Phi_{far}}{\partial x_2 \partial x_3} \\ \frac{\partial^2\Phi_{far}}{\partial x_3 \partial x_1} & \frac{\partial^2\Phi_{far}}{\partial x_3 \partial x_2} & \frac{\partial^2\Phi_{far}}{\partial^2 x_3} \end{bmatrix}. \tag{3.7}$$

$E^{(i)}_{far}$ represents the *ith* element of the field vector and $\ddot{\mathbf{\Phi}}_{far}$. $\ddot{\mathbf{\Phi}}^{(ij)}_{far}$ denotes the element on the *ith* row and *jth* column of the electric field gradient matrix. Eventually, there is

$$\begin{aligned}\mathbf{U}^{(i)}_{far} \quad &= \quad C_1\Phi_{far}(\mathbf{x}_m, \mathbf{F}^{(i)}_n) + C_1 \sum_{j=1}^{3} \mathbf{x}^{(j)}_m E_{far}(\mathbf{x}_m, \mathbf{F}^{(j)}_n) \\ &\qquad -E^{(i)}_{far}(\mathbf{x}_m, C_1(\mathbf{x_n}\cdot\mathbf{F_n})) + C_2 \sum_{j=1}^{3} \ddot{\mathbf{\Phi}}^{(ij)}_{far}(\mathbf{x}_m, \mathbf{F}^{(j)}_n), i = 1, 2, 3.\end{aligned} \tag{3.8}$$

So to evaluate the far-field part of the RPY potential, firstly build the adaptive oct tree according to the particle positions $\{\mathbf{x}_i\}_{i=1}^N$. Then one needs call four times far-field FMM subroutines with charges

$\{\mathbf{F}_n^{(1)}\}_{n=1}^N, \{\mathbf{F}_n^{(2)}\}_{n=1}^N, \{\mathbf{F}_n^{(3)}\}_{n=1}^N, \{C_1(\mathbf{x_n} \cdot \mathbf{F_n})\}_{n=1}^N$ separately. Note here, only the contribution from the far-field part of the harmonic potential will be collected Φ_{far}, field E_{far} and electric field gradients $\ddot{\Phi}_{far}$.

3.1.2 Near-field Rotne-Prager-Yamakawa Potential Evaluation

For the near-field part of the RPY potential, follow Equation 3.1 and compute $\mathbf{U}_{near}$ particle by particle in a direct way. Define $C_0 = \frac{k_B T}{6\pi\eta a}$, when $m = n$, according to Equation 3.1,

$$\mathbf{D}(\mathbf{x}_m, \mathbf{x}_m)\mathbf{F}_m = C_0\mathbf{F}_m. \tag{3.9}$$

When $r_{mn} = |\mathbf{x}_m - \mathbf{x}_n| > 2a$,

$$\mathbf{D}(\mathbf{x}_m, \mathbf{x}_n)\mathbf{F}_n = \frac{C_1}{r_{mn}}\mathbf{F}_n + C_1 G(\mathbf{x}_m, \mathbf{x}_n)\mathbf{F}_n + \frac{C_2}{r_{mn}^3}\mathbf{F}_n - \frac{3C_2}{r_{mn}^2} G(\mathbf{x}_m, \mathbf{x}_n)\mathbf{F}_n, \tag{3.10}$$

where the matrix $G(\mathbf{x}_m, \mathbf{x}_n)$ is

$$G(\mathbf{x}_m, \mathbf{x}_n) = \begin{bmatrix} \frac{\mathbf{r}_{mn}^{(1)}\mathbf{r}_{mn}^{(1)}}{r_{mn}^3} & \frac{\mathbf{r}_{mn}^{(1)}\mathbf{r}_{mn}^{(2)}}{r_{mn}^3} & \frac{\mathbf{r}_{mn}^{(1)}\mathbf{r}_{mn}^{(3)}}{r_{mn}^3} \\ \frac{\mathbf{r}_{mn}^{(1)}\mathbf{r}_{mn}^{(2)}}{r_{mn}^3} & \frac{\mathbf{r}_{mn}^{(2)}\mathbf{r}_{mn}^{(2)}}{r_{mn}^3} & \frac{\mathbf{r}_{mn}^{(2)}\mathbf{r}_{mn}^{(3)}}{r_{mn}^3} \\ \frac{\mathbf{r}_{mn}^{(1)}\mathbf{r}_{mn}^{(3)}}{r_{mn}^3} & \frac{\mathbf{r}_{mn}^{(2)}\mathbf{r}_{mn}^{(3)}}{r_{mn}^3} & \frac{\mathbf{r}_{mn}^{(3)}\mathbf{r}_{mn}^{(3)}}{r_{mn}^3} \end{bmatrix} \tag{3.11}$$

and $\mathbf{r}_{mn} = \mathbf{x}_m - \mathbf{x}_n = (\mathbf{r}_{mn}^{(1)}, \mathbf{r}_{mn}^{(2)}, \mathbf{r}_{mn}^{(3)})$. So the near field potential is

$$\begin{aligned} \mathbf{U}_{near}(\mathbf{x}_m) &= \sum_{\mathbf{x_n} \in \Omega_{near}} \mathbf{D}(\mathbf{x_m}, \mathbf{x_n})\mathbf{F_n} \\ &= C_0\mathbf{F}_m + \sum_{\mathbf{x}_n (n \neq m) \in \Omega_{near}} \mathbf{D}(\mathbf{x_m}, \mathbf{x_n})\mathbf{F_n}. \end{aligned} \tag{3.12}$$

To summarize, the RPY potential for spherical particles with radius a at $\mathbf{x}_m$

$$\mathbf{U}(\mathbf{x}_m) = \mathbf{U}_{near}(\mathbf{x}_m) + \mathbf{U}_{far}(\mathbf{x}_m).$$

$\mathbf{U}_{near}(\mathbf{x}_m)$ will be computed directly by Equation (3.12), while $\mathbf{U}_{far}(\mathbf{x}_m)$ will be evaluated by Equation (3.8) through calling the far-field part of the new version of Laplace FMM described by

Algorithm 1.

3.2 Adaptive Rotne-Prager-Yamakawa Algorithm with Multipole Methods

The algorithm below is the pseudo code for the adaptive, fast RPY algorithm by the new version of fast multipole methods. The adaptive oct-tree structure from the new version of fast multipole methods will still be applied in evaluating the RPY potential. The far-field evaluation of electrostatic potentials, fields, and field gradient will follow the adaptive new version of fast multipole method discussed in Chapter 2. Interested readers can refer to Algorithm 1 for details. This adaptive RPY algorithm is essential for the Brownian dynamic simulations with hydrodynamic interactions.

Algorithm 2 ADAPTIVE RPY ALGORITHM WITH MULTIPOLE METHODS

```
Input : $N$ Particles with locations $\{\mathbf{x}_m\}_{m=1}^N$ and external forces on each particle $\{\mathbf{F}_m\}_{m=1}^N$.
Precision $\epsilon$ and the maximum number of particles in leaf box $s$, Boltzmann constant $k_B$,
temperature $T$, fluid viscosity $\eta$, radius of particle $a$
Output : RPY Potential $\{\mathbf{U}(\mathbf{x}_m)\}_{m=1}^N$.
Choose $pth$ order of the multipole expansion according to $\epsilon$.
Generate the adaptive Oct-Tree structure and obtain Lists $U_b, V_b, W_b, X_b$ for each box $b$.
for $j = 1, 2, 3$ do
    Call the far-file part of Algorithm 1 with charges $\{\mathbf{F}_n^{(j)}\}_{n=1}^N$.
    Collect the harmonic potentials $\Phi_{far}$, fields $E_{far}$ and electric field gradients $\ddot{\Phi}_{far}$.
    Add the corresponding part to $\mathbf{U}_{far}$ by Equation 3.3.
end for
Call the far-field part of Algorithm 1 with charges $\{C_1(\mathbf{x_n} \cdot \mathbf{F_n})\}_{n=1}^N$, collecting the fields $E_{far}$
    and add it to $\mathbf{U}_{far}$.
Compute the near-field RPY potential $\mathbf{U}_{near}$ at each point of the childless box $b$ directly for all
    charges in $U_b$ following Equation (3.1).
```

Note here, since there is only concurrent read of the adaptive data structures, the four far-field FMM evaluations and the near-field calculation can be finished simultaneously by different threads. Here, the RPY solver is parallelized on multi-core systems.

3.3 Numerical Results

In this chapter, the numerical results of the fast RPY solver will be presented by the new version of the multipole method [1]. There are two sets of numerical experiments. One is for spherical particles randomly distributed on the surface of a cylinder, and the other set is for spherical particles randomly distributed on eight faces. The numerical tests were performed with 3-digit, 6-digit, and 9-digit accuracy for spherical Brownian particles randomly distributed on the surface of cylinder and eight faces. The maximum number of particles allowed in each leaf box is set to be 80.

3.3.1 Cylinder

In the first set of experiments, the charges were distributed randomly on the surface of a cylinder. The running time and the numerical error results with 3-digit, 6-digit and 9-digit accuracy using 12 threads are presented in Table 3.1. Table 3.2 shows the running time of the fast Rotne-Prager-Yamakawa solver with different number of threads. The accuracy is 6-digit. Table 3.3 shows the parallel efficiency for particles randomly distributed on the surface of the cylinder. Figure 3.3.1 shows the speedup of fast RPY solver for particles distributed on the surface of cylinder.

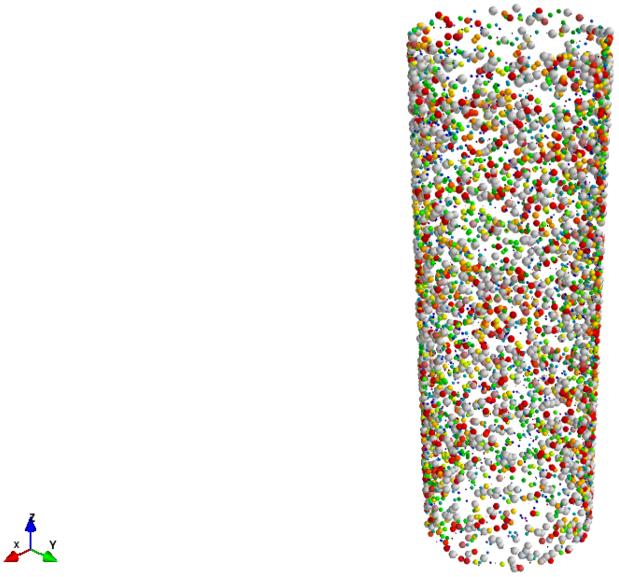

Figure 3.1: Particles randomly distributed on the surface of a cylinder

Table 3.1: Numerical results of RPY solver for particles randomly distributed on the surface of the cylinder(12 threads)

N	E_{3digit}	E_{6digit}	E_{9digit}	$T_{rpy-3digit}$	$T_{rpy-6digit}$	$T_{rpy-9digit}$	T_{dir}
10000	2.3e-5	2.3e-8	1.3e-10	0.06s	0.14s	0.31s	0.50s
20000	4.0e-5	4.0e-8	2.0e-10	0.14s	0.34s	0.77s	2.01s
50000	7.6e-5	7.6e-8	2.8e-10	0.34s	0.87s	1.88s	12.58s
100000	3.7e-5	3.8e-8	2.1e-10	0.65s	1.69s	3.49	50.36s
200000	7.0e-5	6.2e-8	2.9e-10	1.34s	3.44s	6.94	201.56s
500000	8.1e-5	8.0e-8	3.4e-10	3.28s	7.27s	15.0s	1155.48s
1000000	6.3e-4	7.5e-7	1.4e-9	8.01s	19.4s	39.1s	4966.99s

Table 3.2: Running time of RPY solver of 6-digit accuracy for particles randomly distributed on the surface of the cylinder

N	1 thread	2 threads	4 threads	8 threads	12 threads
10000	1.82s	0.92s	0.48s	0.26s	0.19s
20000	3.69s	1.87s	0.98s	0.51s	0.36s
50000	9.20s	4.76s	2.43s	1.26s	0.85s
100000	19.32s	9.80s	4.99s	2.63s	1.84s
200000	41.80s	21.21s	10.97s	5.54s	3.84s
500000	101.79	51.44s	26.30s	13.56s	9.25s
1000000	212.10	106.48	54.93	28.31s	19.21s

Table 3.3: Speedup of RPY solver of 6-digit accuracy for particles randomly distributed on the surface of the cylinder

N	1 thread	2 threads	4 threads	8 threads	12 threads
10000	1	1.96	3.79	7.03	9.68
20000	1	1.98	3.76	7.29	10.34
50000	1	1.94	3.79	7.33	10.81
100000	1	1.97	3.88	7.35	10.50
200000	1	1.97	3.81	7.54	10.90
500000	1	1.98	3.87	7.51	11.01
1000000	1	1.99	3.86	7.49	11.04

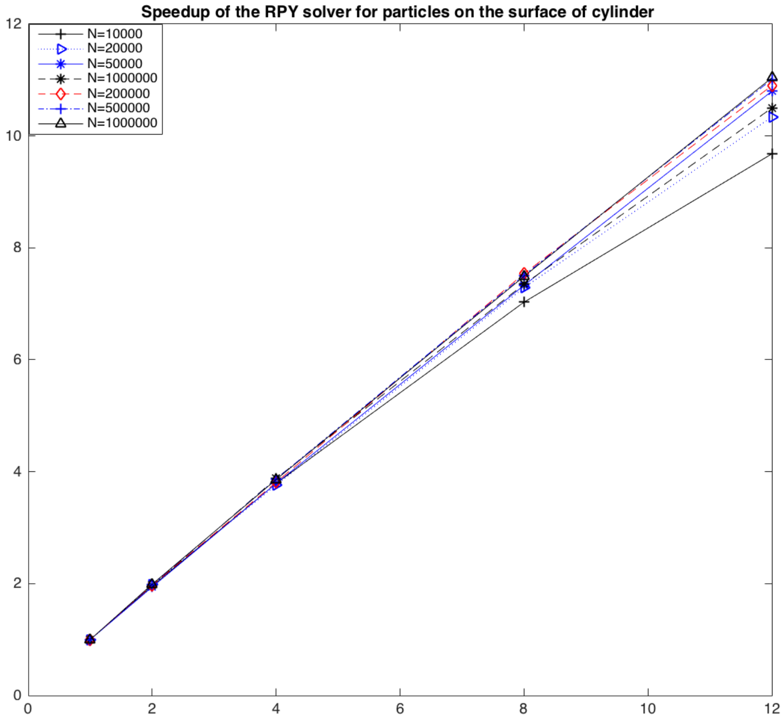

Figure 3.2: Speedup of particles randomly distributed on the surface of the cylinder

3.3.2 Eight Surface

In the second set of experiments, the charges were distributed randomly on an eight surface. The running time and the numerical error results with 3-digit, 6-digit and 9-digit accuracy using 12 threads are presented in Table 3.4. Table 3.5 shows the running time of the fast RPY solver with different number of threads with 6-digit accuracy. Table 3.6 shows the parallel efficiency for particles randomly distributed on the eight surface.

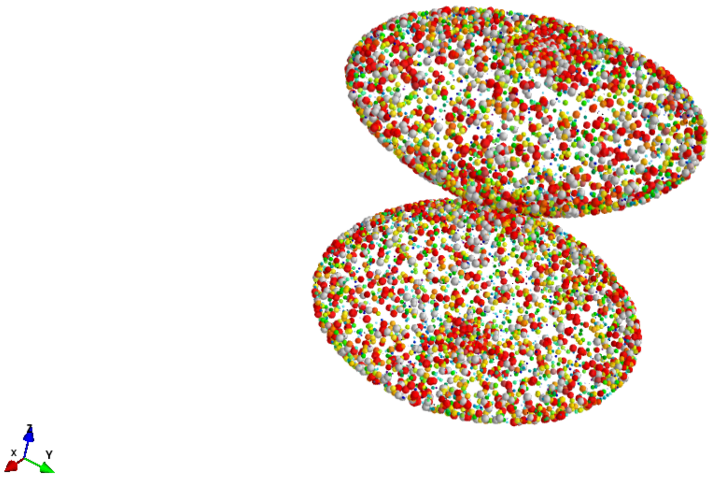

Figure 3.3: Particles randomly distributed on an eight surface

Table 3.4: Numerical Results of RPY solver for particles randomly distributed on the eight surface (12threads)

N	E_{3digit}	E_{6digit}	E_{9digit}	$T_{rpy-3digit}$	$T_{rpy-6digit}$	$T_{rpy-9digit}$	T_{dir}
10000	6.9e-6	1.0e-8	8.2e-12	0.10s	0.23s	0.48s	0.50s
20000	1.1e-5	1.7e-8	1.3e-11	0.23s	0.52s	1.00s	2.02s
50000	2.2e-5	3.3e-8	2.7e-11	0.61s	1.18s	2.34s	12.58s
100000	3.5e-5	5.3e-8	4.4e-11	1.18s	2.53s	4.90s	50.36s
200000	5.7e-5	8.4e-8	7.2e-11	1.82s	3.62s	9.09s	183.38s
500000	1.0e-4	1.6e-7	1.3e-10	5.32s	11.72s	22.83s	1268.32s
1000000	1.6e-4	2.4e-7	2.0e-10	10.73s	19.15s	44.8s	4973.18s

The numerical results show that the fast RPY solver developed here can handle multi-million

Table 3.5: Running time of RPY solver of 6-digit accuracy using different threads for particles randomly distributed on the eight surface

N	1 thread	2 threads	4 threads	8 threads	12 threads
10000	2.39s	0.95s	0.62s	0.27s	0.23s
20000	5.39s	2.20s	1.41s	0.63s	0.52s
50000	12.73s	5.53s	2.72s	1.45s	1.18s
100000	27.25s	11.46s	6.95s	3.23s	2.53s
200000	48.73s	20.37s	12.26s	5.26s	3.62s
500000	108.13s	55.11s	28.13s	14.49s	9.89s
1000000	214.71s	107.18s	65.07s	28.06s	19.15s

Table 3.6: Speedup of RPY solver of 6-digit accuracy using different threads for particles randomly distributed on the surface of the eight surface

N	1 thread	2 threads	4 threads	8 threads	12 threads
10000	1	2.52	3.85	8.85	10.39
20000	1	2.45	3.82	8.56	10.37
50000	1	2.30	4.68	8.78	10.79
100000	1	2.38	3.92	8.44	10.77
200000	1	2.39	3.97	9.26	13.46
500000	1	1.96	3.84	7.46	10.93
1000000	1	2.00	3.30	7.65	11.21

spherical Brownian particle simulations and that the parallel solver has nearly linear speed up on multi-core systems. The fast parallel RPY solver with tunable accuracy makes it computationally viable for large-scale long-time Brownian dynamic simulations.

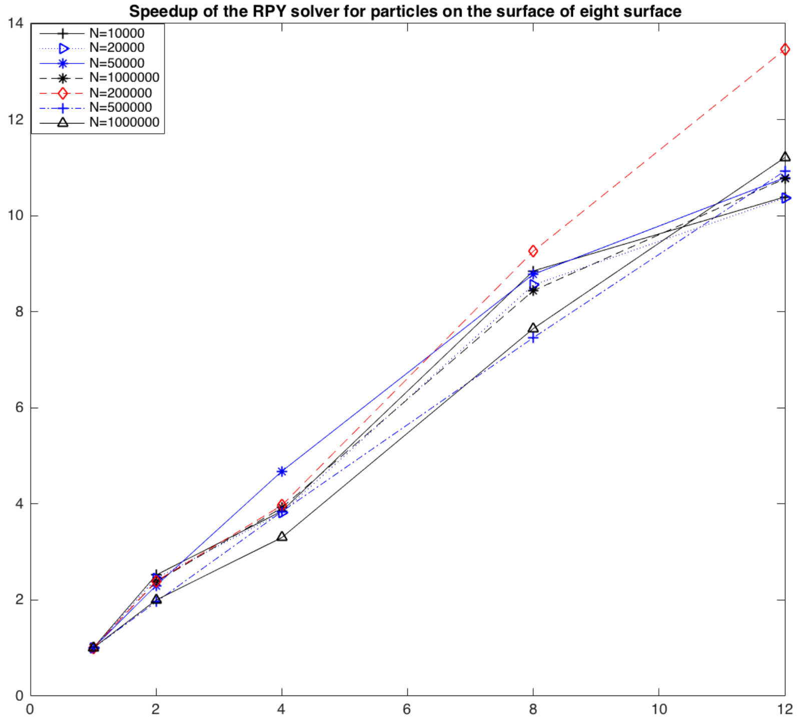

Figure 3.4: Speedup of particles randomly distributed on the eight surface

CHAPTER 4

Hydrodynamic Interactions of Complex, Rigid, Biological Macromolecules in Suspension

Hydrodynamic methods play a vital role in characterizing the structures of biological macromolecules and biomolecular complexes. In Chapter 3, the RPY tensor is employed to approximate the mobility of the spherical particles in fluids. Since the real biological systems are much more complicated, for example, there are boundaries or the immersed particles are not spheres, it is not possible to use the RPY tensor given above. Modeling the dynamics of the macromolecules can be very challenging. First, finding the associated diffusion matrix is fundamentally important in constructing the numerical algorithms for tracking the motion of the many-body macromolecules in viscous fluids. Also, the original Brownian-dynamics model by Ermak and McCammon only consider the translational motion of the spherically symmetric particles or particles composed of spherically symmetric subunits. For complex-structured macromolecules, they also execute rotational Brownian motion arising from the fluctuating torque by the surrounding solvent molecules. Generally speaking, biological macromolecules can be classified into rigid, flexible and semi-flexible types. Many structures, such as proteins and viruses, have rigid, well-defined geometries. For a rigid body, hydrodynamic friction only occurs on the surface of solid particles and the interior of the particle is inaccessible to solvent. The purpose of this chapter is to present an accurate model of the hydrodynamic effect on the translational and rotational dynamics in complex many-body systems with rigid macromolecules. Diffusion including translation and rotational dynamics has been widely studied in a wide range of environments including cytoplasm of cells [79], concentrated suspensions [80], gels or hydrogels [81, 82, 83, 84, 85, 86, 87], and mucus [88].

Earlier work in this field used macromolecules with simple geometric structures (such as spheres, ellipsoids, or cylinders) to describe the shape of fairly symmetric macromolecules in a low resolution. Nonetheless, these simple models are inadequate to accurately describe more complicated structures and cannot reveal the more complicated, unique conformation of macromolecules with higher

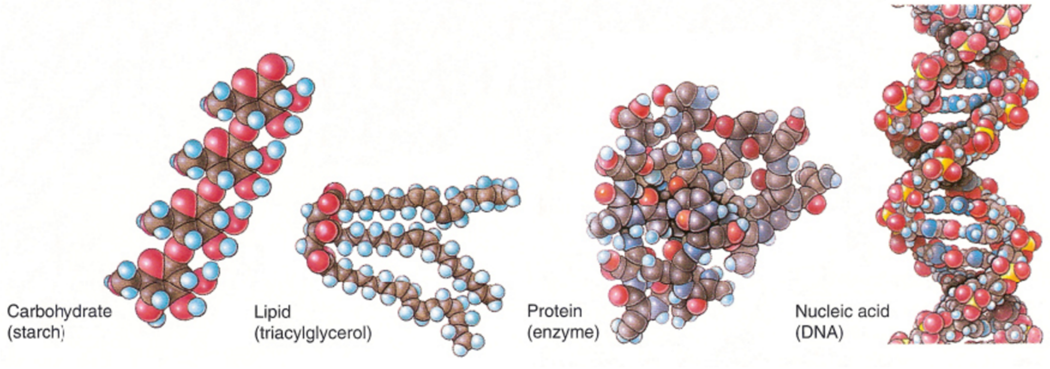

Figure 4.1: Complex biological macromolecule structures [4]

resolutions. To develop a more precise approach for studying the hydrodynamic effects of the macromolecules, Oseen and Burgers firstly took into account the hydrodynamic interactions between frictional subunits [89, 90]. Then Kirkwood modeled polymers as arrays of frictional centers of hydrodynamic resistance [46, 91] based on Oseen-Burgers formulation and developed the theory for the translational frictional coefficients. Those frictional centers can be viewed as subunits making up the macromolecular structure without considering the finite size of these subunits. The hydrodynamic forces are distributed on the subunit surface rather than the centers. Bloomfield advanced Kirkwood's theory by calculating the translational and rotational diffusion coefficients of complex structures in two ways. He developed the shell model in which the shape of a macromolecule has been outlined using an ensemble of identical, small spherical subunits(beads) [92]. Also, in the subunit model he developed, a macromolecule is built up using a finite number of spherical subunits with different size [47]. In subsequent works, McCammon,1976 [11]; Garcia de la Torre and Bloomfield, 1977, 1978 [93, 94, 95]; Swanson et 1978 [96], the approximation contained in Kirkwood's original bead model has been removed and the friction forces at each element were assumed to act at the bead center. The updated bead model has been widely used in Brownian dynamics simulation to approximate the complex structure of macromolecules by smaller and geometrically simpler frictional elements and compute the diffusion tensor between macromolecules.

In this work, a concise model has been built to describe the hydrodynamic interactions between the macromolecules with arbitrary shape. The theory and methods for calculating the hydrodynamic properties between the macromolecules will be shown, including approximate methods that may be needed to treat models with a very large number of elements. Two algorithms have been developed for computing the translational and rotational dynamics in many-body complex systems. A preconditioner has been devised by employing the rigid property of the macromolecules to accelerate the convergence of the algorithms. All of these hydrodynamic theories can be used to calculate the frictional properties of rigid macromolecules more accurately from the structures and provide guidelines for computational tools for the bead modeling.

4.1 Hydrodynamic Interactions

The effects of hydrodynamic interactions mediated by the host solvent are included through a position-dependent inter-particle friction tensor. A commonly used model for describing the hydrodynamic interactions in colloidal suspensions and polymer solutions is the Rotne-Prager-Yamakawa(RPY) tensor. It has been applied to dilute solutions of large molecules and colloidal particles. For symmetric spheres in Stokes flow, the RPY tensor will neglect the hydrodynamic rotation-rotation and rotation-translation coupling and is defined as following:

$$\mathbf{D}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \frac{k_B T}{6\pi\eta a}\mathbf{I} & i = j \\ \frac{k_B T}{8\pi\eta r_{ij}}\left[(\mathbf{I} + \frac{\mathbf{r_{ij}}\otimes\mathbf{r_{ij}}}{r_{ij}^2}) + \frac{2a^2}{3r_{ij}^2}(\mathbf{I} - 3\frac{\mathbf{r_{ij}}\otimes\mathbf{r_{ij}}}{r_{ij}^2})\right] & i \neq j \text{ and } r_{ij} > 2a, \\ \frac{k_B T}{6\pi\eta a}\left[(1 - \frac{9}{32}\frac{r_{ij}}{a})\mathbf{I} + \frac{3}{32a}\frac{\mathbf{r_{ij}}\otimes\mathbf{r_{ij}}}{r_{ij}}\right] & i \neq j \text{ and } r_{ij} \leq 2a. \end{cases}$$

Here, k_B is the Boltzmann constant, T is the absolute temperature, η is the solvent viscosity, a is the radius of each bead. i, j represents the indices of the particles. r_{ij} is the distance between the particle i and particle j. $\mathbf{r_{ij}} = \mathbf{x_i} - \mathbf{x_j}$. $\mathbf{I}$ is the 3×3 identity matrix.

4.2 Bead-Shell Model

Earlier studies distinguish the bead model and the shell model. Bead models are methods for representing a complex-structured particle with a few beads (either identical or different in size) to approximately occupy the volume of the particle. Alternatively, in the shell model, a great number of small, identical, spherical subunits are arranged to outline the overall particle shape. Since the hydrodynamic forces are distributed on the subunit surfaces, the shell model is more appropriate for hydrodynamics. Also, if higher resolution is required, the fine details of the macromolecular

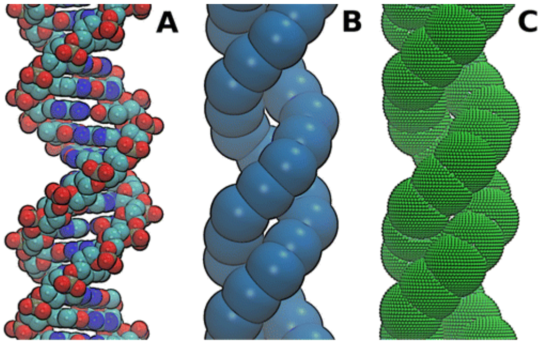

Figure 4.2: (A) Atomistic DNA model (B) Bead Model with each DNA residue modeled with a single pseudoatom (C) Bead-shell model with surfaces covered with spherical elements(By Maciej Dlugosz [5])

structure can be adequately modeled by decreasing the size of the beads in the shell model.

Calculating the diffusion tensor of the macromolecules is essential in the Brownian simulation of macromolecules. The bead model has been applied to reproduce the experimental diffusion coefficients of a single macromolecule analytically [97, 98], where the macromolecule is set to be at least partly flexible. In the Brownian dynamic simulation of one macromolecule, those beads can be viewed as a chemical unit, such as an atom, amino acid or even a protein domain. Only the bead-level diffusion coefficients are computed. For complex-structured macromolecules in many-body systems, it is difficult to obtain the diffusion tensor by the bead model analytically when the macromolecules are modeled as rigid bodies. Numerical methods are constantly applied to obtain the approximation. In 2012, Wang [99] showed that the bead models produce very accurate diffusion tensors for two spheres where beads have no overlap. Here, the bead-shell model is applied to study the hydrodynamic effects for the macromolecules for the many-body system. Note, typical distances between beads are larger than bead sizes. The idea is first to compute the bead-level diffusion tensor, then convert the bead-level diffusion tensor into the diffusion tensor of the macromolecules.

4.3 Mathematical Formulation of the many-body System in Suspensions with Bead Model

Consider a system of m rigid bodies with arbitrary shape suspended in an unbounded viscous fluid at low Reynolds number with body k having N_k beads on its surface. Then the number of beads for the whole system is $N = \sum_{k=1}^{m} N_k$. Given the systematic(non-hydrodynamic) body forces/torques $\vec{F}$ of the whole system, how does one compute the general velocities $\vec{V}$ of the bodies? Here, $\vec{V}$ is the translational/angular velocities of all bodies and $\vec{F}$ is forces/torques of all bodies.

$$\vec{V} = [V_1, \Omega_1, \ldots, V_m, \Omega_m]^T, \vec{F} = [F_1, \tau_1, \ldots, F_m, \tau_m]^T \tag{4.1}$$

Suppose $\vec{\mathbf{v}}_b$ are the velocities of all beads of the whole system, $\mathbf{v}_k$ are the bead velocities of Body k. There is

$$\mathbf{v}_k = [v_k^1, \ldots, v_k^{N_k}]^T, \vec{\mathbf{v}}_b = [\mathbf{v}_1, \ldots, \mathbf{v}_m]^T \tag{4.2}$$

Similarly, the bead forces of Body k $\mathbf{f}_k$ and bead forces for all beads of the whole system are:

$$\mathbf{f}_k = [f_k^1, \ldots, f_k^{N_k}]^T, \vec{\mathbf{f}}_b = [\mathbf{f}_1, \ldots, \mathbf{f}_m]^T. \tag{4.3}$$

In Stokes flow, by omitting the sign, the bead velocities and the bead forces have

$$\vec{\mathbf{f}}_b = \boldsymbol{\zeta} \vec{\mathbf{v}}_b = k_B T \mathbf{D}^{-1} \vec{\mathbf{v}}_b, \tag{4.4}$$

where $\boldsymbol{\zeta}$ is the frictional tensor between the beads. $\mathbf{D}$ is the diffusion tensor describing the hydrodynamic interactions between beads and modeled by RPY tensor. To study the hydrodynamic interactions between the macromolecules with the shape described by the bead-shell model, one needs to compute the grand diffusion matrix. One also wants to know the relationship between the force/torque and the translational/angular velocities on the body level. So if one can find a relationship between

- Bead forces - Body force/torque
- Bead velocities - Body translational/angular velocity

according to Equation (4.4), the conclusion can be easily obtained. The problem becomes how to derive a formula between the translational/angular velocities and the forces/torques *on body level*?

Theorem 4.3.1. *For a many-body system with m macromoleclues, for any $k = 1, \ldots, m$, the number of beads on the kth macromolecule is N_k. Assume $\vec{\mathbf{V}}$ denotes the general velocities including translational velocities*

$$\vec{\mathbf{V}} = (V_1, \Omega_1, \cdots, V_m, \Omega_m)^T,$$

where V_i, Ω_i represent the translational velocity and angular velocity of macromolecule i separately. The general body forces $\vec{\mathbf{F}}$

$$\vec{\mathbf{F}} = (F_1, \tau_1, \cdots, F_m, \tau_m)^T,$$

is a composite of non-hydrodynamic forces and torques of all macromolecules. F_i is the force and τ_i is the torque. $\mathbf{D}$ *is the diffusion tensor between beads. Then the general forces and the general velocities of macromolecules have*

$$k_B T(\vec{Q}\mathbf{D}^{-1}\vec{Q}^T)\vec{\mathbf{V}} = \vec{\mathbf{F}} \tag{4.5}$$

Here, $\mathbf{D} \in R^{3N\times 3N}, \vec{\mathbf{Q}} \in R^{6m\times 3N}$ *and*

$$\vec{Q} = diag(Q_1, \ldots, Q_m), Q_k = \begin{pmatrix} I & I & \ldots & I \\ A_k^{(1)} & A_k^{(2)} & \ldots & A_k^{(N_k)} \end{pmatrix}_{6\times 3_{N_k}}.$$

So the grand frictional tensor *of the macromolecules in the many-body system is*

$$\vec{\boldsymbol{\zeta}} = k_B T(\vec{Q}\mathbf{D}^{-1}\vec{Q}^T). \tag{4.6}$$

And the corresponding grand diffusion tensor *of the macromolecules is*

$$\vec{\mathbf{D}} = (\vec{Q}\mathbf{D}^{-1}\vec{Q}^T)^{-1}. \tag{4.7}$$

Proof. To simplify, each macromolecule will be treated as a body. For body $k = 1, \ldots, m$, the body force is a summation of all bead forces on body k. There is

$$\begin{pmatrix} F_k \\ \tau_k \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N f_k^i \\ \sum_{i=1}^N r_k \times f_k^i \end{pmatrix} = \overbrace{\begin{pmatrix} I & I & \ldots & I \\ A_k^{(1)} & A_k^{(2)} & \ldots & A_k^{(N_k)} \end{pmatrix}}^{Q_k \in R^{6\times 3N_k}} \mathbf{f}_k$$

Here, $A_k^{(i)} = -(A_k^{(i)})^T$ is dependent on the relative position of bead i on body k. $\mathbf{r}_k^{(i)} =$

$(\mathbf{r}_k^{(i,1)}, \mathbf{r}_k^{(i,2)}, \mathbf{r}_k^{(i,3)})$ is the relative position of bead i on body k to its center.

$$A_k^{(i)} = \begin{pmatrix} 0 & -\mathbf{r}_k^{(i,3)} & \mathbf{r}_k^{(i,2)} \\ \mathbf{r}_k^{(i,3)} & 0 & -\mathbf{r}_k^{(i,1)} \\ -\mathbf{r}_k^{(i,2)} & \mathbf{r}_k^{(i,1)} & 0 \end{pmatrix} \tag{4.8}$$

Note, here $A_k^{(i)} = -{A_k^{(i)}}^T$.

The total force/torque for the whole system can be written as

$$\vec{\mathbf{F}} = \underbrace{\begin{pmatrix} F_1 \\ \tau_1 \\ \vdots \\ F_m \\ \tau_m \end{pmatrix}}_{6m} = \underbrace{\begin{pmatrix} Q_1 & & \\ & \ddots & \\ & & Q_m \end{pmatrix}}_{\vec{Q} \in R^{6m \times 3N}} \underbrace{\begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_M \end{pmatrix}}_{3N} = \vec{Q}\vec{\mathbf{f}}_b.$$

Similarly, the bead velocities of body k can be described as

$$\mathbf{v}_k = \begin{pmatrix} V_k + \Omega_k \times \mathbf{r}_1 \\ \vdots \\ V_k + \Omega_k \times \mathbf{r}_{N_k} \end{pmatrix} = \underbrace{\begin{pmatrix} I & -A_k^{(1)} \\ \vdots & \vdots \\ I & -A_k^{(N_k)} \end{pmatrix}}_{T_k = Q_k^T \in R^{3N_k \times 6}} \begin{pmatrix} V_k \\ \Omega_k \end{pmatrix}$$

Since $A_k^{(i)} = -{A_k^{(i)}}^T$, the following equation can be obtained

$$\vec{\mathbf{v}}_b = \underbrace{\begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_m \end{pmatrix}}_{3N} = \underbrace{\begin{pmatrix} Q_1^T & & \\ & \ddots & \\ & & Q_m^T \end{pmatrix}}_{\vec{Q}^T \in R^{3N \times 6m}} \underbrace{\begin{pmatrix} V_1 \\ \Omega_1 \\ \vdots \\ V_m \\ \Omega_m \end{pmatrix}}_{6m \times 1} = \vec{Q}^T \vec{\mathbf{V}}.$$

To summarize, the following equations have been discussed:

$$\begin{cases} \vec{\mathbf{v}}_b &= \frac{1}{k_BT}\mathbf{D}\vec{\mathbf{f}}_b \\ \vec{\mathbf{v}}_b &= \vec{Q}^T\vec{\mathbf{V}} \\ \vec{\mathbf{F}} &= \vec{Q}\vec{\mathbf{f}}_b \end{cases} \tag{4.9}$$

Rewrite the above formula, the conclusion of Theorem 4.3.1 can be obtained. □

The above theorem can be rewritten as

$$\vec{F} = \vec{\boldsymbol{\zeta}}\vec{V}. \tag{4.10}$$

or

$$\vec{V} = \frac{1}{k_BT}\vec{\mathbf{D}} \cdot \vec{F}. \tag{4.11}$$

Here, $\vec{\boldsymbol{\zeta}} = k_BT(\vec{Q}\mathbf{D}^{-1}\vec{Q}^T) \in \mathbb{R}^{6m\times 6m}$ and $\vec{\mathbf{D}} = (\vec{Q}\mathbf{D}^{-1}\vec{Q}^T)^{-1} \in \mathbb{R}^{m\times m}$. The diffusion matrix $\mathbf{D}$ is modeled by the RPY tensor for hydrodynamics between all beads of the macromolecules in the whole system. To obtain the general velocities of the macromolecules, one can either solve the linear equation of Equation (4.10) or compute the matrix-vector product of Equation (4.11). The difficult part of solving Equation (4.10) or Equation (4.11) is it is time-consuming to get $\vec{\boldsymbol{\zeta}}$ and $\vec{\mathbf{D}}$ explicitly since the $\mathbf{D}$ is a dense symmetric positive definite matrix with dimension $3N \times 3N$ and computing the inverse of $\mathbf{D}$ will result in polynomial time complexities.

4.4 Fast Algorithms for the many-body System

Since the friction tensor matrix $\vec{\boldsymbol{\zeta}} = k_BT(\vec{Q}\mathbf{D}^{-1}\vec{Q}^T) \in \mathbb{R}^{6m\times 6m}$, when the number of macromolecules m is small, say two, the friction tensor matrix will be only a 12 by 12 matrix. Because the diffusion matrix of all beads $\mathbf{D}$ is symmetric positive definite and the matrix $\vec{Q}$ is full rank, the friction matrix $\vec{\boldsymbol{\zeta}}$ is still symmetric positive definite. Once an approximation of $\vec{\boldsymbol{\zeta}}$ is achieved, the general velocities $\vec{\mathbf{V}}$ satisfying Equation (4.10) will be easily obtained.

Since the fast RPY solver has been developed in Chapter 2, the matrix-vector product evaluation $\mathbf{D} \cdot F$ can be efficiently obtained. Based on this property, two fast algorithms for solving the Equation

(4.10) have been devised. The first algorithm is based on the block conjugate gradient method for computing the approximation of the matrix $\vec{\boldsymbol{\zeta}}$, which is finding an approximation matrix B. This method works for a small number of macromolecules m. Note here, the approximation error ϵ does not have to be very small and accurate numerical solutions of the general velocities of the macromolecules can still be obtained.

4.4.1 Block Conjugate Gradient Method

Below is the pseudo code of the block conjugate gradient method for computing the approximation of the grand frictional matrix omitting the constant k_BT.

Algorithm 3 Block conjugate gradient method

```
Input : Bead Positions of all bodies $\{\mathbf{x_i}\}_{i=1}^{N}$, $maxiter$, $\epsilon$.
Output : $B \approx \vec{Q}\mathbf{D}^{-1}\vec{Q}^T$.
Let $R_0 = \vec{Q}$, $P_0 = R_0, B_0 = O$.
for  do$k = 0$ to $maxiter$
    Solve $(P_k^T D P_k)\alpha_k = R_k^T R_k$.
    $B_{k+1} = B_k + R_k^T R_k \alpha_k$
    $R_{k+1} = R_k - DP_k\alpha_k$
    if  then$||R_{k+1}|| < \epsilon$
        break;
    else
        Solve $\beta_k$ from $((R_k R_k^T))\beta_k = R_{k+1}^T R_{k+1}$;
        $P_{k+1} = R_{k+1} + P_k * \beta_k$ ;
    end if
end for
return $B_{k+1}$.
```

4.4.2 Numerical Results of the Block Conjugate Gradient Method

In this chapter, the hydrodynamics of spherical macromolecules with subunits on their surfaces will be considered. Many virus have such shapes, for example, the varicella-zoster virus-one of eight herpes viruses known to infect humans and vertebrates. The HIV virus and some nanoparticles also have the spherical shape and their hydrodynamics can be modeled by bead-shell model. Below, bead-shell model will be applied to study the two-body macromolecules of identical size in this simulation. The block conjugate gradient method is applied to obtain the approximation of the grand frictional matrix. Since the grand frictional matrix is only a 12×12 matrix, once obtaining its approximation, it is easy to compute the related linear equations. There are two advantages for applying the block conjugate gradient method in Brownian dynamics simulations for macromolecules. First, it is easy to obtain the general velocities of the macromolecules by solving the linear equation described by Equation (4.10), which means the displacements/orientations of the macromolecules due to the determinstic non-hydrodynamics forces operating on the macromolecules can be obtained by simply multiplying a Δt of the general velocities. Another advantage of this method is that the Brownian displacements and orientations are also easy to compute since they can be obtained by solving a linear equation $\sqrt{(\vec{Q}\mathbf{D}^{-1}\vec{Q}^T)}\vec{\mathbf{V}}_B = \vec{\mathbf{F}}_B$. The right hand side $\vec{\mathbf{F}}_B$ is obtained by white noises and coefficient matrix $\sqrt{(\vec{Q}\mathbf{D}^{-1}\vec{Q}^T)}$ is described by the square root of the grand frictional matrix. $\vec{\mathbf{V}}_B$ is the random displacements and orientations for the macromolecules. Some similar research for computing the approximation of block bilinear form $C^H A^{-1} B$ where square matrix A is large and sparse. B and C are rectangular matrices with the same size in [100]. But the matrix D here are very dense.

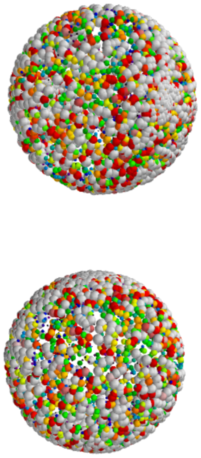

Figure 4.3: Bead-shell model of the two body system

Below is a figure about the iteration number and the numerical errors of the general velocities. As the iteration number is increased in the block conjugate gradient method, a more accurate grand frictional matrix will be obtained. As a result, the general velocities of the macromolecules are more accurate.

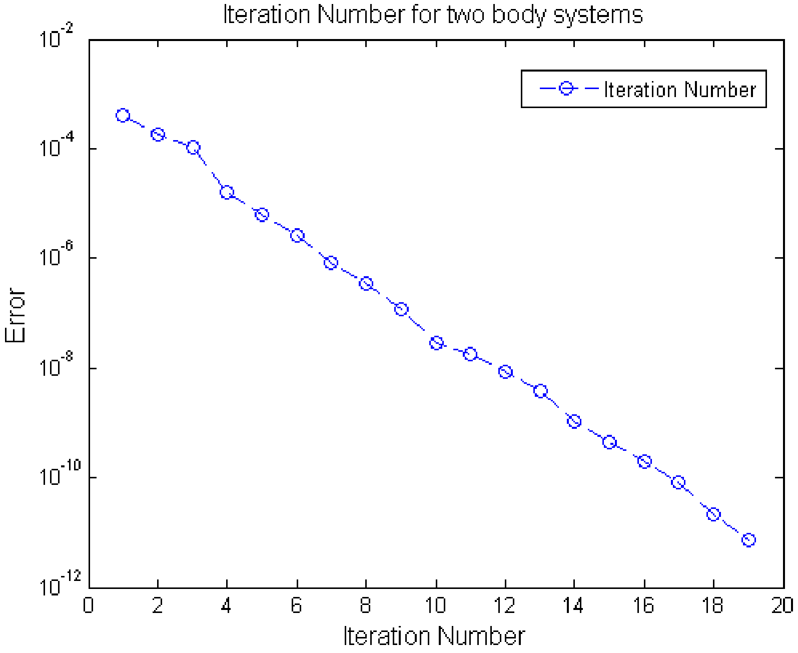

Figure 4.4: Iteration number of the block conjugate gradient method for the hydrodynamics of the two-body system

4.4.3 Schur Complement Method

Since $\vec{\boldsymbol{\zeta}} = k_B T(\vec{Q}\mathbf{D}^{-1}\vec{Q}^T)$, construct matrix

$$M = \begin{pmatrix} \mathbf{D} & -\vec{Q}^T \\ \vec{Q} & O \end{pmatrix}. \tag{4.12}$$

The Schur complement of $M/\mathbf{D} = O - \vec{Q}\mathbf{D}^{-1}\vec{Q}^T = -\frac{1}{k_B T}\vec{\boldsymbol{\zeta}}$.

Theorem 4.4.1. *Solving the general velocities of the macromolecules which satisfy*

$$\vec{\boldsymbol{\zeta}}\vec{\mathbf{V}} = \vec{\mathbf{F}}$$

is equivalent to

$$\vec{\boldsymbol{\zeta}}\vec{\mathbf{V}} = \vec{\mathbf{F}} \Longleftrightarrow M \begin{pmatrix} U \\ \vec{\mathbf{V}} \end{pmatrix} = \begin{pmatrix} O \\ \frac{1}{k_B T}\vec{\mathbf{F}} \end{pmatrix}.$$

Proof. Assume

$$M \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \Rightarrow \begin{cases} \mathbf{D}V_1 = \vec{Q}^T V_2 + F_1 \\ \vec{Q}V_1 = F_2. \end{cases}$$

Because $\mathbf{D}$ is symmetric positive definite, multiply $\vec{Q}\mathbf{D}^{-1}$ on both sides of the first equation above. The equation below can be obtained

$$\vec{Q}\mathbf{D}^{-1}V_1 = \vec{Q}\mathbf{D}^{-1}\vec{Q}^T V_2 + \vec{Q}\mathbf{D}^{-1}F_1 = F_2$$

Let $F_1 = 0$ and $F_2 = \frac{1}{k_B T}\vec{\mathbf{F}}$, the theorem 4.4.1 can be easily verified. □

Since the model described in this chapter is for rigid bodies, the conformation of the rigid bodies will not change along with time which means the relative positions of the beads on the surface of the same macromolecule are the same. Utilizing this characteristic, a preconditioner can be constructed since the inverse of the diffusion matrix for each rigid body, i.e. $\{\mathbf{D}_{ij}\}_{i,j=1}^{N_k}$, can be precomputed.

Given the two-body system,

$$M = \left(\begin{array}{cc|cc} \mathbf{D_{11}} & \mathbf{D_{12}} & -\vec{Q}_1^T & \\ \mathbf{D_{12}^T} & \mathbf{D_{22}} & & -\vec{Q}_2^T \\ \hline \vec{Q}_1 & & \multicolumn{2}{c}{\multirow{2}{*}{O}} \\ & \vec{Q}_2 & & \end{array}\right). \tag{4.13}$$

Let $P_1 = \vec{Q}_1 D_{11}^{-1} \vec{Q}_1^T$, $P_2 = \vec{Q}_2 D_{22}^{-1} \vec{Q}_2^T$, apply the inverse formula of the block matrix. The preconditioner will be obtained

$$M_0 = \left(\begin{array}{cc|cc} \mathbf{D_{11}^{-1}}(I - \vec{Q_1}^T P_1^{-1} \vec{Q}_1 \mathbf{D_{11}^{-1}}) & & \mathbf{D_{11}^{-1}} \vec{Q}_1^T P_1^{-1} & \\ & \mathbf{D_{22}^{-1}}(I - \vec{Q_2}^T P_2^{-1} \vec{Q}_2 \mathbf{D_{22}^{-1}}) & & \mathbf{D_{22}^{-1}} \vec{Q}_2^T P_2^{-1} \\ \hline -P_1^{-1} \vec{Q}_1^T \mathbf{D_{11}^{-1}} & & P_1^{-1} & \\ & -P_2^{-1} \vec{Q}_2^T \mathbf{D_{22}^{-1}} & & P_2^{-1}. \end{array}\right) \tag{4.14}$$

Similarly, the preconditioner for the many-body system can be devised.

4.4.4 Numerical Results of Schur Complement Method

In this section, some numerical results of the Schur complement method will be presented. Consider the spherical rigid bodies with different numbers of spherical subunits on their surface. The figures below show the iteration number of Schur complement method and the numerical errors. The comparison has been made between the Schur complement method with and without preconditioner for the hydrodynamics. The two-body system with different size is first considered. The numerical errors decrease as the iteration number is increased. The numerical errors of the method with preconditioners decrease much faster than the method without preconditioners. As the number of beads for each body increases, the iteration number for the Schur complement method will be increased. However, the iteration numbers for the method with preconditioners keep almost constant which means using preconditioners will keep the fast convergence rate even for larger macromolecules. To prove that, some numerical results will be presented for six-body system with different size. As shown below, the Schur complement method works for many-body macromolecules.

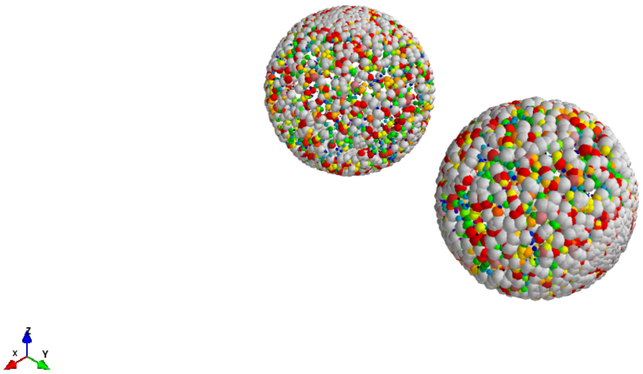

Figure 4.5: Bead-shell model of the two body system with different sizes

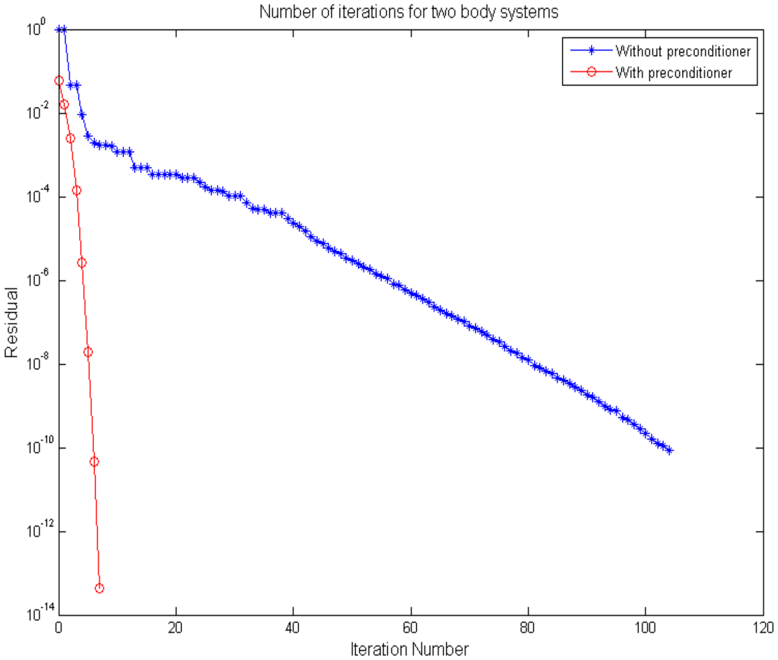

Figure 4.6: Iteration number of Schur complement method for the hydrodynamics of the two-body system with different sizes

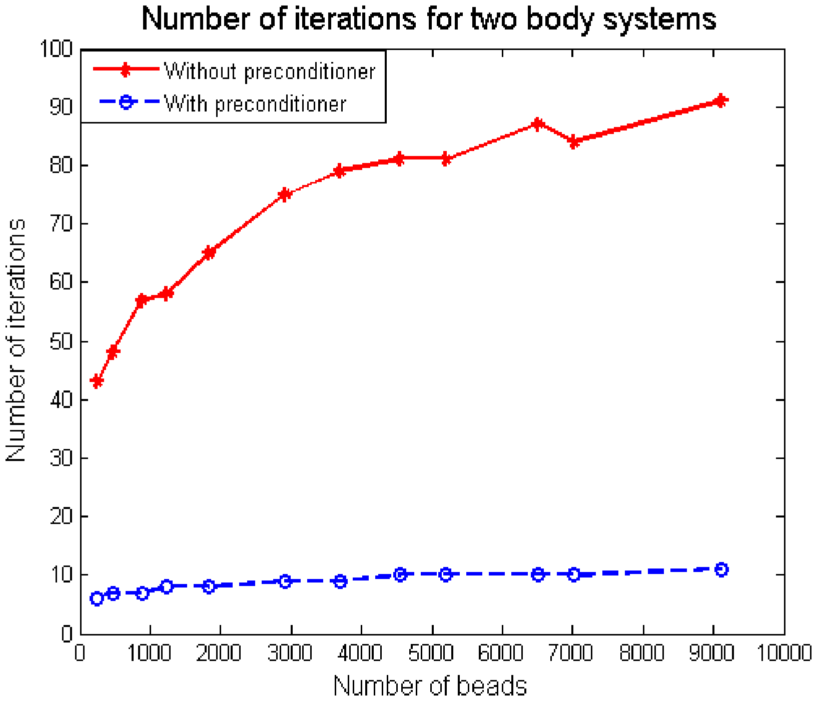

Figure 4.7: Comparison of iteration number of Schur complement method with and without preconditioner for the hydrodynamics of the two-body system with different sizes

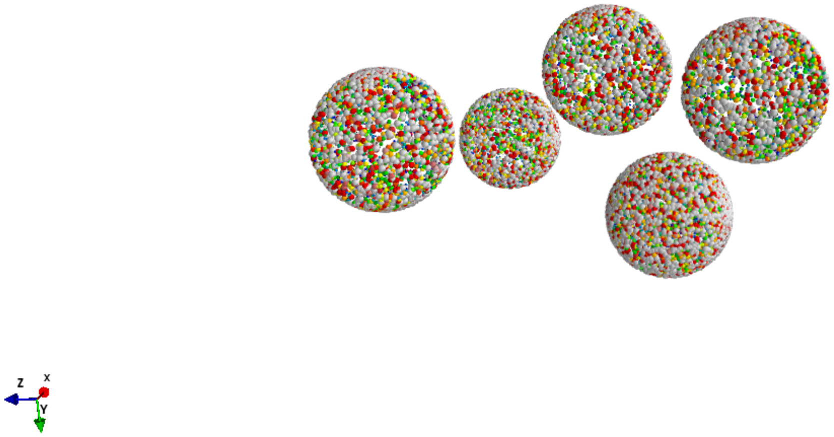

Figure 4.8: The bead-shell model of the many-body system with different sizes

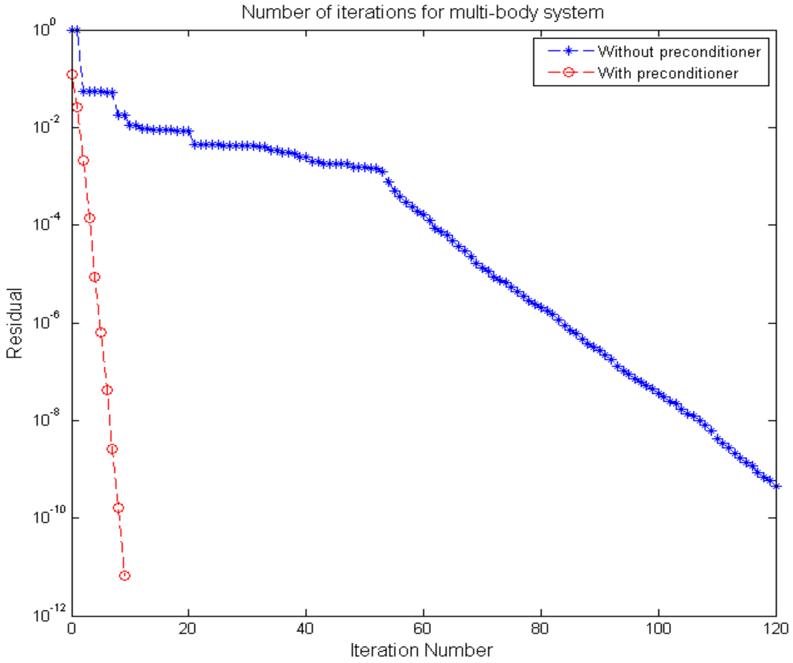

Figure 4.9: Comparison of iteration number of Schur complement method with and without preconditioner for the hydrodynamics of the two-body system with different sizes

CHAPTER 5

Conclusion

The subject of this dissertation is devising fundamental fast algorithms and developing novel computational solvers to advance large-scale, long-time Brownian dynamic simulations with hydrodynamics. The fundamental solvers developed in this dissertation show great promise in parallelization on modern multicore computers and are suitable for simulating a broad class of flows and physical scales. They also allow for further simulation of notable complex-structured macromolecules, as well as forcing terms.

The first part of the dissertation showed the derived formulas for computing the electric field gradients of the electrostatic interactions by the new version of fast multipole method [1]. The computation of the electric field gradients was implemented by the new version of fast multipole method as a new function for existing solvers. The fast evaluation of the electric field gradients is of importance in many applications, such as solid-state physics, biochemistry and depolarization of the nerve fiber in MRI scans.

In the second part of the dissertation, the Rotne-Prager-Yamakawa tensor was applied to describe the hydrodynamic effects of spherical bead particles. A parallelized Rotne-Prager-Yamakawa solver with tunable accuracy for multi-core systems has been developed to evaluate the matrix-vector multiplication between the Rotne-Prager-Yamakawa tensor and the non-hydrodynamic force of the particles suspended in fluids. Combining with the iterative Krylov subspace methods and the Chebyshev approximations, the random placement of the particles satisfying the fluctuation-dissipation theorem can be computed with $O(N)$ complexity. The solver is capable of handling simulations with millions of particles or beads. This ability makes it computationally attractive for large-scale Brownian dynamic simulations.

In the third part, the anisotropic diffusion of the complex, rigid macromolecules in suspension was studied. The bead-shell molecular model was applied to build a new model which describes the hydrodynamic effects on the many-body macromolecules with arbitrary shape. The grand diffusion matrix and the grand friction matrix were calculated explicitly to present the intermolecular hydrodynamics. Two fast algorithms based on the block conjugate gradient method and the Schur complement method have been applied to compute the translational and rotational motion of the macromolecules. Therefore, the corresponding displacements and orientations experienced by the diffusing macromolecules can be tracked for the many-body system in fluids. In this way, one can obtain more realistic dynamics and, consequently, more accurate results with the consideration of the hydrodynamic interactions.

Finally, some future research topics will be briefly discussed to extend the work of this dissertation both in algorithms and applications.

- **Optimization of the fast Roten-Prager-Yamakawa solver**

  The fast parallelized Roten-Prager-Yamakawa(RPY) solver developed on multi-core systems has already shown great scalability. It could still be optimized in two ways. First, by optimizing the parallelization: the four Laplace-FMM calls in the far field and the direct evaluation in the near field of the RPY potential can be computed simultaneously. The reason why it can be parallelized in that way is because there is just concurrent read of the adaptive tree data structures and the mutual exclusion could be avoided in some sense. Secondly, instead of calling existing Laplace FMM and traversing the nodes of the tree structure multiple times, the evaluation of the RPY potential could be computed with just one pass of the upward traversal and downward transversal of the tree structure. In Laplace FMM, each particle is characterized with one scalar charge, while in the RPY evaluation, each particle is associated with a force vector which has three components in total with one component in each direction of the three-dimensional space. In Chapter 2, the optimized Laplace-FMM solver was applied to decompose the far-field RPY potential into four Laplace far-field calls with different charges. By applying the same adaptive tree structure, the evaluation of the RPY solver could be finished with one pass of the tree structure. This technique may involve complicated derivation of the mathematical formulas and the corresponding development of the solver. Moreover, hardware architectures such as GPU and distributed systems may also be worth trying for performance improvements.

- **Novel random process generation**

Current algorithms that have been widely used for random process generation are based on either matrix factorization or iterative Krylov subspace methods combined with spectral approximations. The former methods requires explicit covariance matrix formations as the input for most existing solvers which is memory-consuming. For spherical beads system, the random displacement in Brownian dynamic simulations could be accelerated by the development of the fast RPY solver using the iterative methods. For macromolecules with arbitrary shape, because of the moderate number of macromolecules, their random displacements and orientations can be obtained through the approximations of the grand diffusion matrix by using the block conjugate gradient method showed in Chapter 4. This method can be also generalized to random process generation with other covariance matrices.

- **Preconditioner**

  In Chapter 4, as was showed, proper use of preconditioners can greatly improve the efficiency of the simulations by exploring the rigid characteristic of the macromolecules. With this idea in mind, it is possible to apply preconditioners to accelerate random process generation and reduce the number of the iterators.

- **Spatio-temporal coupling**

  For a large particle ensemble, proper treatment in time integration can not only help to track the motion of the Brownian particles with enough resolution, but can also advance the computation using a larger time step to satisfy the long-time simulations. Basically, the time grids can be partitioned into coarse grids and fine grids. For the motions from long-range and slowly changed forces, the coarse time grids can be applied to compute the displacements and orientations. For the short-range parts of the forces which will have big effects on the motions of the particles, the fine time grids can be applied to capture the changes. Some research has been done on hybrid parareal Krylov deferred correction method and adaptive deferred correction methods for choosing applicable time steps and time grids.

- **Next-generation Brownian dynamic solver with hydrodynamics**

  Accounting for all topics described above, it is hoped that the work in this dissertation will shed light on large-scale and long-time Brownian dynamic simulation with hydrodynamics and lead researchers to a very promising novel solver.

REFERENCES

[1] L. Greengard and V. Rokhlin, "A new version of the fast multipole method for the laplace equation in three dimensions," *Acta numerica*, vol. 6, pp. 229–269, 1997.

[2] H. Cheng, L. Greengard, and V. Rokhlin, "A fast adaptive multipole algorithm in three dimensions," *Journal of computational physics*, vol. 155, no. 2, pp. 468–498, 1999.

[3] B. Zhang, *Integral-equation-based fast algorithms and graph-theoretic methods for large-scale simulations*. PhD thesis, University of North Carolina at Chapel Hill, 2010.

[4] `https://online.science.psu.edu`.

[5] M. Długosz, "Effects of hydrodynamic interactions on the apparent 1d mobility of a nonspecifically bound protein following a helical path around dna," *The Journal of Physical Chemistry B*, vol. 119, no. 45, pp. 14433–14440, 2015.

[6] T. Stylianopoulos, M.-Z. Poh, N. Insin, M. G. Bawendi, D. Fukumura, L. L. Munn, and R. K. Jain, "Diffusion of particles in the extracellular matrix: the effect of repulsive electrostatic interactions," *Biophysical journal*, vol. 99, no. 5, pp. 1342–1349, 2010.

[7] K. Hemant, A. Raizaday, P. Sivadasu, S. Uniyal, and S. H. Kumar, "Cancer nanotechnology: Nanoparticulate drug delivery for the treatement of cancer," *International Journal of Pharmacy and Pharmaceutical Sciences*, vol. 7, no. 3, 2015.

[8] M. Allen and A. Imbierski, "A molecular dynamics study of the hard dumb-bell system," *Molecular Physics*, vol. 60, no. 2, pp. 453–473, 1987.

[9] T. Ando, E. Chow, and J. Skolnick, "Dynamic simulation of concentrated macromolecular solutions with screened long-range hydrodynamic interactions: Algorithm and limitations," *The Journal of chemical physics*, vol. 139, no. 12, p. 121922, 2013.

[10] P. S. Doyle and P. T. Underhill, "Brownian dynamics simulations of polymers and soft matter," in *Handbook of Materials Modeling*, pp. 2619–2630, Springer, 2005.

[11] D. L. Ermak and J. McCammon, "Brownian dynamics with hydrodynamic interactions," *The Journal of chemical physics*, vol. 69, no. 4, pp. 1352–1360, 1978.

[12] P. Mereghetti, R. R. Gabdoulline, and R. C. Wade, "Brownian dynamics simulation of protein solutions: structural and dynamical properties," *Biophysical journal*, vol. 99, no. 11, pp. 3782–3791, 2010.

[13] M. Długosz, J. M. Antosiewicz, and J. Trylska, "Association of aminoglycosidic antibiotics with the ribosomal a-site studied with brownian dynamics," *Journal of chemical theory and computation*, vol. 4, no. 4, pp. 549–559, 2008.

[14] P. Mereghetti, R. R. Gabdoulline, and R. C. Wade, "Brownian dynamics simulation of protein solutions: structural and dynamical properties," *Biophysical journal*, vol. 99, no. 11, pp. 3782–3791, 2010.

[15] M. C. Zwier and L. T. Chong, "Reaching biological timescales with all-atom molecular dynamics simulations," *Current opinion in pharmacology*, vol. 10, no. 6, pp. 745–752, 2010.

[16] G. Chirico and J. Langowski, "Brownian dynamics simulations of supercoiled dna with bent sequences.," *Biophysical journal*, vol. 71, no. 2, p. 955, 1996.

[17] K. Klenin, H. Merlitz, and J. Langowski, "A brownian dynamics program for the simulation of linear and circular dna and other wormlike chain polyelectrolytes," *Biophysical Journal*, vol. 74, no. 2, pp. 780–788, 1998.

[18] C. Chen and B. M. Pettitt, "Dna shape versus sequence variations in the protein binding process," *Biophysical journal*, vol. 110, no. 3, pp. 534–544, 2016.

[19] J. J. Sines, S. A. Allison, and J. A. McCammon, "Point charge distributions and electrostatic steering in enzyme/substrate encounter: Brownian dynamics of modified copper/zinc superoxide dismutases," *Biochemistry*, vol. 29, no. 40, pp. 9403–9412, 1990.

[20] R. C. Wade, B. A. Luty, E. Demchuk, J. D. Madura, M. E. Davis, J. M. Briggs, and J. A. McCammon, "Simulation of enzyme–substrate encounter with gated active sites," *Nature Structural & Molecular Biology*, vol. 1, no. 1, pp. 65–69, 1994.

[21] R. C. Wade, "Brownian dynamics simulations of enzyme-substrate encounter," *Chemical Design Automation News*, vol. 1, no. 11, p. 12, 1996.

[22] J. Trylska, V. Tozzini, A. C. Chia-en, and J. A. McCammon, "Hiv-1 protease substrate binding and product release pathways explored with coarse-grained molecular dynamics," *Biophysical journal*, vol. 92, no. 12, pp. 4179–4187, 2007.

[23] G. Wieczorek and P. Zielenkiewicz, "Influence of macromolecular crowding on protein-protein association rates?a brownian dynamics study," *Biophysical journal*, vol. 95, no. 11, pp. 5030–5036, 2008.

[24] M. Długosz and J. Trylska, "Diffusion in crowded biological environments: applications of brownian dynamics," *BMC biophysics*, vol. 4, no. 1, p. 3, 2011.

[25] T. Ando and J. Skolnick, "Brownian dynamics simulation of macromolecule diffusion in a protocell," *Quantum bioinformatics IV: from quantum information to bio-informatics: Tokyo University of Science, Japan, 10-13 March 2010/editors, Luigi Accardi, Wolfgang Freudenberg, Masanori Ohya*, vol. 28, p. 413, 2011.

[26] D. R. Foss and J. F. Brady, "Structure, diffusion and rheology of brownian suspensions by stokesian dynamics simulation," *Journal of Fluid Mechanics*, vol. 407, pp. 167–200, 2000.

[27] B. Lu, X. Cheng, J. Huang, and J. A. McCammon, "Afmpb: an adaptive fast multipole poisson–boltzmann solver for calculating electrostatics in biomolecular systems," *Computer physics communications*, vol. 181, no. 6, pp. 1150–1160, 2010.

[28] X. Liu and E. Chow, "Large-scale hydrodynamic brownian simulations on multicore and manycore architectures," in *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, pp. 563–572, IEEE, 2014.

[29] C. Beenakker, "Ewald sum of the rotne–prager tensor," *The Journal of chemical physics*, vol. 85, no. 3, pp. 1581–1582, 1986.

[30] T. Darden, D. York, and L. Pedersen, "Particle mesh ewald: An n? log (n) method for ewald sums in large systems," *The Journal of chemical physics*, vol. 98, no. 12, pp. 10089–10092, 1993.

[31] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, “A smooth particle mesh ewald method,” *The Journal of chemical physics*, vol. 103, no. 19, pp. 8577–8593, 1995.

[32] C. R. Anderson, “A method of local corrections for computing the velocity field due to a distribution of vortex blobs,” *Journal of Computational Physics*, vol. 62, no. 1, pp. 111–123, 1986.

[33] A. Brandt and A. Lubrecht, “Multilevel matrix multiplication and fast solution of integral equations,” *Journal of Computational Physics*, vol. 90, no. 2, pp. 348–370, 1990.

[34] J. R. Phillips and J. K. White, “A precorrected-fft method for electrostatic analysis of complicated 3-d structures,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 16, no. 10, pp. 1059–1072, 1997.

[35] Z. Liang, Z. Gimbutas, L. Greengard, J. Huang, and S. Jiang, “A fast multipole method for the rotne–prager–yamakawa tensor and its applications,” *Journal of Computational Physics*, vol. 234, pp. 133–139, 2013.

[36] M. W. Davis, “Production of conditional simulations via the lu triangular decomposition of the covariance matrix,” *Mathematical Geology*, vol. 19, no. 2, pp. 91–98, 1987.

[37] M. Tracy and R. Pecora, “Dynamics of rigid and semirigid rodlike polymers,” *Annual Review of Physical Chemistry*, vol. 43, no. 1, pp. 525–557, 1992.

[38] F. Perrin, “Mouvement brownien d’un ellipsoide-i. dispersion diélectrique pour des molécules ellipsoidales,” *J. phys. radium*, vol. 5, no. 10, pp. 497–511, 1934.

[39] F. Perrin, “Mouvement brownien d’un ellipsoide (ii). rotation libre et dépolarisation des fluorescences,” *J. phys. Radium*, vol. 7, no. 1, pp. 1–11, 1936.

[40] S. H. Koenig, “Brownian motion of an ellipsoid. a correction to perrin’s results,” *Biopolymers*, vol. 14, no. 11, pp. 2421–2423, 1975.

[41] M. M. Tirado and J. G. de la Torre, “Translational friction coefficients of rigid, symmetric top macromolecules. application to circular cylinders,” *The Journal of Chemical Physics*, vol. 71, no. 6, pp. 2581–2587, 1979.

[42] M. M. Tirado and J. G. de la Torre, “Rotational dynamics of rigid, symmetric top macromolecules. application to circular cylinders,” *The Journal of Chemical Physics*, vol. 73, no. 4, pp. 1986–1993, 1980.

[43] G. Youngren and A. Acrivos, “Stokes flow past a particle of arbitrary shape: a numerical method of solution,” *Journal of fluid Mechanics*, vol. 69, no. 02, pp. 377–403, 1975.

[44] G. Youngren and A. Acrivos, “Rotational friction coefficients for ellipsoids and chemical molecules with the slip boundary condition,” *The Journal of Chemical Physics*, vol. 63, no. 9, pp. 3846–3848, 1975.

[45] S. A. Allison and P. Nambi, “Transport of charged macromolecules in an electric field by a numerical method. 1. application to a sphere,” *Macromolecules*, vol. 25, no. 15, pp. 3971–3978, 1992.

[46] J. G. Kirkwood, "The statistical mechanical theory of irreversible processes in solutions of flexible macromolecules. visco-elastic behavior," *Recueil des Travaux Chimiques des Pays-Bas*, vol. 68, no. 7, pp. 649–660, 1949.

[47] V. Bloomfield, K. Van Holde, and W. Dalton, "Frictional coefficients of multisubunit structures. ii. application to proteins and viruses," *Biopolymers*, vol. 5, no. 2, pp. 149–159, 1967.

[48] B. Carrasco and J. G. de la Torre, "Hydrodynamic properties of rigid particles: comparison of different modeling and computational procedures," *Biophysical journal*, vol. 76, no. 6, pp. 3044–3057, 1999.

[49] J. G. de la Torre and V. A. Bloomfield, "Hydrodynamic properties of complex, rigid, biological macromolecules: theory and applications," *Quarterly reviews of biophysics*, vol. 14, no. 01, pp. 81–139, 1981.

[50] J. G. De La Torre, S. Navarro, M. L. Martinez, F. Diaz, and J. L. Cascales, "Hydro: a computer program for the prediction of hydrodynamic properties of macromolecules.," *Biophysical journal*, vol. 67, no. 2, p. 530, 1994.

[51] J. G. de la Torre, M. L. Huertas, and B. Carrasco, "Calculation of hydrodynamic properties of globular proteins from their atomic-level structure," *Biophysical Journal*, vol. 78, no. 2, pp. 719–730, 2000.

[52] F. Diaz, B. Carrasco, M. Lopez Martinez, and J. García de la Torre, "Transient electric birefringence of segmentally flexible macromolecules in electric fields of arbitrary strength," *The Journal of Physical Chemistry B*, vol. 104, no. 51, pp. 12339–12346, 2000.

[53] R. Wittkowski and H. Löwen, "Self-propelled brownian spinning top: dynamics of a biaxial swimmer at low reynolds numbers," *Physical Review E*, vol. 85, no. 2, p. 021406, 2012.

[54] T. R. Evensen, S. N. Naess, and A. Elgsaeter, "Free rotational diffusion of rigid particles with arbitrary surface topography: A brownian dynamics study using eulerian angles," *Macromolecular Theory and Simulations*, vol. 17, no. 2-3, pp. 121–129, 2008.

[55] T. R. Evensen, S. N. Naess, and A. Elgsaeter, "Brownian dynamics simulations of rotational diffusion using the cartesian components of the rotation vector as generalized coordinates," *Macromolecular Theory and Simulations*, vol. 17, no. 7-8, pp. 403–409, 2008.

[56] M. X. Fernandes and J. G. de la Torre, "Brownian dynamics simulation of rigid particles of arbitrary shape in external fields," *Biophysical journal*, vol. 83, no. 6, pp. 3039–3048, 2002.

[57] M. Makino and M. Doi, "Brownian motion of a particle of general shape in newtonian fluid," *Journal of the Physical Society of Japan*, vol. 73, no. 10, pp. 2739–2745, 2004.

[58] E. Dickinson, S. A. Allison, and J. A. McCammon, "Brownian dynamics with rotation–translation coupling," *Journal of the Chemical Society, Faraday Transactions 2: Molecular and Chemical Physics*, vol. 81, no. 4, pp. 591–601, 1985.

[59] V. Marquina and R. Gómez, "Algorithm to compute the electric field gradient tensor in ionic crystals," *Revista mexicana de física*, vol. 58, no. 1, pp. 13–18, 2012.

[60] J. R. Phillips and J. K. White, "A precorrected-fft method for electrostatic analysis of complicated 3-d structures," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 16, no. 10, pp. 1059–1072, 1997.

[61] S. S. Kuo, M. D. Altman, J. P. Bardhan, B. Tidor, and J. K. White, "Fast methods for simulation of biomolecule electrostatics," in *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, pp. 466–473, ACM, 2002.

[62] M. D. Altman, J. P. Bardhan, B. Tidor, and J. K. White, "Fftsvd: A fast multiscale boundary-element method solver suitable for bio-mems and biomolecule simulation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 274–284, 2006.

[63] A. W. Appel, "An efficient program for many-body simulation," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, no. 1, pp. 85–103, 1985.

[64] J. Barnes and P. Hut, "A hierarchical o (n log n) force-calculation algorithm," *nature*, vol. 324, no. 6096, pp. 446–449, 1986.

[65] V. Rokhlin, "Rapid solution of integral equations of classical potential theory," *Journal of Computational Physics*, vol. 60, no. 2, pp. 187–207, 1985.

[66] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of computational physics*, vol. 73, no. 2, pp. 325–348, 1987.

[67] J. E. Rodgers, R. Roy, and T. Das, "Many-body calculation of the electric field gradient in the aluminum atom," *Physical Review A*, vol. 14, no. 2, p. 543, 1976.

[68] F. Liu, H. Zhao, and S. Crozier, "On the induced electric field gradients in the human body for magnetic stimulation by gradient coils in mri," *Biomedical Engineering, IEEE Transactions on*, vol. 50, no. 7, pp. 804–815, 2003.

[69] G. Liu, J. Marshall, S. Li, and Q. Yao, "Discrete-element method for particle capture by a body in an electrostatic field," *International Journal for Numerical Methods in Engineering*, vol. 84, no. 13, pp. 1589–1612, 2010.

[70] A. Aguado and P. A. Madden, "Ewald summation of electrostatic multipole interactions up to the quadrupolar level," *The Journal of chemical physics*, vol. 119, no. 14, pp. 7471–7483, 2003.

[71] A. Volkov, H. F. King, P. Coppens, and L. J. Farrugia, "On the calculation of the electrostatic potential, electric field and electric field gradient from the aspherical pseudoatom model," *Acta Crystallographica Section A: Foundations of Crystallography*, vol. 62, no. 5, pp. 400–408, 2006.

[72] V. D.A., M. A.N., and K. V.K., *Quantum theory of angular momentum : irreducible tensors, spherical harmonics, vector coupling coefficients, 3nj symbols.* World Scientific Pub., 1998.

[73] H. Cheng, *Fast, accurate methods for the evaluation of harmonic fields in composite materials.* PhD thesis, 1995.

[74] E. W. Hobson, *The theory of spherical and ellipsoidal harmonics.* CUP Archive, 1931.

[75] C. A. White and M. Head-Gordon, "Rotating around the quartic angular momentum barrier in fast multipole method calculations," *The Journal of Chemical Physics*, vol. 105, no. 12, pp. 5061–5067, 1996.

[76] J. Rotne and S. Prager, "Variational treatment of hydrodynamic interaction in polymers," *The Journal of Chemical Physics*, vol. 50, no. 11, pp. 4831–4837, 1969.

[77] C. Beenakker and P. Mazur, "Diffusion of spheres in suspension: Three-body hydrodynamic interaction effects," *Physics Letters A*, vol. 91, no. 6, pp. 290–291, 1982.

[78] L. Ying, G. Biros, and D. Zorin *Journal of Computational Physics*, vol. 196, no. 2, pp. 591–626, 2004.

[79] S. B. Zimmerman and A. P. Minton, "Macromolecular crowding: biochemical, biophysical, and physiological consequences," *Annual review of biophysics and biomolecular structure*, vol. 22, no. 1, pp. 27–65, 1993.

[80] J. D. Dwyer and V. A. Bloomfield, "Brownian dynamics simulations of probe and self-diffusion in concentrated protein and dna solutions.," *Biophysical journal*, vol. 65, no. 5, p. 1810, 1993.

[81] S. S. Wijmenga and A. Maxwell, "Rotational diffusion of short dna fragments in polyacrylamide gels: An electric birefringence study," *Biopolymers*, vol. 25, no. 11, pp. 2173–2186, 1986.

[82] H.-J. Sass, G. Musco, S. J. Stahl, P. T. Wingfield, and S. Grzesiek, "Solution nmr of proteins within polyacrylamide gels: diffusional properties and residual alignment by mechanical stress or embedding of oriented purple membranes," *Journal of biomolecular NMR*, vol. 18, no. 4, pp. 303–309, 2000.

[83] N. C. Stellwagen, "The use of transient electric birefringence to characterize the conformation of dna in solution, the mechanism of dna gel electrophoresis, and the structure of agarose gels," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 209, no. 2, pp. 107–122, 2002.

[84] N. Fatin-Rouge, A. Milon, J. Buffle, R. R. Goulet, and A. Tessier, "Diffusion and partitioning of solutes in agarose hydrogels: the relative influence of electrostatic and specific interactions," *The Journal of Physical Chemistry B*, vol. 107, no. 44, pp. 12126–12137, 2003.

[85] N. Fatin-Rouge, K. Starchev, and J. Buffle, "Size effects on diffusion processes within agarose gels," *Biophysical Journal*, vol. 86, no. 5, pp. 2710–2719, 2004.

[86] S. A. Tatarkova and D. A. Berk, "Probing single dna mobility with fluorescence correlation microscopy," *Physical Review E*, vol. 71, no. 4, p. 041913, 2005.

[87] G. Hungerford, A. Rei, M. I. C. Ferreira, K. Suhling, and C. Tregidgo, "Diffusion in a sol-gel-derived medium with a view toward biosensor applications," *The Journal of Physical Chemistry B*, vol. 111, no. 13, pp. 3558–3562, 2007.

[88] H. Shen, Y. Hu, and W. Saltzman, "Dna diffusion in mucus: effect of size, topology of dnas, and transfection reagents," *Biophysical journal*, vol. 91, no. 2, pp. 639–644, 2006.

[89] C. W. Oseen, *Hydrodynamik*, vol. 1. Akad. Verl.-Ges., 1927.

[90] J. Burgers, "Second report on viscosity and plasticity," *Nordemann, New York*, p. 113, 1938.

[91] J. G. Kirkwood, "The general theory of irreversible processes in solutions of macromolecules," *Journal of Polymer Science Part B: Polymer Physics*, vol. 34, no. 4, pp. 597–610, 1996.

[92] V. Bloomfield, W. Dalton, and K. Van Holde, "Frictional coefficients of multisubunit structures. i. theory," *Biopolymers*, vol. 5, no. 2, pp. 135–148, 1967.

[93] D. La Torre, J. Garcia, and V. A. Bloomfield, “Hydrodynamic properties of macromolecular complexes. i. translation,” *Biopolymers*, vol. 16, no. 8, pp. 1747–1763, 1977.

[94] D. La Torre, J. Garcia, and V. A. Bloomfield, “Hydrodynamics of macromolecular complexes. ii. rotation,” *Biopolymers*, vol. 16, no. 8, pp. 1765–1778, 1977.

[95] D. La Torre, J. Garcia, and V. A. Bloomfield, “Hydrodynamics of macromolecular complexes. iii. bacterial viruses,” *Biopolymers*, vol. 16, no. 8, pp. 1779–1793, 1977.

[96] D. C. Teller, E. Swanson, and C. de Haën, “The translational friction coefficient of proteins.,” *Methods in enzymology*, vol. 61, pp. 103–124, 1978.

[97] R. C. Wade, M. E. Davis, B. A. Luty, J. D. Madura, and J. A. McCammon, “Gating of the active site of triose phosphate isomerase: Brownian dynamics simulations of flexible peptide loops in the enzyme.,” *Biophysical journal*, vol. 64, no. 1, p. 9, 1993.

[98] C.-E. Chang, T. Shen, J. Trylska, V. Tozzini, and J. A. McCammon, “Gated binding of ligands to hiv-1 protease: Brownian dynamics simulations in a coarse-grained model,” *Biophysical journal*, vol. 90, no. 11, pp. 3880–3885, 2006.

[99] N. Wang, G. A. Huber, and J. A. McCammon, “Assessing the two-body diffusion tensor calculated by the bead models,” *The Journal of chemical physics*, vol. 138, no. 20, p. 204117, 2013.

[100] L. Du, Y. Futamura, and T. Sakurai, “Block conjugate gradient type methods for the approximation of bilinear form cha-1b,” *Computers & Mathematics with Applications*, vol. 66, no. 12, pp. 2446–2455, 2014.