Summer 2017

Efficient coarse-grained brownian dynamics simulations for dna and lipid bilayer membrane with hydrodynamic interactions

Szu-Pei Fu
New Jersey Institute of Technology

Recommended Citation

Fu, Szu-Pei, "Efficient coarse-grained brownian dynamics simulations for dna and lipid bilayer membrane with hydrodynamic interactions" (2017). *Dissertations*. 32.
https://digitalcommons.njit.edu/dissertations/32

ABSTRACT

EFFICIENT COARSE-GRAINED BROWNIAN DYNAMICS SIMULATIONS FOR DNA AND LIPID BILAYER MEMBRANE WITH HYDRODYNAMIC INTERACTIONS

by
Szu-Pei Fu

The coarse-grained molecular dynamics (CGMD) or Brownian dynamics (BD) simulation is a particle-based approach that has been applied to a wide range of biological problems that involve interactions with surrounding fluid molecules or the so-called hydrodynamic interactions (HIs). From simple biological systems such as a single DNA macromolecule to large and complicated systems, for instances, vesicles and red blood cells (RBCs), the numerical results have shown outstanding agreements with experiments and continuum modeling by adopting Stokesian dynamics and explicit solvent model. Finally, when combined with fast algorithms such as the fast multipole method (FMM) which has nearly optimal complexity in the total number of CG particles, the resulting method is parallelizable, scalable to large systems, and stable for large time step size, thus making the long-time large-scale BD simulation within practical reach. This will be useful for the study of a large collection of molecules or cells immersed in the fluids.

This dissertation can be divided into three main subjects: (1) An efficient algorithm is proposed to simulate the motion of a single DNA molecule in linear flows. The algorithm utilizes the integrating factor method to cope with the effect of the linear flow of the surrounding fluid and applies the Metropolis method (MM) in [N. Bou-Rabee, A. Donev, and E. Vanden-Eijnden, Multiscale Model. Simul. **12**, 781 (2014)] to achieve more efficient BD simulation. More importantly, this proposed method permits much larger time step size than methods in previous literature while still maintaining the stability of the BD simulation, which is advantageous for long-time BD simulation. The numerical results on λ-DNA agree very well with

both experimental data and previous simulation results. (2) Lipid bilayer membranes have been extensively studied by CGMD simulations. Numerical efficiencies have been reported in the cases of aggressive coarse-graining, where several lipids are coarse-grained into a particle of size $4 \sim 6$ nm so that there is only one particle in the thickness direction. In [H. Yuan *et al.*, Phys. Rev. E, **82**, 011905 (2010)], Yuan *et al.* proposed a pair-potential between these one-particle-thick coarse-grained lipid particles to capture the mechanical properties of a lipid bilayer membrane, such as gel-fluid-gas phase transitions of lipids, diffusion, and bending rigidity. This dissertation provides a detailed implementation of this interaction potential in LAMMPS to simulate large-scale lipid systems such as a giant unilamellar vesicle (GUV) and RBCs. Moreover, this work also considers the effect of cytoskeleton on the lipid membrane dynamics as a model for RBC dynamics, and incorporates coarse-grained water molecules to account for hydrodynamic interactions. (3) An action field method for lipid bilayer membrane model is introduced where several lipid molecules are represented by a Janus particle with corresponding orientation pointing from lipid head to lipid tail. With this level of coarse-grained modeling, as the preliminary setup, the lipid tails occupy a half sphere and the lipid heads take the other half. An action field is induced from lipid-lipid interactions and exists everywhere in the computational domain. Therefore, a hydrophobic attraction energy can be described from utilizing the variational approach and its minimizer with respect to the action field is the so-called screened Laplace equation. For the numerical method, the well-known integral equation method (IEM) has great capability to solve exterior screened Laplace equation with Dirichlet boundary conditions. Finally, one then can obtain the lipid dynamics to validate the self-assembly property and other physical properties of lipid bilayer membrane. This approach combines continuum modeling with CGMD and gives a different perspective to the membrane energy model from the traditional Helfrich membrane free energy.

EFFICIENT COARSE-GRAINED BROWNIAN DYNAMICS SIMULATIONS FOR DNA AND LIPID BILAYER MEMBRANE WITH HYDRODYNAMIC INTERACTIONS

by
Szu-Pei Fu

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
and Rutgers, The State University of New Jersey–Newark
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mathematical Sciences

Department of Mathematical Sciences, NJIT
Department of Mathematics and Computer Science, Rutgers–Newark

August 2017

APPROVAL PAGE

EFFICIENT COARSE-GRAINED BROWNIAN DYNAMICS SIMULATIONS FOR DNA AND LIPID BILAYER MEMBRANE WITH HYDRODYNAMIC INTERACTIONS

Szu-Pei Fu

Dr. Yuan-Nan Young, Dissertation Co-Advisor Date
Associate Professor of Mathematical Sciences, New Jersey Institute of Technology

---

Dr. Shidong Jiang, Dissertation Co-Advisor Date
Associate Professor of Mathematical Sciences, New Jersey Institute of Technology

---

Dr. Jonathan H. Luke, Committee Member Date
Chair and Professor of Mathematical Sciences, New Jersey Institute of Technology

---

Dr. Michael S. Siegel, Committee Member Date
Professor of Mathematical Sciences, New Jersey Institute of Technology

---

Dr. Zhangli Peng, Committee Member Date
Assistant Professor of Aerospace and Mechanical Engineering, University of Notre Dame

BIOGRAPHICAL SKETCH

Author:	Szu-Pei Fu
Degree:	Doctor of Philosophy
Date:	August 2017

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mathematical Sciences,
  New Jersey Institute of Technology, Newark, NJ, 2017
- Master of Science in Mathematical and Computational Finance,
  New Jersey Institute of Technology, Newark, NJ, 2010
- Bachelor of Business Administration,
  Soochow University, Taipei, Taiwan, 2006

Major: Applied Mathematics

Presentations and Publications:

S.-P. Fu, Y.-N. Young and S. Jiang, "Efficient Brownian dynamics simulation of DNA molecules with hydrodynamic interactions in linear flows," *Phys. Rev. E*, 91, 063008, 2015.

S.-P. Fu, Z. Peng, R. Kfoury, H. Yuan and Y.-N. Young, "Lennard-Jones type pair-potential method for coarse-grained lipid bilayer membrane simulations in LAMMPS," *Comput. Phys. Commun.*, 210C, 193–203, 2017.

A fast multipole method and a Metropolis method for coarse-grained Brownian dynamics simulations of a DNA with Hydrodynamic Interactions, 67th Annual Meeting of the APS–DFD (San Fransisco, Nov 2014).

Efficient Brownian dynamics simulation of DNA molecules with hydrodynamic interactions in linear flows, The 5th Northeast Complex Fluids and Soft Matter Workshop (NCS5) (NYU Tandon School of Engineering, Jan 2016).

An accurate Metropolis-Hastings algorithm and a fast multipole method for coarse-grained lipid Bilayer membrane in solvent, The 2016 SIAM Conference on Mathematical Aspects of Materials Science (MS16) (Philadelphia, May 2016).

Brownian dynamics simulations of lipid bilayer membrane with hydrodynamic interactions in LAMMPS, The 2016 SIAM Annual Meeting (AN16) (Boston, Jul 2016).

Brownian dynamics simulations of lipid bilayer membrane with hydrodynamic interactions in LAMMPS, 69th Annual Meeting of the APS–DFD (San Fransisco, Nov 2016).

Brownian dynamics simulations of lipid bilayer membrane with hydrodynamic interactions in LAMMPS, The 2017 SIAM Conference on Computational Sciences and Engineering (CSE17) (Atlanta, GA, Mar 2017).

Brownian dynamics simulations of lipid bilayer membrane with hydrodynamic interactions in LAMMPS, APS March Meeting 2017 (New Orleans, LA, Mar 2017).

NJIT DMS Student Summer Talks (NJIT, Summer 2013, Summer 2014, Summer 2015, Summer 2016, Summer 2017).

Frontiers in Applied and Computational Mathematics (FACM) Poster Session (NJIT, May 2014, Jun 2016).

NJIT Graduate Student Association – Graduate Research Day Poster Session (NJIT, Nov 2016).

43rd Northeast Bioengineering Conference (NEBEC) Poster Session (NJIT, Apr 2017).

ICERM-HKUST International Program–Integral Equation Methods, Fast Algorithms and Their Applications to Fluid Dynamics and Materials Science (HKUST, Hong Kong, Jan 2017; ICERM at Brown University, RI, Jun 2017)

This thesis is dedicated to my parents and brother.

ACKNOWLEDGMENT

I would like to acknowledge my Advisors, Dr. Yuan-Nan Young and Dr. Shidong Jiang, for teaching and mentoring me during my PhD years. Without their guidance and patience, this research work would not be possible to be done. I also want to send my appreciation for sharing their research connections with me and letting me be involved in many collaborative research.

Second, I would like to thank my committee members Professor Jonathan Luke, Professor Michael Siegel and Professor Zhangli Peng for taking time to review my dissertation research.

Third, I would like to thank: Professor Nawaf Bou-Rabee for his guidance and discussion on the study of stochastic processes and his Metropolis algorithm; Professor Zhangli Peng for his guidance on learning LAMMPS and his willingness to help whenever needed; Professor Andreas Klöckner and Matt Wala for their help and guidance on using their python code for Integral Equation Method, the HKUST and ICERM for providing financial support and relaxing, collaborative environment when I was writing my thesis; Professor Rolf Ryham for taking time on discussion of action field method and his guidance on learning variational approach of membrane dynamics; NSF grant DMS-1222550, DMS-1614863 and DMS-1418918 for the partial support.

A special thanks must go to my colleagues in the same year, Ivana Seric, Ensela Mema and Aminur Rahman, for the support in my PhD life. We are like a family.

Last, this dissertation research would not have been completed without my family's support, I would like to express my deepest appreciation to my parents and my brother for taking care of me during the PhD career.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

Numerical investigations have been widely utilized to understand fluid-structure interactions (FSI) in biological systems, such as the hydrodynamics of a DNA molecule [26, 42, 49, 78, 82, 88], lipid membranes [25, 60, 61, 100, 101], a vesicle (self-enclosing lipid bilayer membrane) [4, 21, 28, 52, 55, 64, 76, 79–81, 91, 95], red blood cells (RBCs) [54, 70, 84], sperms [63, 83] or flagellated swimmers [56, 98]. The immersed boundary method (IBM) has been an extremely popular tool for simulating fluid-structure interactions. Over the years tremendous progress has been made to refine IBM for various fluid-structure interaction problems in biology [31, 40, 41, 45, 48]. The phase-field method (PFM) is another popular numerical method for fluid-structure interaction [22, 33]. The boundary integral (element) method is an efficient and accurate alternative when an integral formulation exists [73].

Adaptive mesh refinement and/or regularization are often needed for IBM and/or PFM to resolve a sharp interface in the fluid domain. Boundary integral simulations may perform better for a moving boundary problem only when an integral formulation is possible. Therefore, particle-based methods, such as the coarse-grained Brownian dynamics (CGBD) can be a natural choice for numerical simulations for fluid-structure interaction when the geometry of the structure decides the smallest scale in the problem. The advantage of the CGBD model is that, if treated properly and consistently, more biological details may be incorporated in the FSI.

1.1 DNA Simulations: Implicit-Solvent Modeling

The dynamics of a single DNA or polymer macromolecule in fluid flow has been extensively investigated experimentally ([68, 85] and references therein), theoretically [19, 24, 57] and numerically [42, 78]. Bulk rheological experiments such as flow birefringence and light scattering measurements give inference of polymer conformation, orientation, and chain stretch in fluid flows. The advent of single molecule visualizations using fluorescence microscopy allows for the direct observation of complex dynamics of individual macromolecules in dilute solutions under shear, extensional, and general two-dimensional mixed flows [5, 39, 77, 85, 86]. These measurements provide data for direct comparison against fully parametrized models of macromolecules, such as the bead-spring model for DNA with finite extensibility, excluded volume (EV) [71] effects and hydrodynamic interactions (HI) [78]. Brownian dynamics (BD) simulations of bead-spring and bead-rod models with free-draining assumption (no hydrodynamic interactions) give quantitative agreement with short chains of double stranded DNA experiments, for example, $\sim 21\mu$m long λ-DNA [43,49,87]. However, for longer chains of DNA such as $O(10^2)$ μm or longer, HI needs to be included for quantitative agreement. For truly flexible polymers such as single stranded DNA or synthetic polymers, one can expect that HI will be important even for short chains.

Following Ermak and McCammon [24], Schroeder *et al.* modeled the DNA macromolecule as a system of N particles subject to interparticle forces, fluctuating HI and EV forces [43, 49, 78]. They designed a semi-implicit predictor-corrector scheme for simulating the Brownian system, and illustrated how effects of HI and EV between monomers in a flexible polymer chain influence both the equilibrium and non-equilibrium physical properties of DNA macromolecules [78], consistent with the experimental observations. The non-local HI between the DNA macromolecule and the surrounding fluid involves an integral of hydrodynamic forces between a point and the rest of the macromolecule. Within the coarse-grained framework, this integral is equivalent to a sum of all hydrodynamic forces between a bead and the rest of the system. In implicit-solvent modeling we can adopt the Rotne-Prager-Yamakawa (RPY) tensor [74] (i.e., the mobility tensor) for HI effects:

$$D_{ij} = \frac{k_B T}{\zeta_{res}} I_{ij}, \quad \text{if } i = j \tag{1.1}$$

$$D_{ij} = \frac{k_B T}{8\pi\eta r_{ij}} \left[\left(1 + \frac{2a^2}{3r_{ij}^2}\right) I_{ij} + \left(1 - \frac{2a^2}{r_{ij}^2}\right) \frac{\mathbf{r}_{ij}\mathbf{r}_{ij}}{r_{ij}^2} \right], \quad \text{if } i \neq j,\ r_{ij} \geq 2a \tag{1.2}$$

$$D_{ij} = \frac{k_B T}{\zeta_{res}} \left[\left(1 - \frac{9r_{ij}}{32a}\right) I_{ij} + \frac{3\mathbf{r}_{ij}\mathbf{r}_{ij}}{32ar_{ij}} \right], \quad \text{if } i \neq j,\ r_{ij} < 2a \tag{1.3}$$

where D_{ij} is the mobility of bead i due to bead j in three dimensions, I_{ij} the 3×3 identity matrix, and $\zeta_{res} = 6\pi\eta a$ is the bead resistivity with η the solvent viscosity and a the radius of beads.

There are two main challenges for the long-time large-scale BD simulations with HI and EV effects. First, the correlated random noises in the change of displacement vectors at each time step are proportional to $\sqrt{\Delta t}$ with Δt the time step size. This makes the design of high-order marching scheme very difficult and forces very small Δt for many explicit or semi-implicit numerical schemes in order to avoid the numerical instability. The problem becomes much more severe for long-time BD simulations since it then requires a very large total number of time steps for the system to reach the desired state, which very often leads to weeks of simulation time even for one run. Second, the direct evaluation of the particle interaction at each time step requires $O(N^2)$ operations where N is the total number of particles in the system; and the generation of the correlated random displacements requires $O(N^3)$ operations if the standard Choleski factorization is used or $O(KN^2)$ if the Chebyshev spectral approximation is used for computing the product of the matrix square root and an arbitrary vector (here K is the condition number of the covariance matrix) (see, for example, [44]). To summarize, in order to efficiently utilize the BD simulation as a practical tool to study the properties of large systems, say, many polymers or a large collection of DNA molecules in a fluid, it is essential to address the following two questions: how to numerically integrate the system with greater accuracy and better stability property which enables much large time step size? How to expedite the calculations of long-range particle interactions and associated correlated random effects in BD simulations with HI, especially for large N?

For BD simulations near equilibrium, a Metropolis scheme for the temporal integration has been recently proposed [11, 12] for a Markov process whose generator is self-adjoint (with respect to a density function) to expedite simulations to reach equilibrium in a timely fashion. Under this scheme, stable and accurate BD simulations of DNA in a solvent are obtained using time step sizes that are orders of magnitude larger than those for predictor-corrector schemes [42, 49, 78]. However, such a Metropolis scheme relies heavily on the self-adjointness of the Markov process generator for a quiescent flow.

In this dissertation work, we present an efficient algorithm for the simulations of the dynamics of DNA macromolecules under linear flows. Our method is based upon the Metropolis scheme developed in [12] for self-adjoint diffusions, which is applicable for the study of the DNA molecule to its equilibrium configurations in a quiescent flow. When a linear flow such as an extensional or a shear flow is present in the surrounding fluid, the diffusion process is not self-adjoint anymore. We first apply the method of integrating factors to recast the associated system of stochastic differential equations (SDE) into a form such that the effect of the linear flow is taken into account by the integrating factor. We then modify the Metropolis scheme in [12] to update the displacements of beads which are the coarse-grained representation of the long chain DNA molecule. Our numerical experiments show that our scheme allows much greater time step size in the BD simulation and avoids the numerical instability. The numerical results on the study of λ-DNA agree very well with the experimental data [85, 86] and previous simulation results [78]. Moreover, the total simulation time is significantly reduced in our methods as compared with the semi-implicit predictor-corrector scheme [78].

For BD simulations that involve a "large" number of interacting particles (so large that the calculation of their mutual interactions becomes the computational bottleneck), recent work in [44, 53] reduces the computational cost of particle interactions from $O(N^2)$ to $O(N)$ and the cost of generating the correlated random displacements from $O(N^3)$ or $O(KN^2)$ to $O(KN)$. These works yield an essentially linear algorithm with respect to the total number of particles in the BD simulation of interacting particles. The method developed in [44, 53] extends the original fast multipole method (FMM) [29] to the case of the RPY tensor and combines it with the spectral Lanczos decomposition method (SLDM) to generate correlated random vectors whose correlation is determined by the RPY tensor. To demonstrate that long-time large-scale BD simulations (with or without linear flows) for large systems of interacting particles are within practical reach when our modified Metropolis scheme is combined with the fast method in [44,53], we use two examples to illustrate that our algorithms do efficiently capture the HI effects in a large BD system when compared with experimental results: One is the hysteretic extension of a long DNA molecule in a linear extensional flow, and the other is the multiple DNA molecules in an oscillatory shear flow.

1.2 Vesicle and Red Blood Cell Simulations: Explicit-Solvent Modeling

In recent years, great progress has been made to understand the dynamics of vesicles (self-enclosing lipid bilayer membranes) and red blood cells (RBCs) in aqueous solutions due to their relevance in a wide range of fields such as biology, biophysics, and biomedical engineering. A main component of the vesicle and RBC membranes is the amphiphilic lipid molecules, which self-assemble to form liposomes (vesicles) or micelles. In a viscous fluid flow, the vesicle may deform due to the balance between viscous stress, bending resistance and tension forces in the membrane. In this work we focus on pure lipid bilayer membrane and neglect the effects of multiple lipid species and different transmembrane proteins on the lipid bilayer membrane. For a pure lipid bilayer membrane, the equilibrium shapes of vesicles immersed in fluid have been widely studied in continuum modeling and coarse-grained molecular dynamics (CGMD) modeling. In the continuum framework, the dynamics and equilibrium shape of a lipid bilayer membrane is governed by the Helfrich free energy that consists of mean, Gaussian and spontaneous curvatures of the membrane [34]. The total membrane energy is integrated over the surface Ω as

$$E = \int_\Omega \left[\gamma + \frac{B}{2} \left(c_1 + c_2 - c_0 \right)^2 + \bar{\kappa} c_1 c_2 \right] dA, \tag{1.4}$$

where $\{c_1, c_2\}$ are the principle curvatures, γ is the surface tension, B is the bending rigidity, $\bar{\kappa}$ is the saddle-splay modulus and c_0 is the spontaneous curvature. Without membrane tension ($\gamma = 0$) and saddle-splay energy ($\bar{\kappa} = 0$), Equation (1.4) is reduced to the classical Helfrich-Canham energy which consists of only the bending rigidity and spontaneous curvature:

$$E = \int_\Omega \frac{B}{2} \left(c_1 + c_2 - c_0 \right)^2 dA. \tag{1.5}$$

Continuum modeling has successfully reproduced vesicle and RBC dynamics in a fluid flow [67], even though several physical properties of the lipid bilayer membrane (such as membrane diffusivity and temperature-dependent bending rigidity) have to be assumed in the continuum framework. In the aggressive coarse-graining, the lipid bilayer membrane is modeled as a one-particle-thick monolayer of coarse-grained lipid particles, with each particle containing many lipids. By specifying the potential for particle interaction, the lipid properties of hydrophilic heads and hydrophobic tails can be preserved [100]. Hence, a characteristic length scale of one-particle-thick CGMD membrane model is often chosen as the thickness ($4-5$ nm) of lipid bilayer membrane.

In this dissertation work we focus on the meshfree CGMD membrane model, first introduced by Drouffe *et al.* in early 1990s [20]. They showed that their simulation method can well predict the self-assembly property of lipid bilayer membrane. By controlling the model parameters in one-particle-thick CGMD membrane model, Yuan *et al.* showed that the various lipid phases (gel, fluid and gas) and the physically reasonable lipid diffusivity can be achieved [100]. With the development of LAMMPS and advancement of large-scale parallel computing, long-time simulations are now more achievable for examining the dynamics of lipid bilayer membranes in aqueous solutions. Here we adopt CGMD in LAMMPS for numerical investigation of vesicle and RBC dynamics.

To fully understand the dynamics of a lipid bilayer membrane with spontaneous curvature, two approaches are introduced in the particle based MD simulations: (1) Pair-potential for fluid lipid membrane which involves membrane-solvent, fluid membrane network and solvent-solvent interactions [17, 51, 100, 101]. (2) Local multi-body curvature energy which consists of a local curvature potential based on aplanarity, excluded volume (as a repulsive) potential and an attractive potential that depends on the local particle density [60, 61]. Both approaches have their particular advantages in numerical investigation, and results show that they are able to reproduce the self-assembly property of a fluid-phase lipid membrane. In this work we implement the pair-potential function for CG lipid bilayer membrane in LAMMPS. We will illustrate how to simulate the membrane dynamics with either explicit or implicit solvent in LAMMPS to study the membrane property and dynamic shape transitions of vesicles and red blood cells. Two specific applications to the biological systems using LAMMPS are presented in this paper:

1. Vesicle shape transitions

Seifert *et al.* calculated the phase diagram of vesicle shape transitions using the Helfrich free energy (with a spontaneous curvature) described above, and provided detailed theoretical insight to the vesicle equilibrium shapes by comparing with experiments [80, 81]. We will illustrate how to use LAMMPS to model the vesicle shape transition due to volume reduction. In the one-particle-thick CGMD membrane model, we will include coarse-grained water molecules (explicit solvent) to account for hydrodynamic interactions.

2. Resting shapes of RBCs

We extend the CGMD modeling of a vesicle to a RBC, where the surface structure is a lipid bilayer membrane coupled with a layer of cytoskeleton network underneath. Laboratory experiments show that RBC can form stomatocyte, discocyte and echinocyte minimum-energy shapes. In early 2000s, Lim *et al.* introduced the mechanical theory for predicting the stable RBC shapes involving area difference between outer and inner leaflets of RBCs. Lim *et al.* adopted area-difference-elasticity model (ADE) which describes the free energy using spontaneous curvature and geometrical area difference of RBCs to reproduce the RBC shapes under cases of reduce relaxed area difference and compared the simulation results of RBC shapes with experiments [54].

In the past, the hydrodynamics of a RBC in a fluid flow has been simulated by using finite element method (FEM), immersed boundary method (IB), dissipative particle dynamics (DPD) [40, 66, 69] and Langevin dynamics. In this work we adopt Nosé–Hoover algorithms in CGMD and compare our numerical results with previous results from continuum model simulations.

1.3 Amphiphilic Lipid Dynamics: Continuum Modeling

Many experimental results have shown that a stalk structure of fluid membrane occurs when two bilayer surfaces create contacts, therefore the hemifusion diaphragm, membrane fusion and pore expansions are then formed. In recent years Kawamoto *et al.* have performed the MD simulations to study membrane fusion and measured the changes of free energy based on the Helfrich membrane model at the stalk formations. The methodology involves a specific model parameter related to the membrane separations and the elastic parameters (bending rigidity, saddle splay modulus and spontaneous curvature) are obtained from experiments where the types of lipid molecules are DOPC, DOPE and DMPC [46].

In this dissertation work, we propose a new perspective of membrane dynamics from the views of lipid-lipid interactions and energy variations to understand these phenomena which is not able to be reproduced from highly aggressive coarse-grained modeling. This approach can be traced back in early 1970s, a Croatian applied mathematician approximated the potential energy of the fluid structure interactions (FSI) using screed Coulomb force [59]. Referring to recent work, Ryham *et al.* calculate the entire least-energy pathway of membrane fusion and the membrane deformation using the so-called string method in continuum mechanics [75].

The main idea is that the lipid bilayer surface Σ is assembled from distal monolayer Σ_D and proximal monolayer Σ_P. The total energy of the elastic contribution and long-range surface interactions between two monolayers is given by:

$$\begin{aligned} E = &\int_\Sigma \frac{K_C}{2}\left[(\text{div}\ \mathbf{d} + k_0)^2 - k_0^2\right] + \frac{K_\theta}{2}\left|\mathbf{d}\times\mathbf{n}\right|^2 + \frac{K_A}{2}\frac{(a-a_0)^2}{aa_0}dS \\ &+ \int_{\Sigma_P} W(h)\cos^2(\theta)dS + V(\Sigma_F). \end{aligned} \tag{1.6}$$

where the unit director $\mathbf{d}$ represents the direction from single lipid head toward lipid tail, $\mathbf{n}$ the unit surface normal, k_0 the spontaneous curvature, lipid cross section area a (perpendicular to $\mathbf{d}$) and the resting lipid cross section area a_0. Σ_F is the surface when membran fusion occurs. The first integral in Equation (1.6) models the elastic energy including deformation, tilt, splay and stretch. Constant coefficients K_C, K_θ and K_A are splay modulus, tilt modulus and two-dimensional modulus for stretching monolayer area. If $\mathbf{d}$ and $\mathbf{n}$ are parallel everywhere, $(\text{div } \mathbf{d} + k_0)^2 - k_0^2$ would reduce to $(2H - k_0)^2$ where H is the mean curvature.

The second integral describes the van der Waal attractions given by

$$W(h) = \begin{cases} -\frac{H}{12\pi}\left[\frac{1}{h_w^2} - \frac{2}{(h_w-2l_0)^2} + \frac{1}{(h_w+4l_0)^2}\right] + W_0 & h \gg h_0, \\ LP_0 e^{-h/L} & \text{otherwise} \end{cases}, \tag{1.7}$$

where $H = 1\ kT$ is the Hamaker coefficient for hydrocarbon across water, W_0 is a constant, h_w the water thickness, l_0 the resting length of lipid and θ the angle between the proximal surface and horizontal plane. For the interactions between proximal surfaces separated by small distance h, a repulsive potential is given for hydration pressure where P_0 is the pressure at zero separation and L is the decay length determined from pressure-distance curves. The last term in Equation (1.6) is the hydrophobic attraction potential for membrane fusion and is given by the following formula:

$$V(\Sigma_F) = \sigma \int_\Omega \rho|\nabla\eta|^2 + \rho^{-1}|\eta|^2 dx. \tag{1.8}$$

where $\sigma = 11kT$ is the interfacial tension of a hydrocarbon-water interface. The hydrophobic attraction field η satisfies $\eta = 1$ on the fissure surface Σ_F and $\eta = 0$ on the monolayer surfaces. We aim to replace the membrane total energy in Equation (1.6) by only considering the hydrophobic attraction potential $V(\Sigma)$ shown above since the variation of this energy resolves the dynamics of lipid and the lipid orientations and configurations can be obtained. As the consequence, the tilt, splay and stretch are all preserved when we perform the energy evaluation onto whole membrane surfaces.

At this point, the only thing left is to solve the action field η numerically then apply it to the membrane energy. It can be derived that the minimizer of the energy satisfies the BVP:

$$\rho^2\Delta\eta = \eta, \quad \eta = 1 \text{ on } \Sigma_T, \quad \eta = 0 \text{ on } \Sigma_H. \tag{1.9}$$

where ρ is the attraction decay length which can be obtained from experiments. We define the hydrophilic and hydrophobic layers as Σ_T and Σ_F, respectively. The equation above is the so-called screened Laplacian where the fundamental solutions in two and three dimensional spaces are given by

$$\begin{aligned} G_{2D}(\mathbf{x},\mathbf{x}') &= \frac{1}{2\pi}K_0(\rho^{-1}|\mathbf{x}-\mathbf{x}'|) = \frac{i}{4}H_0^{(1)}(\rho^{-1}|\mathbf{x}-\mathbf{x}'|), \\ G_{3D}(\mathbf{x},\mathbf{x}') &= \frac{1}{4\pi|\mathbf{x}-\mathbf{x}'|}e^{\frac{-|\mathbf{x}-\mathbf{x}'|}{\rho}}. \end{aligned} \tag{1.10}$$

The series solution for the screened Laplace equation can be acquired from the paper [30] and an accurate series formation can be achieved by adopting the quadrature by expansion (QBX) method [47]. To solve the BVP above numerically we can use the integral equation method (IEM) combining with the QBX, the FMM and the iterative solver. That is, with this approach, no “artificial” attractive and repulsive potentials will be involved in the action field model and this thus becomes more realistic to the molecular details of membrane dynamics.

This dissertation is organized as the follows: In §2 we demonstrate the simulations of a single DNA macromolecule in linear fluid flow where we adopt the implicit modeling with Metropolis integrator and the FMM can be an efficient algorithm when the tensor-vector product involves very high counts of operations. In §3 a pair-potential model for fluid membrane is introduced and we exhibit the full CGMD simulations in LAMMPS. We also discuss the methodology of accounting for the hydrodynamic interaction in this work. The dynamics of vesicle and RBC from LAMMPS simulations are demonstrated and we compare with previous simulation results [67, 101] where the solvent and internal fluid are coarse-grained. In §3.5 we propose a new approach of membrane energy model where an action field induced from lipid-lipid interactions and the energy variation can capture the membrane dynamics and fusion formation in nano scale. Within this approach, we adopt the IEM coupling with the use of the QBX and the FMM methods to solve the action field numerically and integrate the energy variation to acquire the lipid dynamics. Finally, §5 includes an overall conclusion of this dissertation research and discussion for potential future works. We provide the input script and detail of setting up the case studies in LAMMPS in §A.

CHAPTER 2

COARSE-GRAINED DNA IN LINEAR FLOWS

2.1 Brownian Dynamic Simulation of a DNA Molecule with HI

The DNA or polymer macromolecule is coarse-grained into a system of N beads described by the Langevin equation [24] with hydrodynamic interactions. The governing equation for the position vector $\mathbf{r}_i$ of the ith bead is

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \sum_j \zeta_{ij} \cdot \left(\mathbf{v}_j - \frac{d\mathbf{r}_j}{dt} \right) + \mathbf{F}_i + \sqrt{2} \sum_j \sigma_{ij} \cdot W_j, \tag{2.1}$$

where m_i is the mass of bead i, $\mathbf{v}_j$ is the solvent velocity, and ζ_{ij} is the friction coefficient tensor. The coefficient matrix σ connects the thermal fluctuations of the particles through hydrodynamic interactions. In the Ermak-McCammon model [24], it is related to ζ with $\zeta = \sigma^\top \sigma / k_B T$, where $k_B T$ is the thermal energy. W_j is the thermal fluctuation modeled as a Wiener process with mean 0 and variance dt. Thus, the RHS of Equation (2.1) is the total force acting on the bead i including the drag force, total inter-particle force and the thermal fluctuating HI.

Ignoring the bead inertia, Equation (2.1) can be written as a first-order stochastic differential equation (SDE):

$$d\mathbf{r}_i = \left(\kappa \cdot \mathbf{r}_i + \sum_{j=1}^{N} \frac{\partial D_{ij}}{\partial \mathbf{r}_j} + \sum_{j=1}^{N} \frac{D_{ij} \cdot \mathbf{F}_j}{k_B T} \right) dt + \sqrt{2} \sum_{j=1}^{i} \alpha_{ij} \cdot dW_j, \tag{2.2}$$

where κ is the transpose of the constant velocity gradient tensor of the linear far-field flow velocity and $\mathbf{v}_i = \kappa \cdot \mathbf{r}_i$ ($\mathbf{v}_j = 0$ in a quiescent flow). The random Wiener process in the SDE dW_j is related to dt as: $dW_j = \sqrt{dt}\mathbf{n}_j$ where $\mathbf{n}_j$ is a random vector with the standard Gaussian distribution.

D is the mobility tensor of size $3N \times 3N$ and for the N-bead chain the tensor D is related to the thermal energy through the friction coefficient tensor ζ_{ij} as $\sum_l \zeta_{il} D_{lj} = k_B T \delta_{ij}$. As in [24, 78], we use the RPY tensor for D.

In the absence of external driving forces, the covariance between the bead displacements satisfy the following relation

$$\langle d\mathbf{r}_i d\mathbf{r}_j \rangle = 2 D_{ij} dt. \tag{2.3}$$

Hence, the coefficient matrix α is connected with D via the formula $D = \alpha^\top \alpha$. We remark here that the choice of α is not unique and fast algorithms for generating these correlated random displacements actually take advantage of this fact. Finally, we observe that for the RPY tensor, $\sum_{j=1} \frac{\partial D_{ij}}{\partial \mathbf{r}_j}$ is always zero and Equation (2.2) is reduced to

$$d\mathbf{r}_i = \left(\kappa \cdot \mathbf{r}_i + \sum_{j=1}^{N} \frac{D_{ij} \cdot \mathbf{F}_j}{k_B T} \right) dt + \sqrt{2} \sum_{j=1}^{i} \alpha_{ij} \cdot dW_j. \tag{2.4}$$

2.1.1 Nondimensionalization of the SDE (2.4)

The bead-spring chain model is widely used for BD simulations of a DNA molecule. In the bead-spring chain model, the DNA molecule is represented as a chain of N beads of radius a with adjacent beads connected by a spring. Each spring contains $N_{k,s}$ Kuhn steps of length b_k. So the maximum length of each spring is $N_{k,s} b_k$, and the characteristic contour length of the double stranded DNA molecule L is approximately $(N-1) N_{k,s} b_k$ as the size of each bead is much smaller than the length of each spring and thus neglected. We denote the Hookean spring constant by H. The characteristic length l_s is chosen to be $l_s = \sqrt{k_B T / H}$ and the characteristic time t_s is chosen to be $t_s = \zeta_{res}/4H$, where ζ_{res} is the bead resistivity appeared in the RPY tensor (1.3). We scale the length and time by l_s and t_s, respectively and nondimensionalize SDE(2.4) into the following dimensionless form:

$$d\mathbf{r}_i = \left(\kappa \cdot \mathbf{r}_i + \sum_{j=1}^{N} D_{ij} \cdot \mathbf{F}_j \right) dt + \sqrt{2} \sum_{j=1}^{i} \alpha_{ij} \cdot dW_j, \tag{2.5}$$

Here with a slight abuse of notation, we have used the same notation to denote all corresponding dimensionless quantities.

2.1.2 Choices of the Velocity Gradient Tensor κ

We now specify the velocity gradient tensor κ in SDE (2.5) and restrict our attention to the following two linear planar flows. The first one is the extensional flow where $v_x = \dot{\epsilon}x, v_y = -\dot{\epsilon}y$ with $\dot{\epsilon}$ the extension rate. The second is the shear flow where $v_x = \dot{\gamma}y, v_y = 0$ with $\dot{\gamma}$ the shear rate. We define the Peclet number $Pe = \dot{\epsilon}\zeta/4H$ for the extensional flow and $Pe = \dot{\gamma}\zeta/4H$ for the shear flow, respectively. Then the dimensionless velocity gradient tensor κ in SDE (2.5) is given by the following formulas:

$$\kappa_{ext} = \begin{pmatrix} Pe & 0 & 0 \\ 0 & -Pe & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \kappa_{shear} = \begin{pmatrix} 0 & Pe & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{2.6}$$

Here $\kappa = \kappa_{ext}$ for the extensional flow and $\kappa = \kappa_{shear}$ for the shear flow.

2.1.3 Specification of the Forcing Term $\mathbf{F_i}$

The force $\mathbf{F_j}$ in SDE (2.5) contains two parts: the force exerted by the connected springs and the force due to the finite size of the beads. We adopt the Marko-Siggia's wormlike chain (WLC) spring law [57] to model the spring force between beads. In the WLC model, the dimensionless spring force acting on the ith bead by the ith spring is

$$\mathbf{F}_i^s = \sqrt{\frac{N_{k,s}}{3}} \left[\frac{1}{2} \frac{1}{\left(1 - \frac{Q_i}{Q_0}\right)^2} - \frac{1}{2} + \frac{2Q_i}{Q_0} \right] \frac{\mathbf{Q}_i}{Q_i}, \tag{2.7}$$

where $i = 1, \ldots, N-1$, $\mathbf{Q}_i = \mathbf{r}_{i+1} - \mathbf{r}_i$ is the distance vector between bead $\mathbf{r}_{i+1}$ and $\mathbf{r}_i$, Q_i is the length of $\mathbf{Q}_i$, and Q_0 is the maximum distance between these two beads. Since all interior beads are connected with two springs from two sides, the net entropic spring force acting on the ith bead is

$$\mathbf{F}_i^{\text{entropy}} = \mathbf{F}_i^s - \mathbf{F}_{i-1}^s, \quad \mathbf{F}_0^s = \mathbf{F}_N^s = 0, \tag{2.8}$$

with $i = 1, \ldots, N$. For later use, we also record the potential for the ith spring below

$$U_{WLC}(\mathbf{Q}_i) = \frac{1}{2}\sqrt{\frac{N_{k,s}}{3}}\left(\frac{Q_0^2}{Q_0 - Q} - Q + \frac{2Q^2}{Q_0}\right). \tag{2.9}$$

For the force due to the finite size of the beads, we adopt the excluded volume force in [71, 78] given by the formula

$$\mathbf{F}_i^{EV} = -\sum_{j=1, i\neq j}^{N} \frac{9\sqrt{3}z}{2}\exp\left(-\frac{3r_{ij}^2}{2}\right)\mathbf{r}_{ij} \tag{2.10}$$

where $z = \left(\frac{1}{2\pi}\right)^{3/2}\tilde{v}N_{k,s}^2$, and $\tilde{v} = 2ab_k^2/l_s^3$ is the dimensionless excluded volume parameter. And the excluded volume potential between bead i and bead j is given by

$$U_{ij}^{EV} = \frac{3\sqrt{3}z}{2}\exp\left(-\frac{3r_{ij}^2}{2}\right). \tag{2.11}$$

Finally, the total force acting on bead i is the sum of the spring forces and the excluded volume forces, that is,

$$\mathbf{F}_i = \mathbf{F}_i^{\text{entropy}} + \mathbf{F}_i^{EV}. \tag{2.12}$$

2.2 Numerical Algorithm for BD Simulations in Linear Flows

In the past, a semi-implicit predictor-corrector scheme [39, 78, 87] is often used for the temporal integration in BD simulations. A major problem associated with that scheme is that a very small time step size has to be used in order to avoid the numerical instability, which leads to an excessively large number of time steps and a very long total simulation time. Recently, a Metropolis integrator has been developed to integrate the self-adjoint diffusion equations [12] for BD simulations in a quiescent flow.

Here we extend the algorithm in [12] to study BD simulations in linear flows. We first introduce an integrating factor $e^{-\kappa t}$ and rewrite SDE (2.5) as follows:

$$d\big(e^{-\kappa t}\mathbf{r}_i\big) = \mathrm{e}^{-\kappa t}\Big[D(\mathbf{r}_i)\mathbf{F}(\mathbf{r}_i)dt + \sqrt{2}\alpha(\mathbf{r}_i)dW_i\Big]. \tag{2.13}$$

Let $K(t) = \exp(-\kappa t)$. We now introduce a new variable $\mathbf{x}_i = K(t)\mathbf{r}_i$ (i.e., $\mathbf{r}_i = K(-t)\mathbf{x}_i$). Then the original SDE (2.13) can be rewritten in terms of $\mathbf{x}_i$ as follows:

$$d\mathbf{x}_i = K(t)\Big[D(K(-t)\mathbf{x}_i)\mathbf{F}(K(-t)\mathbf{x}_i)dt + \sqrt{2}\alpha(K(-t)\mathbf{x}_i)dW\Big]. \tag{2.14}$$

The generator of SDE (2.14) is given by the following formula:

$$Lf(\mathbf{x}_i) = \frac{1}{\nu(\mathbf{x}_i)}\mathrm{div}(\nu(\mathbf{x}_i)K(t)D(K(-t)\mathbf{x}_i)K(t)^T Df(\mathbf{x}_i)) \tag{2.15}$$

where we introduce the stationary density given by

$$\nu(\mathbf{x}_i) = \exp(-U(K(-t)\mathbf{x}_i)). \tag{2.16}$$

We denote the total energy by U which is the sum of WLC spring energy U_{WLC} and EV potential energy U^{EV}. It is easy to see that the generator of the transformed stochastic differential equation with respect to $\nu(\mathbf{x}_i)$ is self-adjoint. Thus the algorithm in [12] can be directly applied to (2.14). We now update the position vector as follows:

1. Compute the vector $\tilde{\mathbf{x}}_i^{n+1}$ and update $\mathbf{x}_i^n$ by the following formulas:

$$\tilde{\mathbf{x}}_i^{n+1} = \mathbf{x}_i^n + K(t_n)G(K(-t_n)\mathbf{x}_i^n)\Delta t + \sqrt{2\Delta t}K(t_n)B(K(-t_n)\mathbf{x}_i^n)dW_i, \tag{2.17}$$

where the functions G and B are defined by the formulas:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x} + \frac{2}{3} D(\mathbf{x})\mathbf{F}(\mathbf{x})\Delta t, \\ G(\mathbf{x}) &= \frac{5}{8} D(\mathbf{x})\mathbf{F}(\mathbf{x}) - \frac{3}{8} D(\mathbf{x})\mathbf{F}(\mathbf{x}_1) - \frac{3}{8} D(\mathbf{x}_1)\mathbf{F}(\mathbf{x}) + \frac{9}{8} D(\mathbf{x}_1)\mathbf{F}(\mathbf{x}_1), \end{aligned} \tag{2.18}$$

$$\begin{aligned} \mathbf{x}_2 &= \mathbf{x} - \frac{2}{3} D(\mathbf{x})\mathbf{F}(\mathbf{x})\Delta t, \\ B(\mathbf{x})B(\mathbf{x})^\top &= \frac{1}{4} D(\mathbf{x}) + \frac{3}{4} D(\mathbf{x}_2). \end{aligned} \tag{2.19}$$

We then apply the Metropolis integrator to obtain the updated $\mathbf{x}_i^{n+1}$.

2. Calculate the acceptance probability α as follows:

$$\alpha(\mathbf{x}_i^n, \tilde{\mathbf{x}}_i^{n+1}) = \min\left(1, C \exp\left[-\frac{|d\tilde{W}_i|^2}{2} + \frac{|dW_i|^2}{2} - U(\tilde{\mathbf{x}}_i^{n+1}) + U(\mathbf{x}_i^n) \right]\right), \tag{2.20}$$

where $C = \det B(\mathbf{x}_i^n) / \det B(\tilde{\mathbf{x}}_i^{n+1})$, $U = U_{WLC} + U^{EV}$ is the total potential energy, and $d\hat{W}_i$ is obtained via the formula

$$B(\tilde{\mathbf{x}}_i^{n+1}) d\tilde{W}_i = B(\mathbf{x}_i^n) dW_i + \sqrt{2\Delta t}\, G(\tilde{\mathbf{x}}_i^{n+1}). \tag{2.21}$$

3. Generate a Bernoulli random number γ, that is, generate a uniformly distributed random number β on $[0, 1]$ and set γ to 1 if $\beta \leq \alpha$ and 0 otherwise.

4. Compute the updated position vector at time $t = t_{n+1}$ by the formula

$$\mathbf{r}_i^{n+1} = \gamma K(-t_{n+1}) \tilde{\mathbf{x}}_i^{n+1} + (1 - \gamma)\mathbf{r}_i^n \tag{2.22}$$

In other words, the position vector will be updated only if the Bernoulli random number γ is equal to 1. This is the essence of the Metropolis algorithm for Monte-Carlo simulations.

2.3 Numerical Results

Common measures of the "stretch" of a DNA molecule under flow are the molecular fractional extension ($\hat{x}$ is the unit vector in the x direction)

$$X \equiv \max_i(\mathbf{r}_i \cdot \hat{x}) - \min_i(\mathbf{r}_i \cdot \hat{x}), \tag{2.23}$$

and its ensemble average $\langle X \rangle \equiv \frac{1}{M}\sum X$, where M is the total number of experiments (or simulations). Here we first compare the transient fractional extensions of a λ-DNA between the experimental data, semi-explicit numerical simulations [78], and our Metropolis scheme simulations. The initial DNA configurations in these simulations are the equilibrium DNA configurations in the absence of flow from the Metropolis scheme.

For the purpose of comparison, we use the same values of physical and model parameters as in [78]. That is, the viscosity η of solvent is 8.4 $cP(= mPa \cdot s)$ and the relaxation time τ is 21.0 seconds. The λ-DNA is modeled with $N = 29$ beads of radius $a = 0.101$ μm connected by 28 springs, where each spring has $N_{k,s} = 40$ Kuhn steps of size $b_k = 0.132$ μm and the contour length L is 150 μm. Finally, the excluded volume parameter $v = 0.0034$ μm^3.

To mimic the experimental configurations, it is essential [78] to first simulate the DNA molecule to its equilibrium in a quiescent flow, i.e., $\kappa \cdot \mathbf{r}_i = 0$ in SDE (2.5), which is now a self-adjoint stochastic differential equation that can be efficiently solved to an equilibrium state using the Metropolis scheme in Section 2.2. At the beginning of the no-flow simulations, beads are equally spaced on the x-axis. The Metropolis scheme allows for relatively large time step $\triangle t$ (an order of magnitude larger), consequently saving a significant amount of computation time for running no-flow simulations compared to the semi-implicit predictor-corrector scheme in [78]. The flow-free simulation is continued until an equilibrium configuration is reached, which is often 10-20 relaxation times (τ). After the equilibrium is reached for a DNA in a quiescent flow, we then turn flow on in the simulations and sum up $d\mathbf{r}_i$ to obtain the updated configuration and the mean fractional extension of a DNA molecule under linear flow.

In Figures 2.1–2.6, we use the Deborah number to label different flows. As a dimensionless flow strength, the Deborah number De is equal to $\dot{\epsilon}\tau$ for the extensional flow and $\dot{\gamma}\tau$ for the shear flow. We would like to remark here that $\dot{\gamma}\tau$ is also called the Weissenberg number in the case of the shear flow in many literature. The transient fractional extension from these simulations is summarized in Figure 2.1, which shows two sets of comparison for Deborah number $De = 0.98$ ($\dot{\epsilon} \approx 0.0467\ \text{sec}^{-1}$) and $De = 4.0$ ($\dot{\epsilon} \approx 0.1905\ \text{sec}^{-1}$) for panels (a) and (b), respectively. Figure 2.1 is simulated by using the modified Metropolis integrator scheme with an integrating factor (Equation (2.5) in Section 2.2, $\kappa = \kappa_{ext}$). Thin curves are individual trajectories from experiments, filled circles are the ensemble average from experiments, filled triangles are ensemble average from Schroeder *et al.* [78], and our results are the empty triangles. We observe that, in both panels, our results are in good agreement with the experiment results. However, our simulations are orders of magnitude more efficient because a time-step $\Delta t = 10^{-4}\tau = 2.1 \times 10^{-3}$ sec is used for results in panels (a), and $\Delta t = 10^{-3}/\dot{\epsilon} = 5.25 \times 10^{-3}$ sec is used for panel (b). In comparison, a much smaller time step for $De = 4.0$ and $De = 0.98$ cases are necessary for the predictor-corrector scheme [78]. The advantage of using Metropolis integrator is to capture the physical phenomena of the model. Error bars in the figures denote the standard deviation calculated from our numerical simulation data (triangles). No error bars are provided for the data from the experiment (filled circles) and Schroeder's simulation (empty circles).

Similar comparison of a single DNA molecule in a planar extensional flow between experiment and simulation are also conducted in [42]. Figure 2.2 compares our results against those from [86] for a 21 μm DNA molecule in an extensional flow

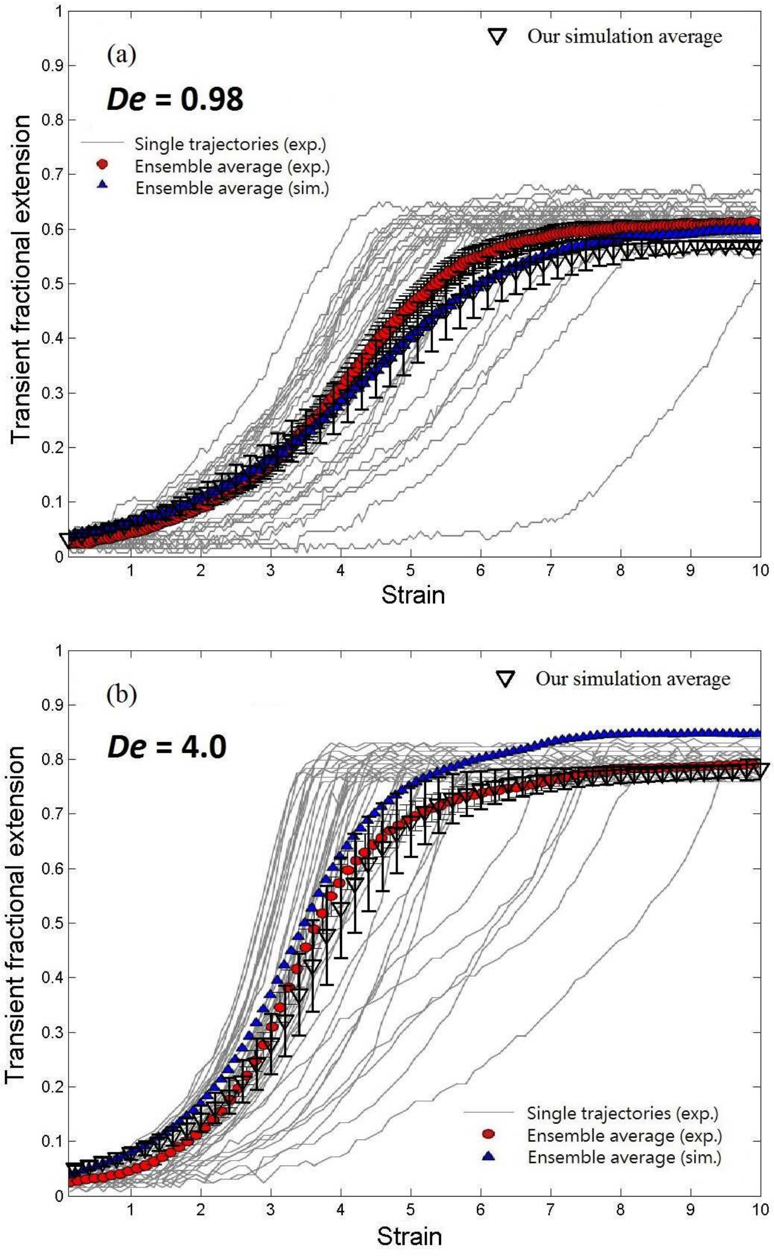


Figure 2.1 Transient fractional extension for a 7-lambda ($L = 150\ \mu m$) DNA in a planar extensional flow. 60 trajectories from simulations are used for ensemble average.

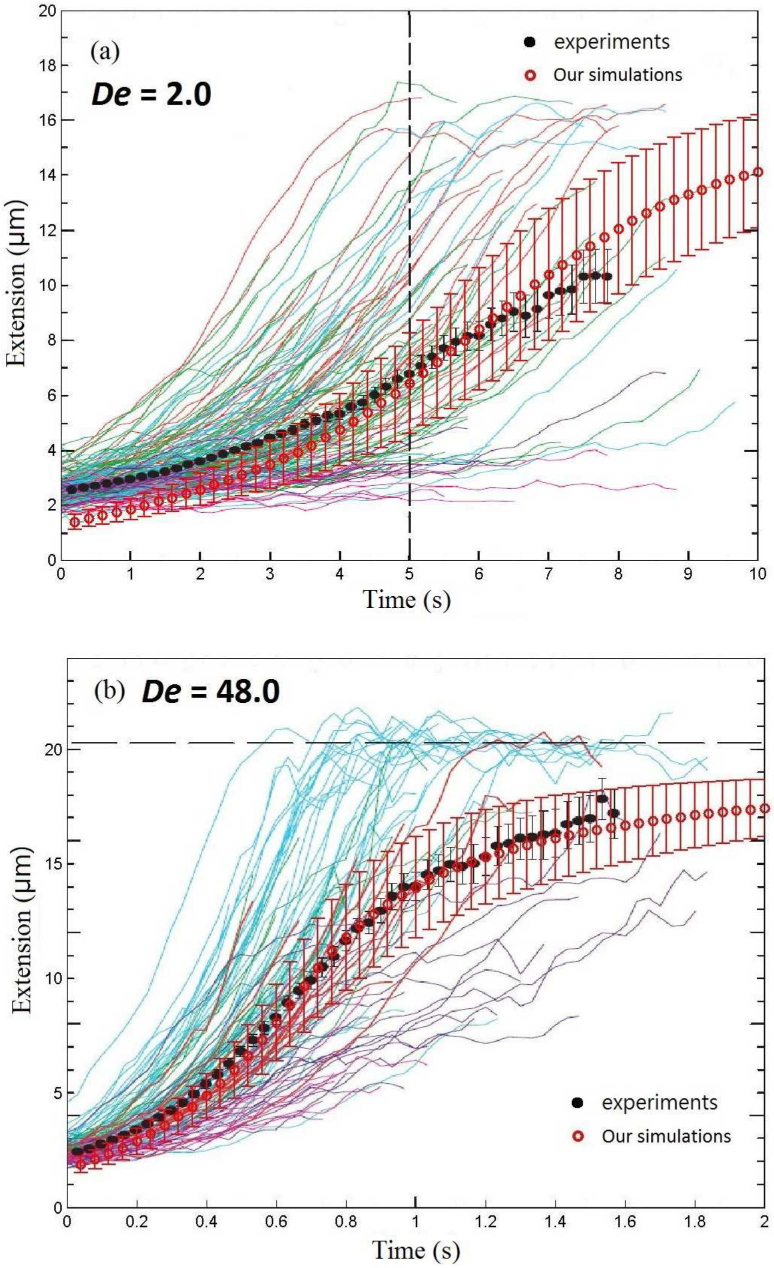


Figure 2.2 Comparison between experiments [86] (thin curves for individual trajectories and filled circles for the average) and our numerical simulations (empty circles). The vertical dashed line in figure 2(a) shows the point below which continuous data could not be collected in some experiments. The horizontal dashed line in figure 2(b) shows the steady-state of the stretched $\sim 22\ \mu m$ λ-DNA.

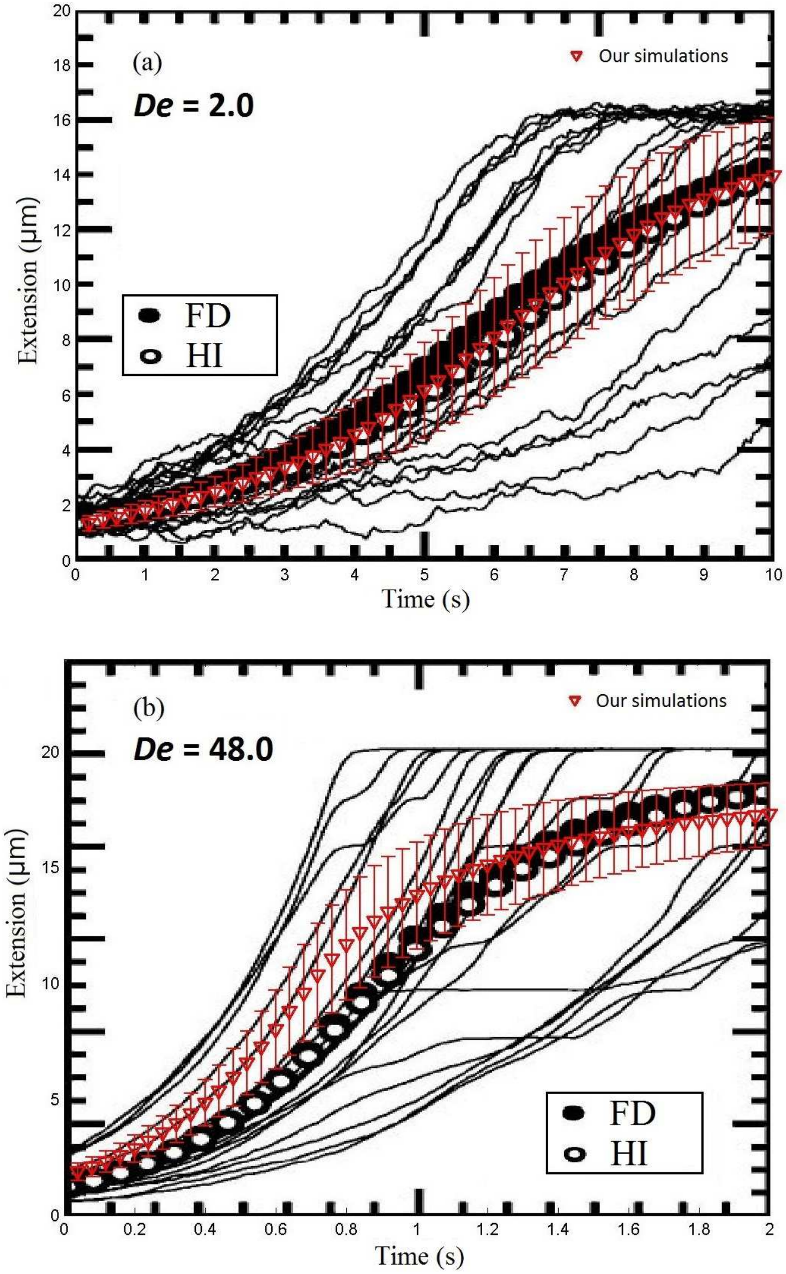


Figure 2.3 Comparison between numerical results [42] (filled and empty circles for the assemble averages of FD and HI) and our numerical simulations (empty triangles). Solid curves are single trajectories of HI simulations from [42].

with $N = 11$, $b_k = 0.106\ \mu m$, $N_{k,s} = 19.8$, $a = 0.077\ \mu m$, 23°C for the temperature and $v = 0.0012\ \mu m^3$. Figure 2.2(a) is for $De = 2.0$, $\dot{\epsilon} = 0.5\ \text{sec}^{-1}$, $\tau = 4.1$ sec and $\eta = 43.3$ cP. Figure 2.2(b) is for $De = 48.0$, $\dot{\epsilon} = 2.8\ \text{sec}^{-1}$, $\tau = 17.3$ sec and $\eta = 182$ cP. Thin curves are trajectories from experiments [86], filled circles are the ensemble average of experimental results, and empty circles are the ensemble average from our modified Metropolis integrator simulations. For $De = 2.0$ (Figure 2.2(a)) our average is almost identical to the simulation average from [42] (bottom panel of their Figure 2). For $De = 48.0$ (Figure 2.2(b)), Our simulation results are in better agreement with experimental results than those from Jendrejack *et al.* [42] and we show these comparisons in Figure 2.3. In these Metropolis integrator simulations $\Delta t = 10^{-3}$ sec for both $De = 2.0$ in Figure 2.3(a) and $De = 48.0$ in Figure 2.3(b). Even though this time step is slightly smaller than those used in [42], our Metropolis algorithm with the integrating factor is second-order accurate [11, 12] and no matrix inversion is needed. In Section 2.4, we describe how our numerical algorithm can be further improved when the system size is large by using FMM to efficiently calculate the HI.

Next we compare the mean fractional extension of a DNA molecule against experiments [85] and Jendrejack *et al.*'s simulations [42]. The parameters for simulations are [85]: bead radius $a = 0.077\ \mu m$, and temperature is fixed at 20°C. Two viscosities are considered in the experiments, $\eta = 60$ cP and 220 cP for the shear flow cases, while only $\eta = 60$ cP is used for the case of extensional flow (based on the experiments in [85]). For the corresponding simulations in [42] the number of beads is 11, Kuhn step size $b_k = 0.106\ \mu m$, the number of springs per Kuhn step $N_{ks} = 21$, and the contour length $L = 22\ \mu m$.

Figure 2.4 shows the fractional extension versus time for three cases of Deborah numbers ($De = 3.2,\ 6.3,$ and 76.0) when DNA molecule is under the simple shear flow. Since the relaxation time τ is fixed, shear rate $\dot{\gamma}$ is higher at higher De. As

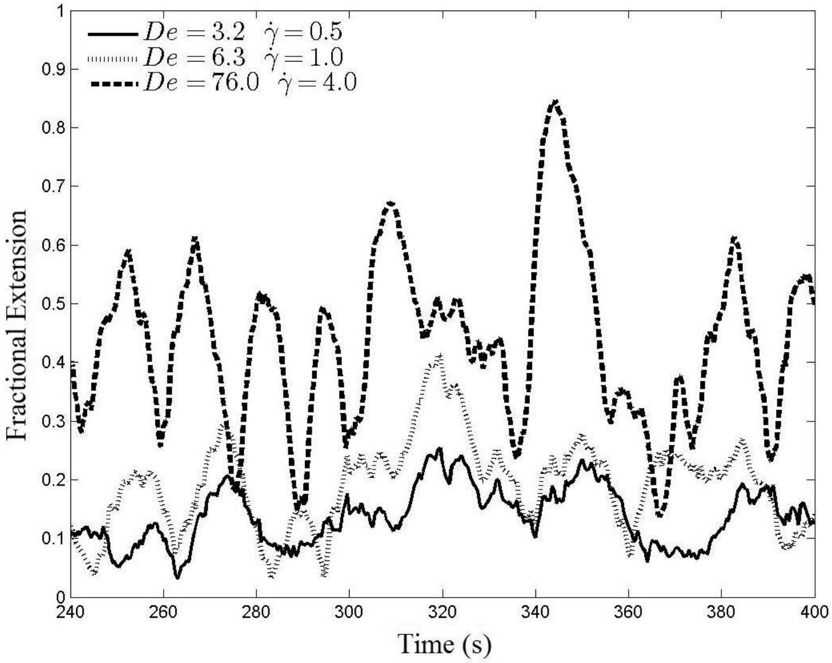


Figure 2.4 Fractional extensions for different De and shear rate $\dot{\gamma}$ from our simulations with $(De, \dot{\gamma}) = (3.2, 0.5)$, $(6.3, 1.0)$, $(76.0, 4.0)$, respectively. The relaxation time τ is 6.3 sec for the first two cases and 19.0 sec for the third case. The time steps are: $\Delta t = 10^{-3}$ sec for $De = 3.2$, $\Delta t = 5 \times 10^{-4}$ sec for $De = 6.3$, and $\Delta t = 2.5 \times 10^{-4}$ sec for $De = 76.0$.

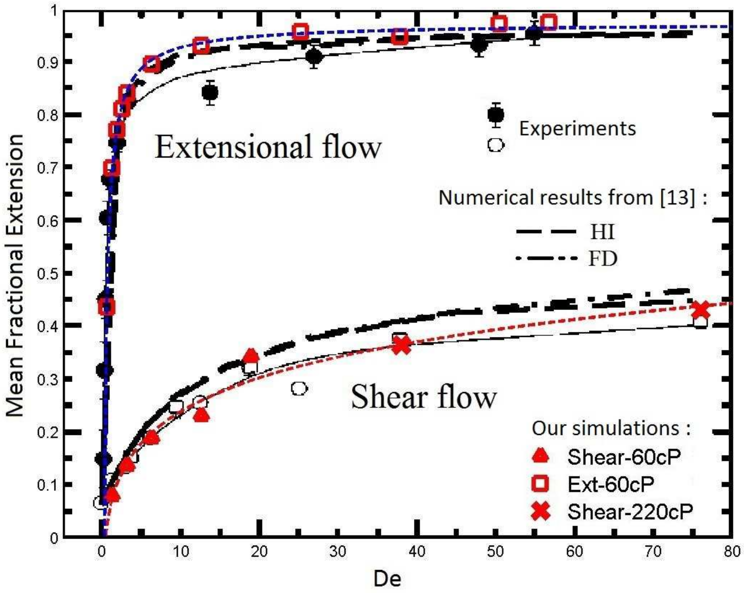


Figure 2.5 Cases of mean fractional extensions for shear flow and extensional flow. Experimental data [85] are symbols with error bars, bead model with and without HI (see legend) are from [42] and our results as red symbols (empty circles for the extensional flow; triangles and crosses for the shear flow).

expected, larger mean extension of the DNA molecule is expected at a higher shear rate. From these results, the mean fractional extension is computed by taking the averages over a long duration.

Figure 2.5 shows the comparison of mean fractional extension between experiments [85], Jendrejack *et al.*'s simulations [42] and our simulations. Experimental data are shown in filled dark disks for the extensional flow and dark circles for the shear flow, and the thin solid curves are their best fits. Simulation results from [42] are thick dashed (with HI) and dash-dotted (without HI, or free-draining (FD)) curves. Our simulation results are denoted by red symbols in the legends, and their best fits are the thin dashed curves. It is clear that our results agree well with experimental data for the shear flow cases. For the extensional flow cases, our results agree better with simulation results from [42] for all values of De. At larger De (De $\geq$ 40), all three agree well for the extensional flow cases.

Schroeder *et al.* also investigated the hysteresis of stretch-coil transition of a long λ-DNA ($\sim 1300\mu$m) in an extensional flow [77]. Figure 2.6 shows the comparisons of single trajectories of DNA extensions over strains between their experimental data [77] and our numerical results. For the initially coiled DNA, the simulation starts without flow for several relaxation times as done in previous numerical simulations [77]. Also following their procedure for the initially stretched DNA , we first run the simulations with a high Deborah number ($De = 15$) until equilibrium, and then gradually lower the flow strength (Deborah number) until the desired values are reached (at $t = 0$): $De = 0.30$ for Figure 2.6(a) and $De = 0.57$ for Figure 2.6(b). The parameters for simulations are the following: $a = 0.28$ μm, $N_s = 123$, $N_{ks} = 80$, $b_k = 0.132$ μm, $\nu = 0.00032$ μm^3 and $\tau = 126.0$ sec. We set the solvent viscosity to be 1 cP and time step size (a) $\Delta t = 10^{-2}$ sec and (b) $\Delta t = 5 \times 10^{-3}$ sec are used for simulations. The agreement with the experimental data demonstrates that our numerical methods are able to capture the hysteric transition between stretched and coiled DNA in an extensional flow.

2.4 Extension to Large Systems

In the numerical algorithm described in Section 2.2, the RPY tensor D is constructed explicitly, the matrix vector product $D\mathbf{F}$ is computed directly, and the uppertriangular matrix B is obtained by the Cholesky decomposition with its determinant simply the product of its diagonal entries. This is affordable for the numerical experiments presented in Section 2.3 since the total number of beads $N = 29$. However, for large systems the computational cost of these standard direct algorithms becomes prohibitively expensive since the matrix vector product $D\mathbf{F}$ requires $O(N^2)$ operations, the Cholesky factorization requires $O(N^3)$ operations, and each BD simulation often requires more than 10^5 time steps. Thus, fast algorithms become a necessity in order to make long-time large-scale BD simulations practical.

As mentioned in Section 1.1, recently a fast multipole method for the RPY tensor (RPYFMM) has been developed in [53]. The fundamental observation in [53] is that the RPY tensor can be decomposed as follows:

$$D_{ij} = C_1 \left[\frac{\delta_{ij}}{|\mathbf{x}-\mathbf{y}|} - (x_j - y_j) \frac{\partial}{\partial x_i} \frac{1}{|\mathbf{x}-\mathbf{y}|} \right] + C_2 \frac{\partial}{\partial x_i} \frac{x_j - y_j}{|\mathbf{x}-\mathbf{y}|^3},$$

where $C_1 = \frac{k_B T}{8\pi\eta}, C_2 = \frac{k_B T a^2}{12\pi\eta}$.

With this decomposition, the matrix vector product $D\mathbf{v}$ for a given vector $\mathbf{v}$ can be interpreted as a linear combination of four harmonic sums with suitably chosen source charges and dipoles. In other words, the matrix vector product $D\mathbf{v}$ can be evaluated by four calls of the classical FMM for Coulomb interactions in three dimensions [16]. Thus, the RPYFMM avoids the explicit construction of the RPY tensor and reduces the computational cost of $D\mathbf{v}$ to $O(N)$ in both CPU time and memory storage.

We observe further that the Cholesky factor B of the RPY tensor D can be replaced by any matrix C which satisfies the same matrix equation $CC^\top = D$ (note that there are actually infinitely many matrices satisfying this matrix equation, see, for example, [44] for details). Indeed, [53] also proposed to replace the Cholesky factor B by $\sqrt{D}$ and compute $\sqrt{D}\mathbf{v}$ by combining the classical Spectral Lanczos Decomposition Method (SLDM) with the RPYFMM. The resulting algorithm has $O(\kappa N)$ complexity with κ the condition number of the RPY tensor D. We remark here that for most BD simulations with HIs, the beads do not overlap with each other due to the EV force and our numerical experiments show that the condition number of the RPY tensor in this case is fairly low. This indicates that the RPYFMM-SLDM method is essentially a linear algorithm for computing $\sqrt{D}\mathbf{v}$. The timing results presented in Table 4.1 and Table 4.2 clearly demonstrate of linear scaling of the RPYFMM and RPYFMM-SLDM methods.

Table 2.1 Timing Results (sec) for Computing $T = D\mathbf{v}$ by RPYFMM

N	T_{RPYFMM}	T_{Direct}	E_{RPYFMM}
1,000	0.20897	0.31495	1.6008e-02
10,000	1.6058	30.6643	5.5339e-02
100,000	16.172	2738.48	8.3803e-02
1,000,000	160.24	271009.4	1.1603e-01

Table 2.2 Timing Results (sec) for Computing $T = \sqrt{D}\mathbf{v}$ by RPYFMM-SLDM.

N	m	T_{SLDM}	$E_{relative}$
1,000	4	0.54192	6.21032e-06
10,000	4	9.03360	6.24604e-04
100,000	6	111.80	7.92857e-04
1,000,000	12	2180.8	2.91239e-04

Finally, we would like to remark here that recent developments in the fast multipole methods and fast direct solvers also enable a linear algorithm for computing the determinant of a matrix with certain hierarchical low rank structure [2,35,36]. By incorporating all these fast methods into our current numerical scheme, we obtain a numerical algorithm which is stable even for relatively large time step size and scales linearly with respect to the number of particles (or beads) in the system.

Figure 2.7 and Figure 2.8 are simulation snapshots of many DNA in an oscillatory shear flow. Similar to the previous work [89] we define the background oscillatory shear as $\mathbf{U}_0 = (\dot{\gamma}\sin(2\pi\omega t)y, 0, 0)$ (on the $x - y$ plane in each panel), where $\omega = (20000\Delta t)^{-1}$, $\Delta t = 0.00128$ and $\dot{\gamma} = 1.0$ is the shear rate for simulations. For this simulation we include 25 DNA molecules, each of which has a rest contour length of 150 μm. The parameter set of 150 μm long DNA (29 beads) is used (total number of beads is 725). In the simulation, each molecule has the same initial extension. Figure 2.9 illustrates the correlation between mean molecule extension and the oscillating shear flow magnitude. Figure 2.10 shows that the spring energy dominates the total energy during the first period ($t = 25.6$) since the DNA molecules stretch under shear flow. After one period of time, DNA molecules turn to coiled states and EV potential energy is dominant due to molecule-molecule interactions.

Finally, the timing result for one time step Δt including forcing calculations and matrix-vector multiplications with the use of RPYFMM and RPYFMM-SLDM is $\sim$ 0.3 sec. However, it takes $\sim$ 98.0 sec at each time step for direct calculations involving construction of RPY tensor and direct factorization using Cholesky decomposition. This result shows that one can reduce much computational cost by using the $O(N)$-operation algorithms when N is large.

2.5 Conclusion

We have extended the Metropolis integrator in [12] to study BD simulations with HIs in linear flows. The method utilizes the integraing factor to absorb the effect of the linear flow and permits much larger time step sizes for BD simulations with HIs in linear flows. We have applied our method to study the fractional stretch and the mean stretch of a single λ-DNA molecule in planar linear flows. Our numerical results agree very well with experimental data [85, 86] and other simulation results [78] in the literature.

We have also discussed the extension of our method to large systems in Section 2.4. By incorporating the RPYFMM and other fast algorithms into the scheme, the resulting algorithm admits large time step sizes and has nearly optimal complexity (i.e., $O(N)$ or $O(N \log N)$) in the number of particles in the system. Thus, even though many of these fast algorithms have a large prefactor (say, $C \geq 1000$) in front of N, the combination of our fast algorithm with modern computers makes long-time large-scale BD simulations with HIs within practical reach. We are currently incorporating these fast algorithms into the modified Metropolis integrator and applying the resulting algorithm to study the lipid bilayer membrane of the red blood cells in the blood flow. Results from these ongoing work are being analyzed now and will be reported in a timely fashion.

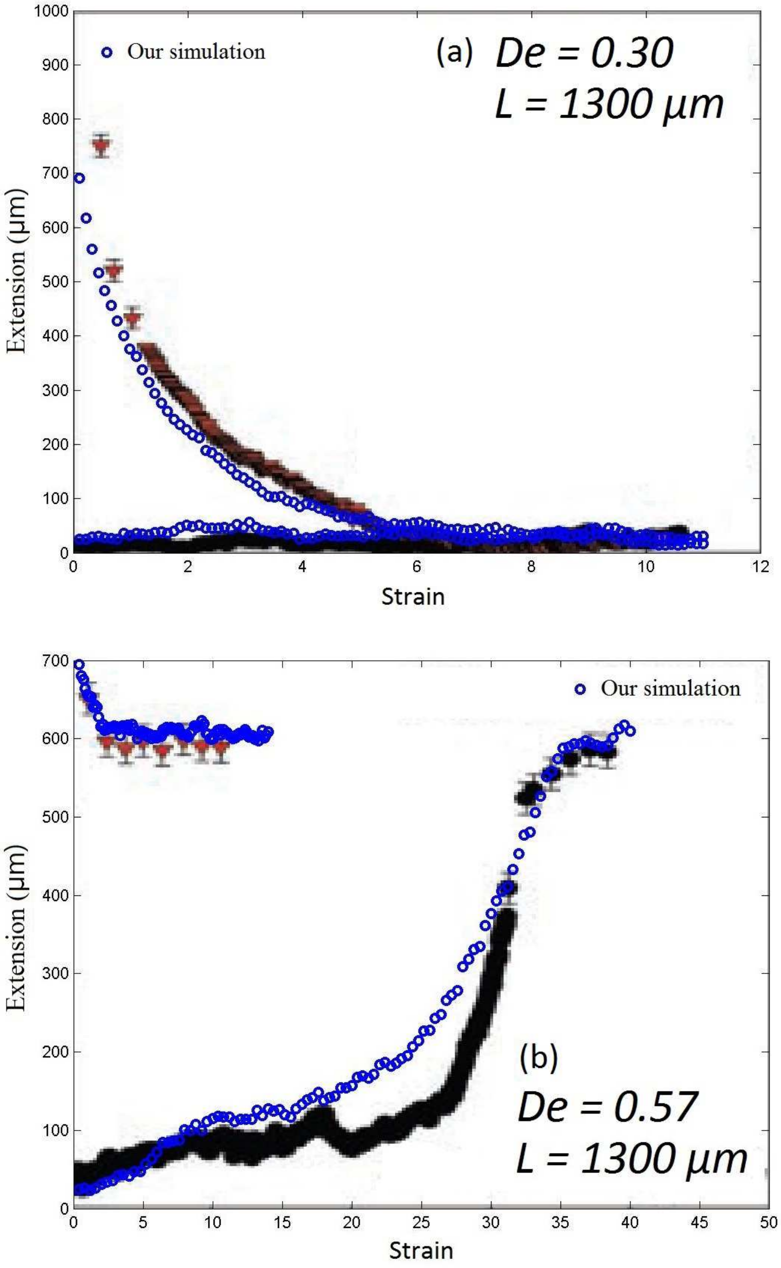


Figure 2.6 Molecular extensions for 1.3 mm DNA in an extensional flow for $De = 0.30$ (top) and $De = 0.57$ (bottom). Filled symbols are experimental data from [77]. Blue circles are our simulation results. Time steps $\Delta t = 5 \times 10^{-2}$ sec for $De = 0.30$ (top), and $\Delta t = 10^{-3}$ sec for $De = 0.57$ (bottom).

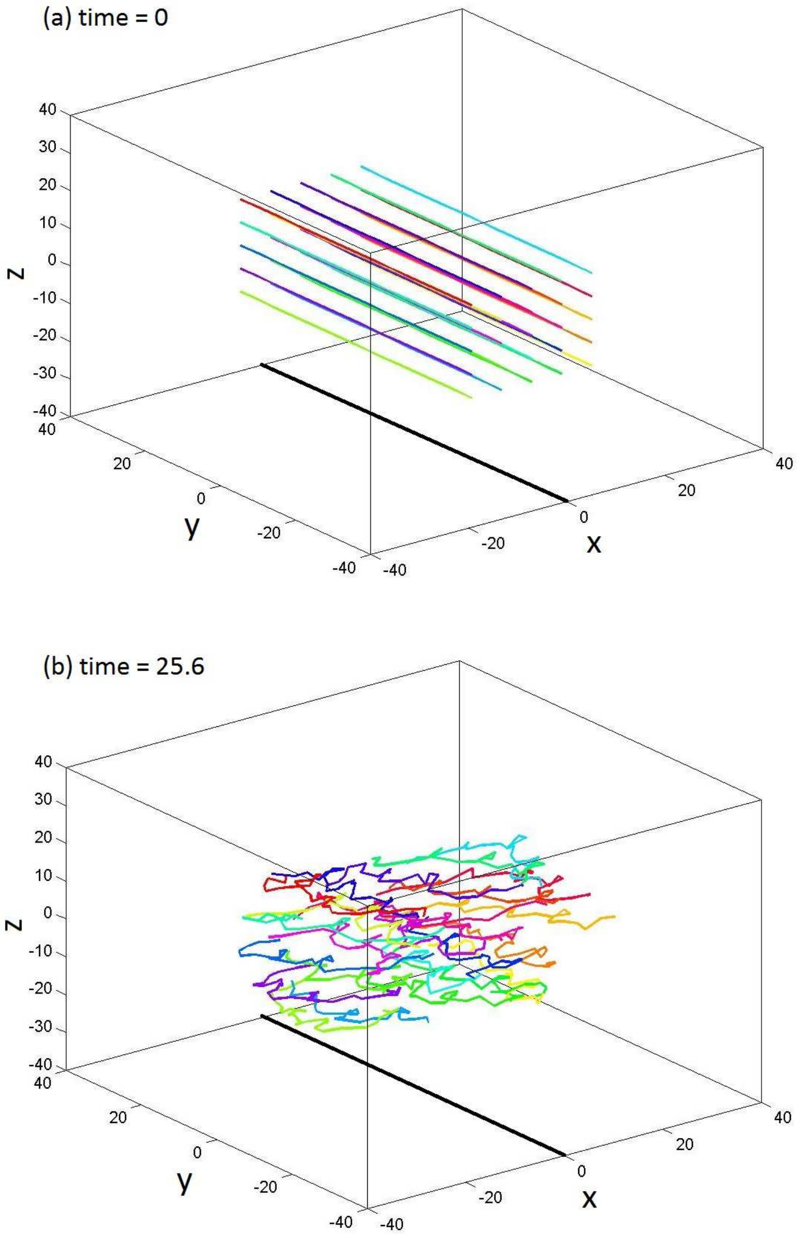
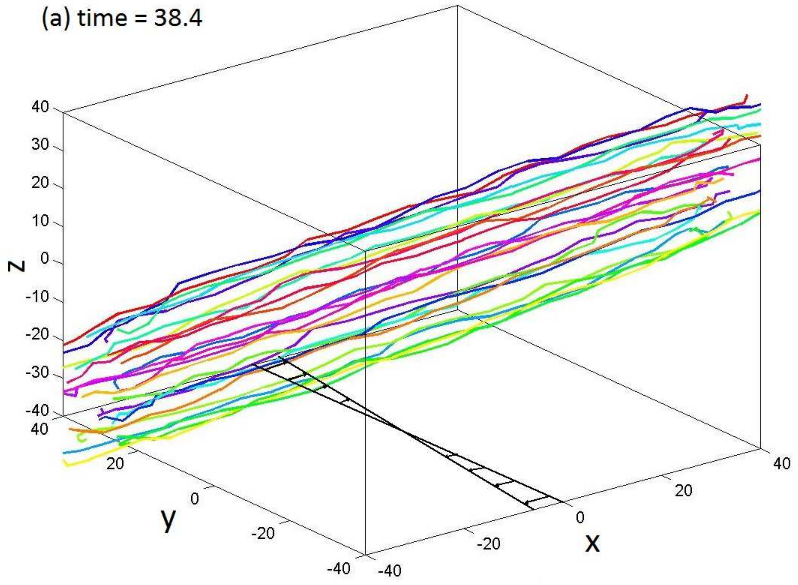


Figure 2.7 Numerical experiments of many-DNA in an oscillatory shear flow at (a) $t = 0$ and (b) $t = 25.6$, when shear flow velocity is zero.



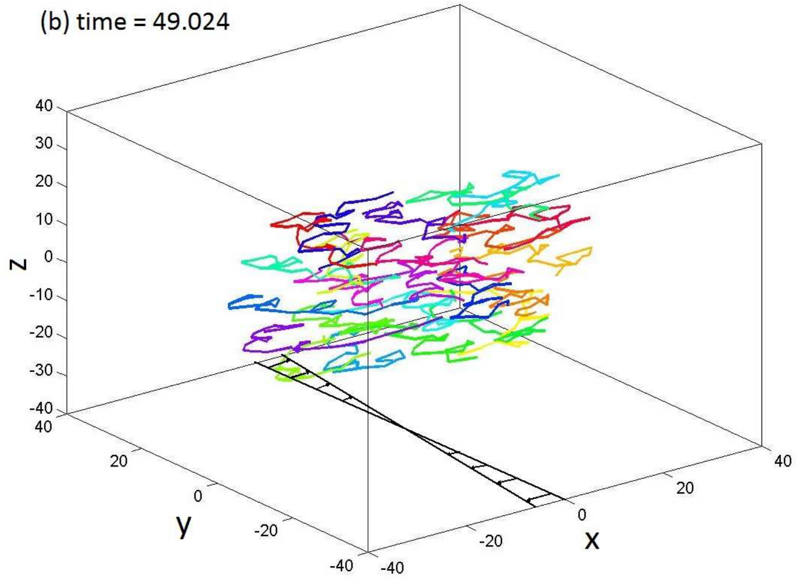


Figure 2.8 Numerical experiments of many-DNA in an oscillatory shear flow at (a) $t = 38.4$ and (b) $t = 49.024$.

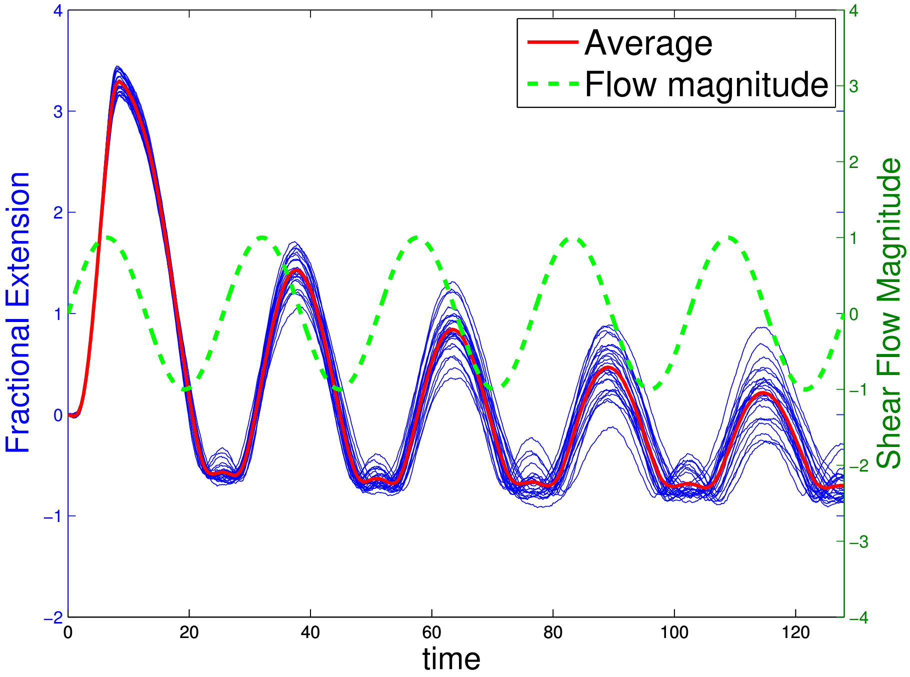


Figure 2.9 Numerical simulations for 25 DNA molecules in an oscillatory shear flow. Blue trajectories are fractional extensions of each molecule, dashed green curve is the magnitude of periodic shear flow and solid red curve is the assemble average.

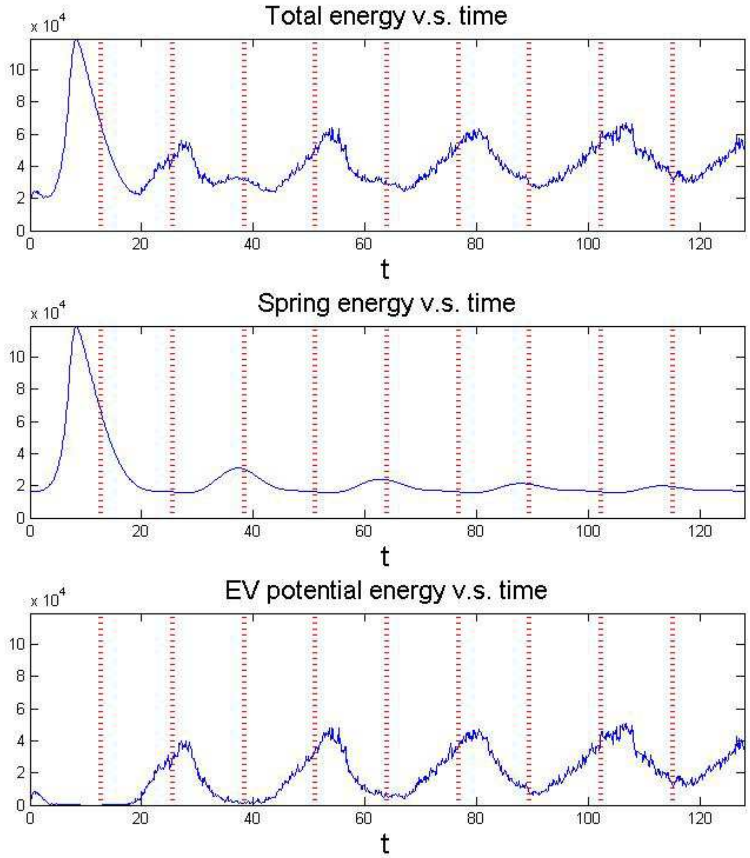


Figure 2.10 Energies versus time t for 25 DNA molecules in an oscillatory shear flow. Total enegy (top) is the sum of WLC spring energy (middle) and EV potential energy (bottom). Red dotted vertical lines represent half period of flow oscillation.

CHAPTER 3

COARSE-GRAINED LIPID BILAYER MEMBRANE – VESICLES AND RED BLOOD CELLS SIMULATIONS IN LAMMPS

3.1 Model Descriptions

3.1.1 Coarse-Grained (CG) Modeling

The advantage of using coarse-grained modeling is to reduce the computational cost. Without losing physical properties of lipid bilayer membrane, characteristic length scale in CG modeling can be much larger than atomistic sizes and simulations can be performed for a much longer time. Figure 3.1 shows that lipid bilayer membrane can be represented as: (a) 1 bead for lipid head and rigid rod for lipid tail [62]; (b) 1 bead for lipid head and 2 beads for lipid tail [96]; (c) 1 bead for a collection of lipids with the bead diameter as the thickness of lipid bilayer membrane [60, 61]. For all LAMMPS simulations in this work, we adopt the coarse-graining in panel (c), the one-particle-thick meshless model where the characteristic length is the diameter of coarse-grained particle (lipid bilayer membrane thickness).

3.1.2 Pair-Potential Model for Coarse-Grained Membranes

In this work, we implement in LAMMPS the lipid-lipid interaction potential function for CGMD simulations of a lipid bilayer membrane. Developed by Yuan *et al.*, the interaction potential between coarse-grained lipid particles is constructed to account for the head-head, tail-tail and head-tail interactions between the coarse-grained lipid mesoscopic molecules [100, 101]. Figure 3.2 shows the schematic of inter-particle interactions, angular parameters and the approximation of spontaneous curvature c_0. d_0 is the average inter-particle distance and the dimensionless spontaneous curvature is $\tilde{c_0} = R_0 c_0$ where R_0 is the radius of spherical body. $\{\theta_i, \theta_j\}$ indicate the orientations of particle pair $\{\mathbf{r}_i, \mathbf{r}_j\}$.

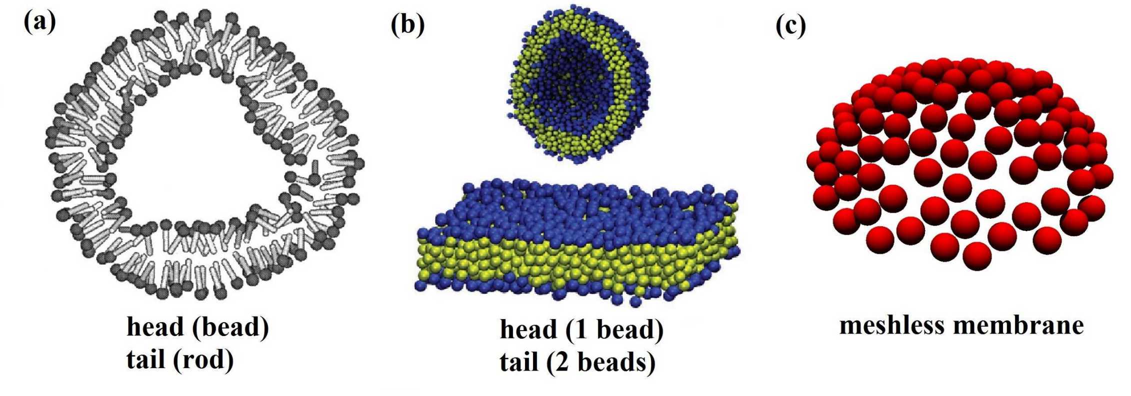


Figure 3.1 Various CG membrane models: (a) 1 bead for lipid head and rod for lipid tail; (b) 1 bead for lipid head and 2 beads for lipid tail; (c) 1 bead for a collection of lipids without mesh. (panel (a) and (b) are reprinted from [96]; panel (c) is reprinted from [60])

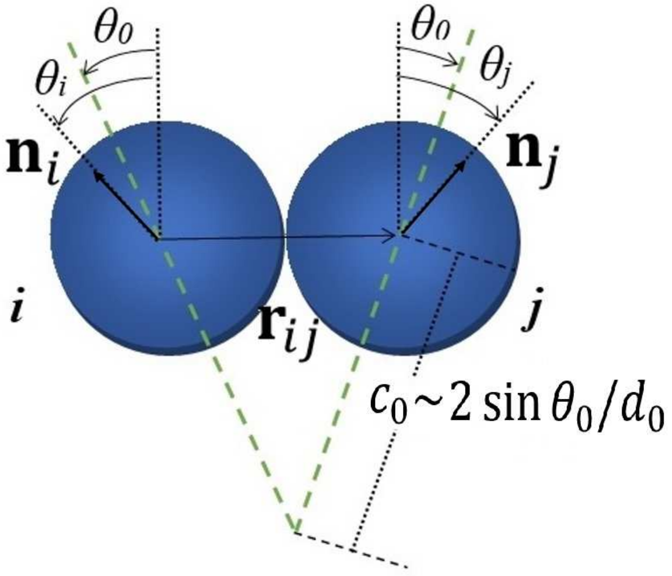


Figure 3.2 Schematics of normal vectors, angular parameters for the particle pair $\{\mathbf{r}_i, \mathbf{r}_j\}$ and relationship between angle parameter θ_0 and spontaneous curvature c_0 where d_0 is the average interparticle distance.

Denoting the position of i^{th} CG particle as $\mathbf{r}_i$, for each pair of particles $\{\mathbf{r}_i, \mathbf{r}_j\}$, we only consider the repulsive potential $u_R(r)$ and attractive potential $u_A(r)$ which are given by the following formulas

$$\begin{aligned} u_R(r) =& \epsilon\left[\big(\frac{r_{min}}{r}\big)^4 - 2\big(\frac{r_{min}}{r}\big)^2\right] \\ u_A(r) =& -\epsilon\cos^{2\zeta}\left(\frac{\pi}{2}\frac{(r-r_{min})}{(r_c-r_{min})}\right), \end{aligned} \tag{3.1}$$

with $r = |\mathbf{r}_{ij}| \equiv |\mathbf{r}_i - \mathbf{r}_j|$. The exponent ζ controls the slope of the attractive branch and r_c is the cutoff radius. r_{min} is the distance which minimizes the potential energy $u_A(r)$ and $r_{min} = \sqrt[6]{2}\sigma$, where σ is the length unit. ϵ is the energy unit which we set $k_BT = 0.23\epsilon$ for numerical simulations in Section 3.4.

We then define an angular function $\phi(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j)$ which depends on the relative orientation between particle pair $\mathbf{r}_i$ and $\mathbf{r}_j$:

$$\phi(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j) = 1 + \mu(a(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j) - 1) \tag{3.2}$$

$$a(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j) = (\mathbf{n}_i \times \hat{\mathbf{r}}_{ij}) \cdot (\mathbf{n}_j \times \hat{\mathbf{r}}_{ij}) + \sin\theta_0(\mathbf{n}_i - \mathbf{n}_j) \cdot \hat{\mathbf{r}}_{ij} - \sin^2\theta_0. \tag{3.3}$$

where $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r$, μ is the parameter related to bending rigidity and θ_0 is the parameter related to the spontaneous curvature. The pair-interaction potential U of each pair of particles $\{\mathbf{r}_i, \mathbf{r}_j\}$ is expressed in terms of the angular function ϕ, $u_R(r)$, and $u_A(r)$ as

$$U(\mathbf{r}_{ij}, \mathbf{n}_i, \mathbf{n}_j) = \begin{cases} u_R(r) + [1 - \phi(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j)]\epsilon, & r < r_{min} \\ u_A(r)\phi(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j), & r_{min} < r < r_c. \end{cases} \tag{3.4}$$

Figure 3.3(a) shows the variation of $a(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j)$ with $\{\theta_i, \theta_j\}$, and Figure 3.3(b) shows the dependence of the attractive component of the potential $U(\mathbf{r}_{ij}, \mathbf{n}_i, \mathbf{n}_j)$ on parameter ζ. From the formulas above, the simplest case is when the normal vectors $\{\mathbf{n}_i, \mathbf{n}_j\}$ are parallel, which gives $a = 1$ and $\phi = 1$.

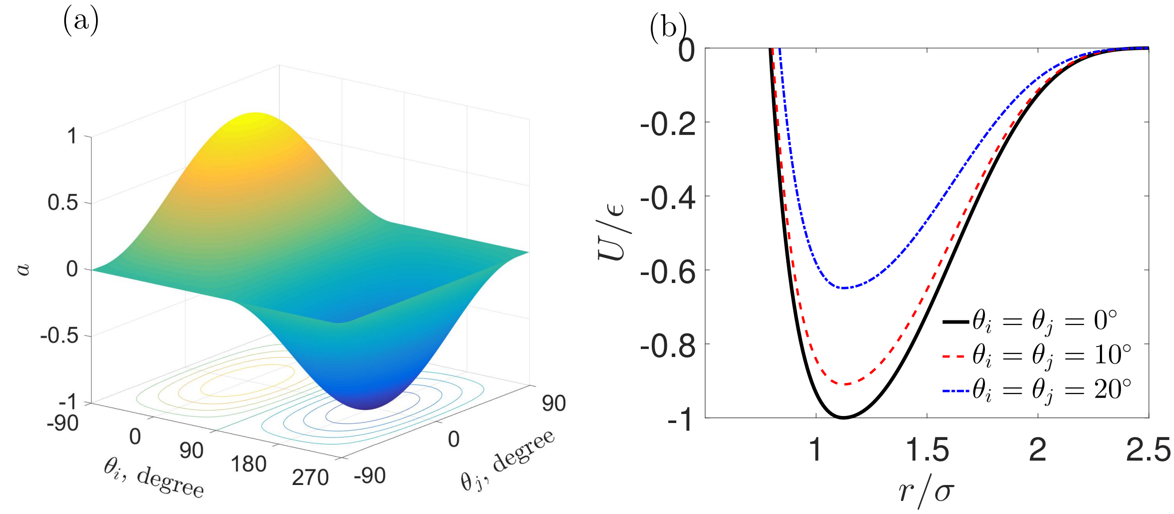


Figure 3.3 (a) Surface plot for the values of $a(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j)$ versus corresponding $\{\theta_i, \theta_j\}$; (b) Dependence of potential energy on angular parameter θ_0 when the model parameter $\zeta = 4.0$.

3.1.3 Cytoskeleton

The cytoplasmic membrane of a RBC is coupled to a cytoskeleton network. Within the coarse-grained formulation, the cytoskeleton network is modeled as a polymeric network that contains three basic types of coarse-grained particles: (1) Junction complexes (actin protofilament and protein band 4.1) are located at the end of spectrin tetramers, (2) Spectrin tetramers (composed of consecutive bonded beads), and (3) Ankyrin proteins located in middle of spectrin beads which connect network to transmembrane proteins. Since we focus on the RBC shape transition at the small deformation regime, we use harmonic springs to model the connectivity between the coarse-grained lipid bilayer membrane and the cytoskeleton network through binding with spectrin, ankyrin and other linking proteins. The harmonic bond potential is given by:

$$E_{harmonic} = K(r_{ij} - r_0)^2, \tag{3.5}$$

where K is a constant and r_0 is the equilibrium distance of each bond. The number of coarse-grained cytoskeleton particles depends on the average distance between each pair of ankyrins. As the cytoskeleton network is enclosed by the lipid bilayer membrane, strong thermal fluctuations may cause nonphysical phenomenon in the CGMD simulations. For example, the coarse-grained cytoskeleton particles may move to the cell exterior. In our simulations precautions are made to prevent this from happening. In the initial configuration of the CGMD simulations of RBC, we put no water molecule between the membrane and cytoskeleton network. Figure 3.4 shows the initial configuration of a coarse-grained RBC and water molecules. Including internal water molecules and external water molecules, we have total 7 types of particles for simulations in LAMMPS.

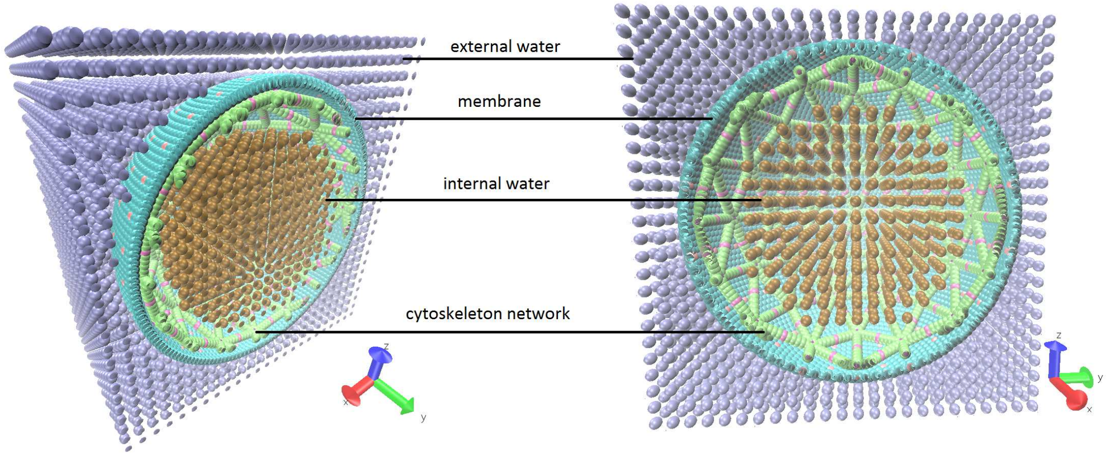


Figure 3.4 Cross-section snapshot of cubical simulation box for coarse-grained RBC simulation in LAMMPS with periodic boundary condition; 7 types of particles are shown in this image: lipid membrane (blue), transmembrane protein (pink), junction complexes (dark purple), spectrin tetramers (lime), ankyrin (light purple), internal water (brown) and external water (gray).

3.1.4 Langevin Dynamics

The dynamics of a coarse-grained macromolecule has been modeled by the Langevin equation. In the case of a coarse-grained lipid bilayer membrane, the Langevin equation with constant friction coefficient is often adopted:

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = -\zeta \frac{d\mathbf{r}_i}{dt} + \mathbf{F}_i + \sqrt{2k_BT\zeta}\mathbf{W}_i, \tag{3.6}$$

where m_i is the mass of coarse-grained particle i, ζ is the friction coefficient and $\mathbf{F}_i$ is the interparticle force. The coefficient of Wiener process $\mathbf{W}_i$ connects the thermal fluctuations of the particles through hydrodynamic interactions. Thermal fluctuation effect is significant in this length scale following from fluctuation-dissipation theorem therefore the axisymmetric case is not considered in this work. In the absence of external driving forces, the covariance between the bead displacements satisfies the following relation

$$\langle \mathbf{W}_i(t) \rangle = 0, \quad \langle \mathbf{W}_i(t) \mathbf{W}_j(t') \rangle = 2k_B T \zeta \delta_{ij} \delta(t - t'). \tag{3.7}$$

Therefore, the magnitude of thermal fluctuation can be controlled by fixing the system temperature.

3.1.5 Nondimensionalization

For LAMMPS simulations, we nondimensionalize length, time and temperature units by following the scaling laws specified in the LAMMPS `lj` unit style:

$$\begin{aligned} l_s =& \sigma, \quad \text{i.e.,} \quad x^* = \frac{x}{\sigma} \\ t_s =& \tau, \quad \text{i.e.,} \quad t^* = t\sqrt{\frac{\epsilon}{m\sigma^2}} \\ T^* =& T\frac{k_B}{\epsilon}, \end{aligned}$$

where m is the mass of each CG membrane particle and T is the temperature.

3.2 Hydrodynamic and Interparticle Interactions

The Lennard-Jones (L-J) potential is used through all this work for the interactions between coarse-grained lipid particles, coarse-grained cytoskeleton and water molecules. As water molecules may pass through the porous structures in the cytoskeleton, we need to take the membrane-water interactions into account. In addition, we also add L-J potentials for cytoskeleton network and membrane to prevent proteins from coming out of the RBC. The L-J potential is given by the following formula

$$E_{LJ} = 4\epsilon\left[\left(\frac{\sigma_{eq}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{eq}}{r_{ij}}\right)^{6}\right], \tag{3.8}$$

where r_{ij} is: (a) distance between two coarse-grained water molecules; (b) distance of each pair of coarse-grained cytoskeleton particles; (c) distance between membrane or cytoskeleton and water molecules. σ_{eq} is the equilibrium length to the interactions.

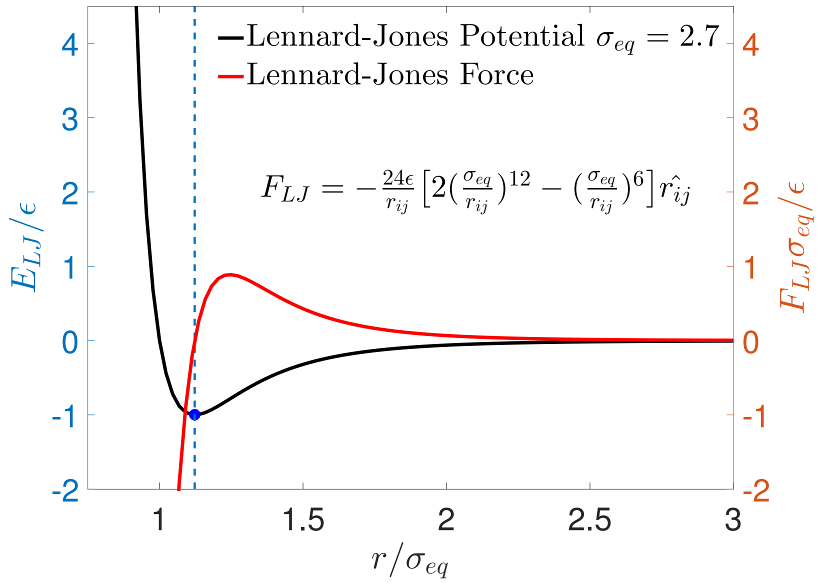


Figure 3.5 Lennard-Jones potential and force curves with $\sigma_{eq} = 2.7$ and the energy constant $\epsilon = 1.0$. The blue dot represents the case when the minimum energy occurs at $r_{min}/\sigma_{eq} = 2^{1/6} \approx 1.12$.

Figure 3.5 shows one example when $\sigma_{eq} = 2.7, \epsilon = 1.0$ case. For water-water interactions, a smaller value of σ_{eq} implies a smaller effective volume occupied by the coarse-grained water molecules. Instead of changing the number of coarse-grained water molecules to adjust the interior RBC/vesicle volume, we adjust σ_{eq} for the coarse-grained internal water molecules to control the volume in RBC. This approach is advantageous as gives the desired interior volume in LAMMPS simulations.

3.2.1 Modified Harmonic Bond Function

The water volume enclosed in the lipid bilayer membrane is controlled by adjusting the effective radius of the coarse-grained water molecules. For the case of a RBC, it is important to ensure that RBC cytoskeleton is stress-free in the initial configuration so we can conduct comparison with previous results in the literature. One way to ensure a stress-free RBC cytoskeleton is by modifying the equilibrium bond length in Equation (3.5) so that initially the harmonic bond energy is zero for the cytoskeleton.

Thus, we modified the harmonic bond in LAMMPS by calling initial configuration x_0 and calculate the bond length $l_0(\mathbf{r}_i, \mathbf{r}_j)$ between each pair of particles at the beginning of simulation. Different from the harmonic bond function in LAMMPS where the bond length r_0 is a constant in Equation (3.5), the modified harmonic bond energy is now

$$\tilde{E}_{harmonic} = K(r_{ij} - l_0(\mathbf{r}_i, \mathbf{r}_j))^2, \tag{3.9}$$

where $l_0 = |x_0, ij|$ is the initial length of the bond. This slight modification helps us achieve a stress-free configuration for the cytoskeleton before we reduce the RBC volume. The detailed LAMMPS implementation of this proposed modified harmonic bond is stated in Appendix A.

3.2.2 Choices of Thermostat Algorithms

Quite a few thermostat algorithms are available in LAMMPS to provide the desired system temperature, such as Langevin thermostat [32], Berendsen thermostat [7] and Nosé–Hoover thermostat [37]. In particular, Nosé–Hoover thermostat is one of the most accurate integration methods in molecular dynamics simulations, and LAMMPS users often use Nosé–Hoover thermostat with NVT (constant number of particles, volume and temperature), NVE (constant number of particles, volume and total energy) or NPT (constant number of particles, system pressure and temperature) ensembles to run the simulations of biological system. For running the equilibrium state of vesicle and RBC simulations, we combined the NVT and NPT ensembles to control the system. We observed that using NPT ensemble on water molecules and cytoskeleton network and NVT ensemble for coarse-grained membrane is able to acquire numerical stable configurations.

3.2.3 Volume Control of Water Molecule Inside the Cell

In our CGMD simulations of lipid bilayer membrane dynamics, we have assumed that no internal fluid molecules will be able to move across the membrane or cytoskeleton and vise versa for the solvent. By adjusting the equilibrium cut length of L-J potential for internal water-water interactions, a smaller σ_{eq} implies less neighbors of water molecules to be taken into account, and this means that the water volume inside the cell is reduced. For instance, our goal may be to reduce the vesicle volume during the simulation within a given amount of time. Using the ramp function with respect to σ_{eq}, we can achieve nearly constant rate of change in vesicle volume. Figure 3.6 is an example, where the green line shows the linear decrease of σ_{eq} from 2.7 at $t \sim 0.8\mu$s (when the equilibrium configuration is reached) to $\sigma_{eq} \sim 2.1$ at $t \sim 8\mu$s. The corresponding vesicle volume scaled to the initial vesicle volume v_0 is depicted in the blue curve. We observe that the wiggles of blue curve is due to the thermal fluctuation effect and the trend of decreasing volume is near linear.

3.2.4 Simulating Procedure in LAMMPS

Figure 3.7 is the general procedure in which LAMMPS users should follow solid arrows for performing vesicle simulations and both dashed and solid arrows for RBC simulations:

1. For complicated geometries such as RBC which requires an elliptic region for membrane and a hexagonal network for cytoskeleton network, one can use

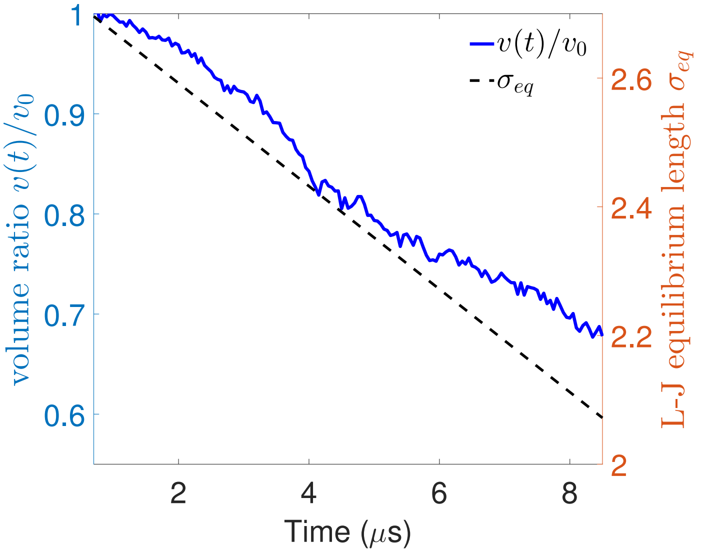


Figure 3.6 Dashed line represents the linear decreasing L-J equilibrium length σ_{eq} over time and the solid curve is the volume fraction $v(t)/v_0$ over time. The occurrence of wiggles is due to the effect of thermal fluctuation presented in the simulation. Clearly, the volume of cell is reduced while σ_{eq} is decreasing.

MATLAB script to generate the initial configurations. Otherwise, LAMMPS can handle spherical region or rectangular patch with the command `region`.

2. Prepare LAMMPS input script to initialize the particles mass, velocities, time step size and the simulation run length.

3. Follows from the requirement of LAMMPS data format as shown in Table A.1 in Appendix A, one needs to call "hybrid" style for using multiple atom styles in LAMMPS. Depends on the subject of simulations, one may need to include the bond styles for specific bonds, for instance, RBC simulations.

4. Use proposed pair-potential as the main pair style for bilayer membrane; use L-J potential for membrane-water, cytoskeleton-water, water-water and membrane-cytoskeleton interactions.

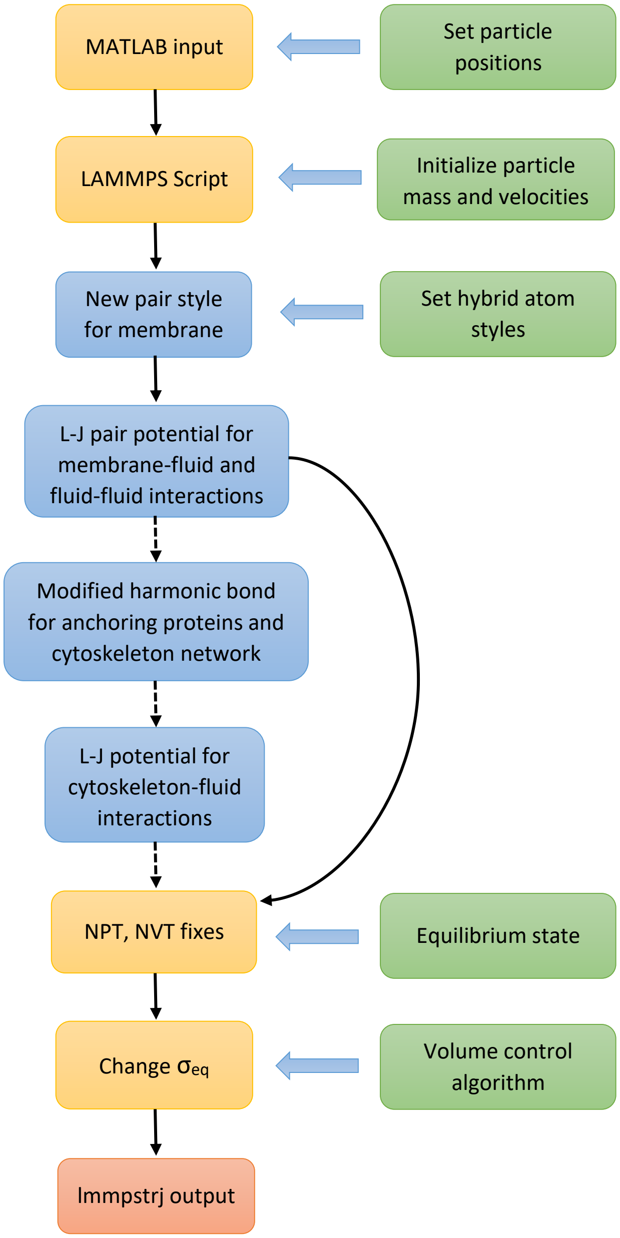


Figure 3.7 Flow chart of simulation procedures in LAMMPS. For simple membrane structure such as planar membrane or vesicle, the procedure follows the solid arrows; for RBC simulations all steps connected by dashed arrows are included.

5. Setup modified harmonic bond energy for linking proteins between membrane and network. Similarly, setup modified harmonic bond energy for spectrin tetramer and ankyrin bonds.
6. For the steps of running equilibrium state, we use the mix of NPT and NVT fixes as our thermostats.
7. Adjust the value of σ_{eq} in L-J potential for internal water-water interactions to achieve the cell volume reduction.
8. Finally, generate LAMMPS output data into the file format ".lammpstrj".

3.3 Membrane Properties

We first show that the proposed pair potential for coarse-grained lipid molecules reproduces some of the basic membrane properties such as in-plane diffusivity, bending rigidity and membrane tension. For the diffusivity, we study a planar membrane patch where the side $L \sim 40\sigma$ and the particle number N is 5822. The time step size is $\Delta t = 0.01\tau$ and we adopt the NVE ensemble with Berendsen pressure control algorithm for 3×10^6 steps. Since the system may take some time to equilibrate, we follow the protocols in the literature [60, 61] to average over the last 1.5×10^6 steps for considerations. The diffusivity can be observed by tracking the mean-square-displacement (MSD) which is given by the following formula:

$$MSD(t) = \frac{1}{N} \sum_{i=1}^{N} \langle (\mathbf{r}_i(t) - \mathbf{r}_i(0))^2 \rangle, \tag{3.10}$$

where MSD is an accumulated value over the time period t. From Einstein's equation for 2D membrane, we can compute the 2D diffusivity by

$$D(t) = \frac{MSD(t)}{4t}. \tag{3.11}$$

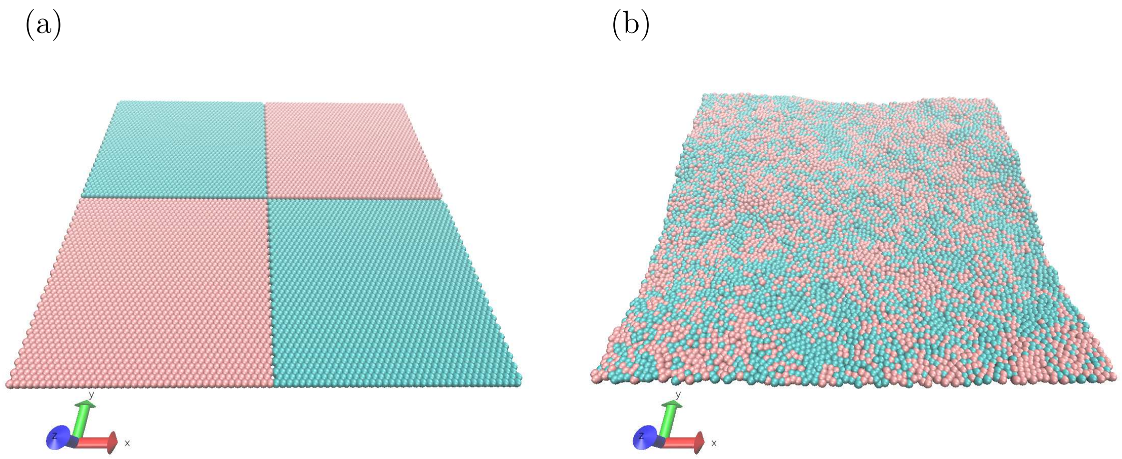


Figure 3.8 (a) Initial state of a planar membrane patch where we separate the membrane with two different colors. (b) Membrane configuration at $t \sim 1$ ms (10^6 time steps). We observe that membrane particles can travel through the membrane behaving as a fluid structure.

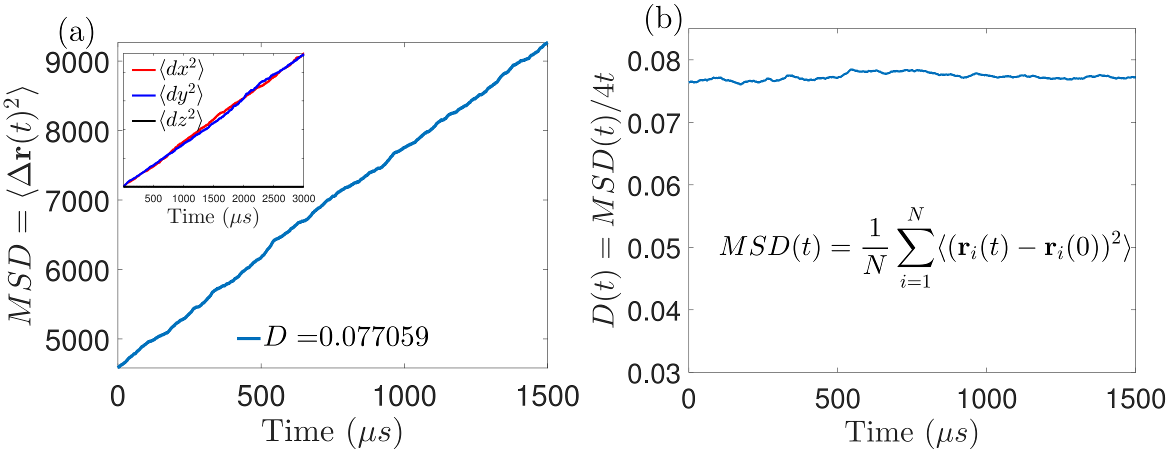


Figure 3.9 (a) Mean-square-displacement (MSD) versus time: linear MSD represents the diffusion property of fluid membrane; (b) From the relationship between the diffusivity and the MSD, we obtain the plot of diffusivity versus time. For this simulation, the parameters are: $\zeta = 4, \mu = 3, \sin\theta_0 = 0, T = 0.23$ and $N = 5822$.

Figure 3.8(a) shows initial state of the membrane patch, and the membrane shape at $t \sim 1$ ms in (b). It is clear that coarse-grained particles move randomly within the membrane as the lipids are in fluid phase. In LAMMPS, mean-square-displacement (MSD) can be computed and stored by adding specific commands. Figure 3.9 shows MSD and diffusivity as we collect the data every 200 time steps. Figure 3.9(a) gives the linearly increasing MSD which represents the pure diffusivity. From the use of the Equation (3.11), we track the diffusivity over time and the result is shown in Figure 3.9(b) which shows the constant diffusion of membrane.

Next we extract the bending rigidity from the membrane fluctuation spectrum. Given a profile function $h(x, y)$ of the planar membrane patch, its Fourier transform

$$\tilde{h}(q) = \frac{1}{L} \sum_{n} h(x, y) e^{i\vec{q}\cdot(x,y)}, \tag{3.12}$$

where the wave vector $\vec{q} = \frac{2\pi}{L}(n_x, n_y)$ and the wavenumber q is the norm of wave vector $\vec{q}$. Following [10], the bending rigidity and the membrane tension for a coarse-grained membrane patch in a 3D periodic domain can be approximated by the following formula

$$\langle |\tilde{h}(q)|^2 \rangle = \frac{k_B T}{L^2(\gamma q^2 + \kappa q^4)}, \tag{3.13}$$

where γ is the membrane tension and κ is the bending rigidity.

For this numerical study, we generate a large membrane patch in a periodic domain with size $L \sim 140\sigma$. The particle number $N = 23452$ and the time step size $\Delta t = 0.01\tau$. The total steps for simulation is 22×10^6 and we dump the trajectories of membrane into a .xyz file every 1000 time steps. After running 12000τ for equilibrium state, we calculate the values of fluctuation spectra by using 2D Fast Fourier Transform in MATLAB which gives fast evaluations to 2D discrete Fourier transform of height function $h(x, y)$.

Figure 3.10 shows that the fluctuation spectra of our simulation results lay on both q^{-2} and q^{-4} lines. From the fitting to Equation (3.13), we obtain the approximations of bending rigidity $\kappa \approx 18\ k_BT$. This result matches the experimental data of bending rigidity for vesicles and we observe from case studies that the bending rigidity is independent of the size of membrane domain. Therefore, in the following section, we perform the numerical applications by using the same parameter set as one used here.

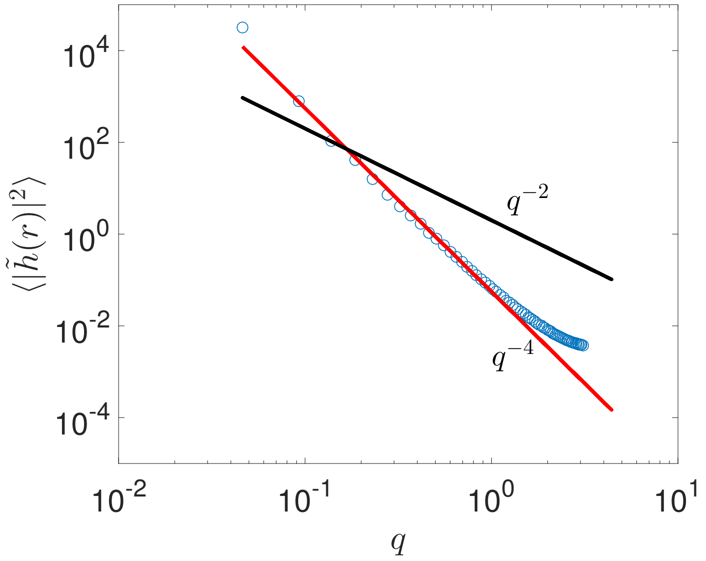


Figure 3.10 Blue circles are our simulation results for the fluctuation spectra of membrane height versus the quantities q where q is the norm of two dimensional wave vector. The parameters for this simulation are: $\zeta = 4, \mu = 3, \sin\theta_0 = 0, T = 0.23$ and $N = 23452$.

3.4 Applications to Biological Systems

We conduct two sets of simulations – vesicle shape transitions and resting shapes of RBCs. Visual Molecular Dynamics (VMD) 1.9.1 is used to generate snapshots from the simulation data.

3.4.1 Vesicle Shape Transitions

Yuan *et al.* used the pair potential in their CGMD simulations of a vesicle and demonstrate the shape transitions of vesicles in [101]. Without using water molecules in their CGMD vesicle simulations, they calculated the volume by local triangulation. The vesicle volume is controlled by a penalty body force from an energy E_V that drives the system to a desirable equilibrium volume V from an initial volume V_0:

$$E_V = \frac{1}{2} K_V \left(\frac{V}{V_0} - 1 \right)^2 \tag{3.14}$$

where K_V is a constant. In principle, we can consider the Equation (3.14) as a spring energy exerted on the vesicle volume. Therefore, the energy E_V should be incorporated into the total free energy. Once E_V is incorporated, enormous amount of computations is needed for local triangulation to the enclosed vesicle membrane at each time step as the volume transitions toward the desirable value [101]. This makes the computation very slow and inefficient.

Rather than local triangulation for volume calculation, we apply coarse-grained particles to explicitly model water molecules with an effective L-J potential for water-water interactions, and adjust the equilibrium radius σ_{eq} to achieve the volume control. With no penalty energy in the total free energy, at each time step we record the membrane configuration and we are able to compute the vesicle volume by using volume approximation of convex hull in MATLAB.

In the following, we report that we can reproduce some of theoretical results using the same set of algorithms in LAMMPS. For vesicle simulations, we only consider a layer of coarse-grained membrane interacting with solvent and calculate the inter-particle force using the proposed pair-potential. We use a spherical vesicle as the initial configuration for of the coarse-grained membrane. The initial vesicle configuration is first generated on MATLAB and a LAMMPS input file is prepared.

We note that the file format for hybrid atom styles should comply with the LAMMPS requirement. For example, volume parameter is essential for peridynamics atom style and the volume of each particle is set to be a fixed constant in this simulation .

The parameter set for the pair-potential is the same as in previous work [101]: angular parameter $\zeta = 4$, the parameter $\mu = 3$ is from orientation dependent function $\phi(\hat{\mathbf{r}}_{ij}, \mathbf{n}_i, \mathbf{n}_j)$ in Equation 3.2 and θ_0 can be calculated from desired spontaneoues curvature c_0. For the energy unit ϵ to the system we set $k_B T = 0.23\epsilon$ and the cutoff length in pair-potential is $r_c = 2.6$. The thickness of membrane $\sigma_{ves} \approx 5$ nm and the diameter of vesicle is $50\sigma_{ves}$. We run simulations in periodic box where the side is $70\sigma_{ves}$. As mentioned in Section 3.2.2, we apply NVT ensemble for water interactions and NPT ensemble for the bilayer membrane. We then perform the simulation for vesicle shape transitions by using the ramp function to adjust σ_{eq} in the L-J potential for water-water interactions. Moreover, the equilibrium length of Lennard-Jones potential to the water-water interactions is initially set with $\sigma_{eq} = 2.7$. As described in Section 3.2, we reduce vesicle volume by decreasing the value of σ_{eq}. From single run simulation with wide range of change of σ_{eq}, say, from 2.7 to 1.5, we are able to track that specific shapes occur when σ_{eq} is at corresponding level. After obtaining the desired vesicle shapes, we unfix the ramp function and relax each case for several time units to obtain the equilibrium states. Finally, we recorded the numerical experiments and corresponding σ_{eq} values. We observed that within certain range of σ_{eq} the vesicle will remain at the specific shapes as shown in Figure 3.11.

Our simulation data show that one can obtain prolate shape, dumbbell shape, biconcave shape, stomatocyte-like shape and inward budding cell when $\sigma_{eq} \approx \{2.5, 2.3, 2.0, 1.9, 1.7\}$, respectively. Notice that according to [101], this shape transition can be achieved by using fast rate of volume change $\dot{v}$ within 200τ where τ is the dimensionless time unit and in real time τ is of order 0.1 μs. The dimensionless time step size $\Delta t = 0.01$ is used for this simulation and total number of time steps is vary for cases of desired shapes. Figure 3.11 shows that starting with spherical vesicle, when the spontaneous curvature $\sin\theta_0$ is set to be 0, we have the shape transitions which are prolate ($v = 0.8$), dumbbell ($v = 0.7$), biconcave ($v = 0.6$), stomatocyte ($v = 0.65$) and inside budding shapes ($v = 0.45$) with corresponding volume of internal water.

With similar setup, we then perform the simulations for the cases when the spontaneous curvature c_0 is nonzero and the results are shown in Figure 3.12. We observed that different from the case when $c_0 = 0$, for $c_0 = 2$ and $c_0 = 4$, the vesicle forms tube like configuration and outward budding shapes.

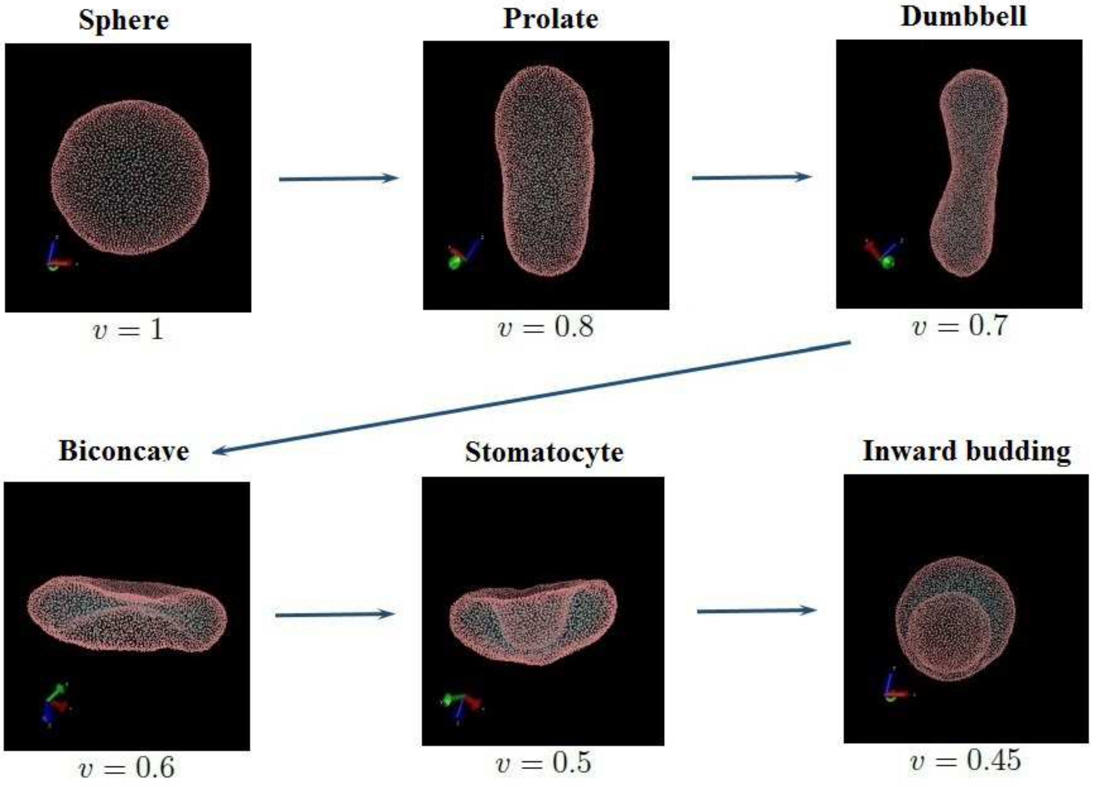
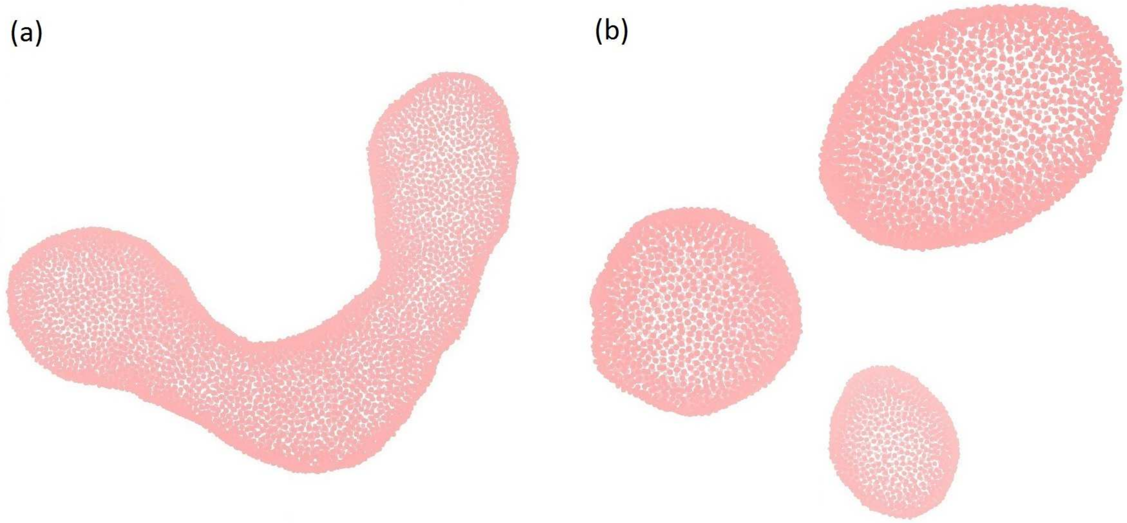


Figure 3.11 Shape transitions of vesicle for corresponding values of vesicle volume v when the spontaneous curvature $c_0 = 0$. According to [101], this case of transition occurs when the volume change rate is high ($\dot{v} = 1.75 \times 10^{-3}\tau^{-1}$).



Tube **Outward budding**

Figure 3.12 Possible shapes of vesicles when curvature c_0 is nonzero: (a) Tube like configuration ($c_0 = 2$) and (b) outward budding shape ($c_0 = 4$) when $v = 0.65, 0.45$, respectively.

3.4.2 Resting Shapes of RBCs

Recall the modified harmonic bond function in Section 3.2.1, the stress-free configuration of cytoskeleton network plays an important role in the study of resting shapes of RBCs, by minimizing the total elastic energy of the system. Lim *et al.* compared their simulation results with experiments for resting shapes of RBCs for cases of different spontaneous curvatures and access areas between inner and outer leaflets of the lipid bilayer [54].

In this study, we consider slightly eccentric spheroid as the stress-free configuration of a RBC. Denote the volume of each initial cell as V_s and the volume of original stress-free spheroid cell as V_0, we performed simulations for the cases when the ratio V_s/V_0 are $\{0.9, 0.925, 0.95, 0.975, 0.995, 0.998\}$. To validate our simulations against the continuum simulation results shown in [67], we reduced the volume of RBC from V_0 to $0.65V_0$ and reproduced the results under values of spontaneous curvatures c_0. Figure 3.13 shows the numerical results of stress-free RBC using finite element method (FEM) [67]. With the use of the same parameter set as given in the first part of numerical simulations, one additional NPT fix is needed for cytoskeleton network then we can obtain the stress-free state for both membrane and cytoskeleton.

The experimental observed size of RBC is $6 \sim 8\ \mu m$ in diameter but here we inherit the same cell size from previous section where the diameter of cell is $50\sigma_{rbc}$ and the gap between cytoskeleton and membrane is set to be $2\sigma_{rbc}$. As done in the previous simulations, we used periodic boundary condition to run the simulations. With enough amount of coarse-grained particles on membrane and cytoskeleton, the small size of the cell can also achieve the same deformations as ones occurred in real size RBC. In addition, this setup reduces computational cost. We would like to make a remark here that due to the volume of internal water is not fixed for cases of spheroid, σ_{eq} should be carefully adjusted and the cytoskeleton inside the cell takes space therefore the values of σ_{eq} from previous simulation of vesicles are not feasible.

Our simulation results generated from LAMMPS are shown in Figure 3.13. Comparing our result with the numerical results from continuum model using FEM, it is clear that for V_s/V_0 is at $\{0.9, 0.925, 0.95\}$ we have close resting shapes which are biconcave shapes. Also, for V_s/V_0 is at $\{0.975, 0.995, 0.998\}$ we have bowl shapes which are nearly identical to the results from continuum modeling. We want to point out that the shape transition from bowl shape to biconcave shape is well predicted in this simulation and we include the zoom-in views of both shapes in Figure 3.14. Lastly, due to the thermostat generated from LAMMPS, thermal fluctuation would be a huge factor to the simulation in LAMMPS which may cause numerical instability. Thus, the time step size for this simulation is smaller than one in previous sections. Here we used $\Delta t = 0.005$.

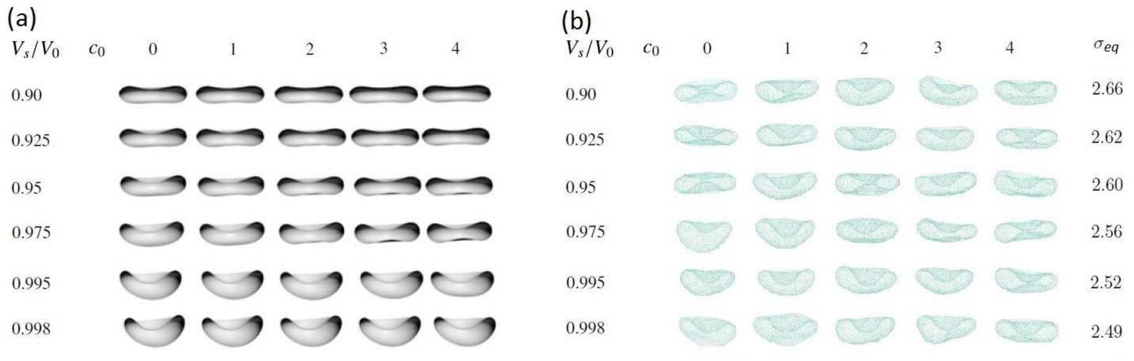


Figure 3.13 (a) FEM simulation results for resting shapes of RBC reprinted from [67]; (b) Our simulation results using LAMMPS for various spontaneous curvature c_0 versus cases of V_s/V_0. We also provide the equilibrium length σ_{eq} for internal fluid interactions in each simulation.

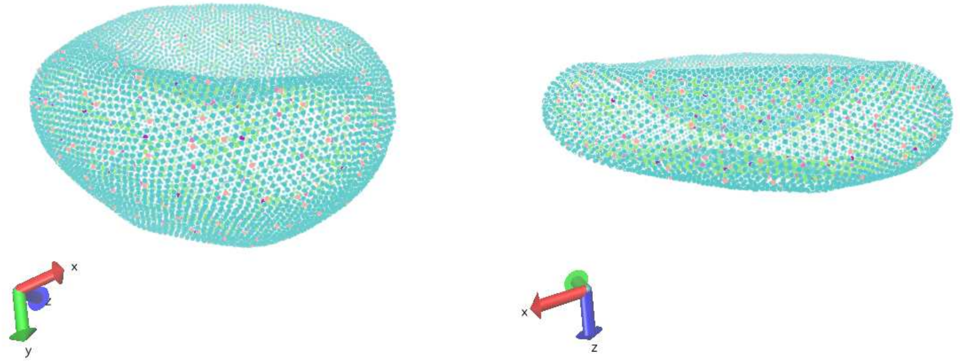

(a) (b)

Figure 3.14 Possible resting shapes of stress-free RBCs: (a) Bowl shape (Stomatocyte); (b) Biconcave shape.

3.5 Conclusion

In this work, we incorporate into LAMMPS the pair-potential developed for one-particle-thick CGMD simulations of lipid bilayer membranes. Using LAMMPS we demonstrated that the dynamics of a lipid bilayer membrane immersed in a viscous fluid can be simulated with explicit solvents. We also provide the instruction for preparing the simulation setups consisting of the coarse-grained vesicles, RBCs, internal fluid and solvent. In order to apply the stress-free configuration of the cytoskeleton of RBCs, we modify the built-in harmonic bond energy to acquire the equilibrium state of cytoskeleton network.

Our simulation results show that the orientation-dependent pair-potential for the coarse-grained lipids well captures the membrane properties such as membrane diffusivity, bending rigidity and membrane tension (by evaluating the mean-square-displacement and fluctuation spectrum of height function). Furthermore we incorporate coarse-grained water molecules to account for the hydrodynamic interaction between a lipid bilayer membrane and the fluid around it. A Lennard-Jones potential is adopted for interactions between solvent molecules, and by adjusting the equilibrium length σ_{eq} we can control the volume enclosed inside the lipid bilayer membrane or RBC. In our CGMD simulations using LAMMPS we are able to reproduce the shape transitions of vesicle for cases of desired equilibrium volumes. We also perceive the contrast between results in continuum modeling and ones in CGMD for the resting shapes of RBC.

LAMMPS is equipped with the capability for parallel computing with OpenMPI. To illustrate how the approaches presented in this paper may be practical for parallel simulation of realistic biological membranes, here we demonstrate some timing results from parallel computation of the CGMD model of the lipid bilayer membrane: (a) for a total number of CG lipid particles $N = 23452$ running parallel computing on a cluster (with 2.53 GHz 6-core processor) with 160 CPUs, it takes about 3 hours to integrate the system to 1 ms; (b) Table 4.2 shows the timing results of GUV simulations with 144 CPUs. In other words, it becomes practical to couple our pair function with more complicated biological system.

Table 3.1 Timing Results (sec) for Running 10^5 Time Steps of GUV Simulations with 144 CPUs

$D_{vesicle}$	N_{total}	$N_{bilayer}$	N_{water}	T_{100k} (sec)	L_{box}
250 nm	20011	8346	11665	179.70	350 nm
500 nm	147750	31404	116346	439.84	700 nm
1000 nm	1152245	125588	1026657	2761.13	1400 nm

CHAPTER 4

ACTION FIELD METHOD FOR MOLECULAR DYNAMICS OF LIPID BILAYER MEMBRANES

In this work we aim to adopt an aggressive coarse-grained modeling where each bead represents a collection of lipid molecules and consider the action due to lipid-lipid interactions. With the known physical properties of lipid–hydrophobic tail and hydrophilic head, we can prescribe a condition where the action field exists when tail-tail interactions effects are significant and is absent from the tail-head interaction. With this setup, the orientations of lipids are implicitly defined and axial rotations can be derived and calculated.

4.1 Action-Field of Lipid-Lipid Interactions

Figure 4.1(a) shows the schematic of membrane fusion process. From panel A to panel F, we can observed that proximal layers form pores when membrane fusion begins and the lipids at contact surfaces change corresponding orientations. Figure 4.1(b) is the Janus particle representation of a lipid bilayer membrane where each Janus particle represents single lipid. In other words, the diameter D of each Janus particle is a half of thickness of membrane which is $D \sim 2.5$ nm. The panel (c) of Figure 4.1 shows that from this setup the orientations of Janus particles are assigned implicitly from lipid heads toward lipid tails.

The usage of Janus particles for forming the lipid bilayer membrane refers to the aggressive coarse-graining. Different from the implementation in [75] for the field evaluations where they marked points on lipid tails and considered separation angles, we propose to solve the action field solutions by using the IEM which has capability on solving the screened Laplacian in any geometry with Dirichlet boundary conditions. In this work, we consider our Janus particles as spheres (or ellipsoids) where the

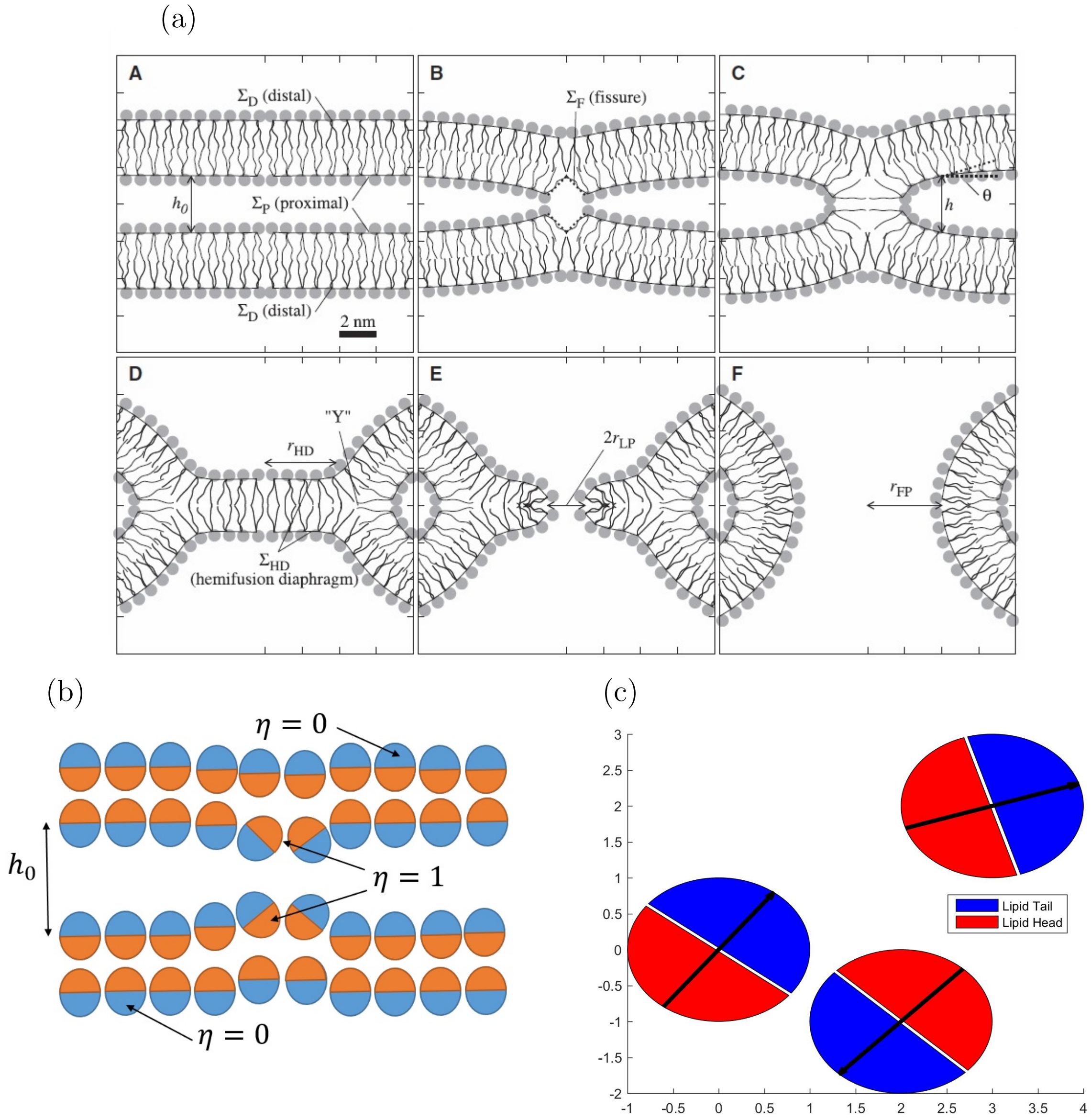


Figure 4.1 (a) Membrane fusion process from two interacting separate membranes in A to two fused membranes in F. This figure is reprinted from [75]; (b) The schematic of Janus particles representing the lipid bilayer membrane where the coarse-grained lipid tail is in orange and lipid head is in blue; (c) The orientations of Janus particles are assigned from lipid heads toward lipid tails.

solution exists only on the particle surfaces. As shown in the panel (b) of Figure 4.1, we propose that the solution of action field $\eta = 1$ on lipid tails and $\eta = 0$ on lipid heads. Moreover, recall from the proposed boundary conditions, we denote the surface of one hemisphere representing the lipid tail as Σ_T and the surface of the other one for lipid head as Σ_H. Hence, the solution on the boundary is given by: $\eta = 1$ on Σ_T and $\eta = 0$ on Σ_H.

From calculus of variation, we can derive the variation of the attraction potential δV in the computational domain and update particle positions and directions in terms of δV. Given initial i^{th} particle center position and direction $\{\mathbf{x}_i, \mathbf{d}_i\}$ and suppose that the action field η solves Equation (1.9), we have the following hydrophobic attraction energy on the membrane surfaces:

$$V = \sigma \int_{\mathbb{R}\setminus\bigcup\limits_i B_i} \rho|\nabla\eta|^2 + \rho^{-1}|\eta|^2 d\mathbf{x}, \tag{4.1}$$

where σ the surface tension and ρ the screened length to the system. Refer to [75], the parameters are given by $\sigma = 11k_BT/nm^2$ and $\rho = 0.22\ nm$.

Before proceeding to the calculation of energy variations, we first nondimensionalize the proposed hydrophobic attraction energy potential by the following scaling laws:

$$\text{length scale: } [L] = D = 2.5 \text{ nm};$$
$$\text{energy scale: } [E] = \sigma \times [L]^2 = 68.75k_BT.$$

This implies that all dimensionless parameters are given by:

$$\begin{aligned}\bar{D} =& 1.0;\\ \bar{\rho} =& 0.088;\\ \bar{\sigma} =& \sigma\frac{[L]^2}{[E]} = 1.0.\end{aligned}$$

For the following derivations we omit the bars from notations and the variation with respect to time perturbation dV/dt at the domain is given by

$$
\begin{aligned}
\frac{dV}{dt} =& \int_{\bigcup_i \partial B_i} \Big(\rho|\nabla\eta|^2 + \rho^{-1}|\eta|^2\Big) \cdot \mathbf{u} \cdot \hat{\mathbf{n}} d\mathbf{x} \\
&+ \int_{\mathbb{R}^3 \setminus \bigcup_i B_i} 2\Big[\rho\nabla\eta \cdot (\delta\nabla\eta) + \rho^{-1}\eta \cdot (\delta\eta)\Big] d\mathbf{x}.
\end{aligned} \tag{4.2}
$$

where $\hat{\mathbf{n}}$ the unit outward normal of the boundary and $\mathbf{u}$ the local velocity field. Use integration by parts on the first term of the second integral above, we then obtain:

$$
\begin{aligned}
\frac{dV}{dt} =& \int_{\bigcup_i \partial B_i} \Big(\rho|\nabla\eta|^2 + \rho^{-1}|\eta|^2\Big) \cdot \mathbf{u} \cdot \hat{\mathbf{n}} + 2\rho(\nabla\eta \cdot \hat{\mathbf{n}})\eta_t dS \\
&+ \int_{\mathbb{R}^3 \setminus \bigcup_i B_i} 2\Big(- \rho\Delta\eta + \rho^{-1}\eta\Big)(\delta\eta) d\mathbf{x}.
\end{aligned} \tag{4.3}
$$

where we can prove that $\eta_t = \nabla\eta \cdot \mathbf{u}$ and the energy variations become:

$$
\begin{aligned}
\frac{dV}{dt} =& \int_{\bigcup_i \partial B_i} \Big(\rho|\nabla\eta|^2 + \rho^{-1}|\eta|^2\Big) \cdot \mathbf{u} \cdot \hat{\mathbf{n}} + 2\rho(\nabla\eta \cdot \hat{\mathbf{n}})(\nabla\eta \cdot \mathbf{u}) dS \\
&+ \int_{\mathbb{R}^3 \setminus \bigcup_i B_i} 2\Big(- \rho\Delta\eta + \rho^{-1}\eta\Big)(\delta\eta) d\mathbf{x}.
\end{aligned} \tag{4.4}
$$

The last term of the equation above vanishes automatically when we minimize the energy. Once the action field η is solved, we can evaluate the approximation of $\nabla\eta$ on the domain boundary. Therefore the energy variation (4.4) can be obtained and the particle configuration and net torque $\{\mathbf{x}_i, \tau_i\}$ can be updated by the following formulas from Euler-Lagrange derivatives:

$$
\begin{aligned}
\xi_x \frac{d\mathbf{x}_i}{dt} =& \int_{\partial B_i} \frac{\delta V}{\delta \mathbf{x}} dS \\
=& \int_{\partial B_i} \Big(\rho|\nabla\eta|^2 + \rho^{-1}|\eta|^2\Big) \cdot \hat{\mathbf{n}} + 2\rho\nabla\eta(\nabla\eta \cdot \hat{\mathbf{n}}) dS, \quad \text{when} \quad \mathbf{u} = \delta\mathbf{x}
\end{aligned} \tag{4.5}
$$

$$
\xi_\theta \frac{d\theta_i}{dt} = \tau_i = \int_{\partial B_i} 2a\rho(\nabla\eta \cdot \hat{\mathbf{n}})(\nabla\eta \cdot \hat{\mathbf{t}}) dS, \quad \text{when} \quad \mathbf{u} = \omega(\mathbf{x} - \mathbf{x}_i) = \omega a \hat{\mathbf{n}}, \tag{4.6}
$$

where a is the radius of Janus particle and ω is a two-dimensional rotation matrix which is given by:

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix},$$

and ξ_x and ξ_θ are drag coefficients of translational and angular velocities. $\hat{\mathbf{t}}$ is the unit tangent vector and θ_i is the corresponding angle of i^{th} particle orientation with respect to the x-axis. The relationship above states that the angular velocity is proportion to the torque with a reciprocal of rotational drag coefficient.

To determine the drag coefficients ξ_x and ξ_θ, we assume that all Janus particles are immersed in the water where the viscosity is given by $\mu = 0.93$ cP when the temperature is $T = 296.15$K. For a spherical particle of radius a, the drag coefficients are known as $\xi_x = 6\pi\mu a$ and $\xi_\theta = 8\pi\mu a^3$. Neglecting the thermal fluctuations for now and using the hydrodynamic drag coefficient for a sphere of radius $a = 1.25$ nm, we scale time by the following characteristic time for the coarse-grained lipid (janus) particle

$$\text{time scale } [T] = \frac{6\pi\mu a}{[E]}[L]^2 \approx 4.88 \times 10^{-10} \text{ second.}$$

Thus the translational drag ξ_x will be scaled to one and the rotational drag will be scaled as

$$\bar{\xi}_\theta = 8\pi\bar{\mu}(\frac{\bar{a}}{2})^3 = \bar{\mu}\pi = \frac{1}{3}.$$

4.2 Integral Equation Method

For solving action field η numerically we adopt the IEM which only utilizes the surface potentials to compute the solutions with boundary data when the governing equation is homogeneous. The simulation procedures are described as the follows:

1) Recall the Screened Laplace equation and write it in the form of PDE in term of a differential operator L:

$$Lu = 0 \quad \text{in } \Omega, \quad u = f \quad \text{on } \Gamma, \tag{4.7}$$

where Ω is the simulating domain and Γ is the boundary of the target geometry.

2) From the potential theory, we can write down the solution in terms of a linear combination of single layer potential S and double layer potential D:

$$u = CS\sigma + D\sigma = C\int_\Gamma G(x,y)\sigma(y)dS_y + \int_\Gamma \frac{\partial G(x,y)}{\partial n}\sigma(y)dS_y, \tag{4.8}$$

where $\sigma(x)$ is an unknown density in the computational domain and C is a constant. The so-called Brakhage-Werner parameter $C = i$ is used through this work [13]. The Green's function is given by the (1.10) and the series solution is formed by modified Bessel function. Therefore, we can express the solution above by QBX method [47]

$$D\sigma(x) = \sum_{l=-\infty}^{\infty} \alpha_l^{QBX} K_l(\rho^{-1}r)e^{il\theta}. \tag{4.9}$$

The coefficients α_l can be computed numerically and is given by:

$$\alpha_l^{QBX} = \frac{i}{4}\int\limits_\Gamma H_l^{(1)}(i\rho^{-1}|x' - c|)e^{il\theta'}\sigma(x')dx' \quad (l = -p, -p+1, \ldots, p), \tag{4.10}$$

where $(|x' - c|, \theta')$ denote the polar coordinates of the point x' with respect to the point c.

3) From one side limits for both exterior and interior problems, a jump relation is given by:

$$f = D^{*}\sigma(x)\Big|_{\Gamma} = \lim_{x^{\pm}\to x} D\sigma(x^{\pm}) \pm \frac{1}{2}\sigma(x), \tag{4.11}$$

where the sign depends on the type of problem. We then need to use iterative solver such as GMRES to solve the density $\sigma(x)$. After obtaining the density $\sigma(x)$ we can use the FMM to solve $u = D\sigma$.

4) The boundary condition for i^{th} Janus particle is given by

$$f(r_i, \theta_i) = \begin{cases} 1 & -\frac{\pi}{2} < \theta_i < \frac{\pi}{2} \\ 0 & \frac{\pi}{2} < \theta_i < \frac{3\pi}{2} \end{cases}. \tag{4.12}$$

where the angle θ_i corresponds to the particle orientation $\mathbf{d}_i$.

In order to have smooth boundary condition, one suitable option is to use hyperbolic functions instead of adopting step functions. For example, one candidate can be:

$$f(r_i = a_i, \theta_i) = \frac{\tanh(\theta_i + \frac{\pi}{4}) - \tanh(\theta_i - \frac{\pi}{4})}{2\tanh(\frac{\pi}{4})}, \tag{4.13}$$

where a_i is the radius of the i^{th} particle.

5) We also need to obtain the action field gradient $\nabla\eta$ when we solve the field η.

Remarks: 1. the accuracy of iterative solver is set to be 10^{-8} for the trial case. 2. η and $\nabla\eta$ may be solved in the same script. Otherwise, it can be computed by using central difference scheme. 3. Post processing includes variation calculation configuration updates and visualization. 4. The base problem solving package can be acquired from

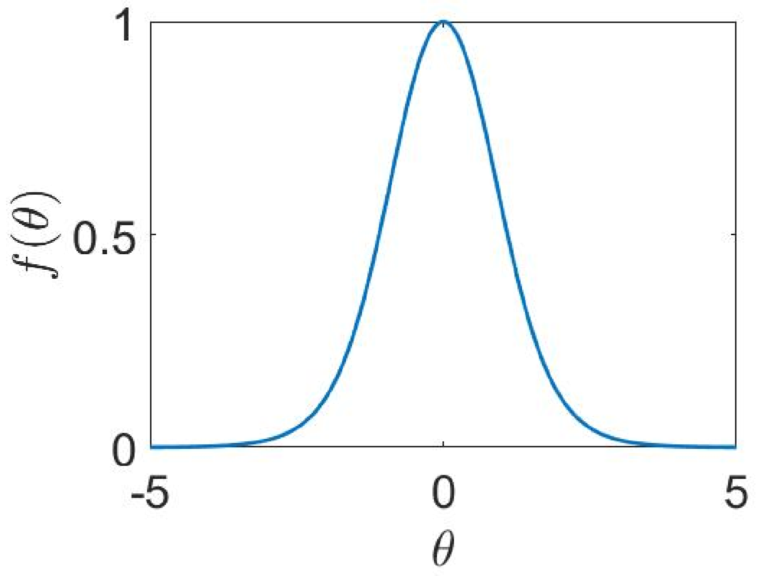


Figure 4.2 Boundary function (4.13).

`http://github.com/inducer/pytential`

developed by Dr. Andreas Klöckner. All codes are written in Python and C++ languages and post-processing script is performed in MATLAB.

The BVP (1.9) is solvable from the use of separation of variables in polar coordinates and the series solution is in terms of the modified Bessel's equation for N-particle system:

$$\begin{aligned}\eta(r,\theta) &= \sum_{j=1}^{N} \eta_j(r,\theta) = \sum_{i=j}^{N} R_j(r)\Theta_j(\theta) \\ &= \sum_{j=1}^{N}\sum_{k=1}^{\infty}\left[\left(C_{j1k} I_{\sqrt{k}}(\frac{r}{\rho}) + C_{j2k} K_{\sqrt{k}}(\frac{r}{\rho})\right)\left(C_{j3k} e^{i\sqrt{k}\theta}\right)\right] \\ &+ \sum_{j=1}^{N}\left[C_{j10} I_0(\frac{r}{\rho}) + C_{j20} K_0(\frac{r}{\rho})\right)\left(A_{j0}\theta + B_{j0}\right)\right]\end{aligned} \tag{4.14}$$

where

$$\begin{aligned} I_\alpha(x) =& \sum_{m=0}^{\infty} \frac{1}{m!\Gamma(m+\alpha+1)}\Big(\frac{x}{2}\Big)^{2m+\alpha} \\ K_\alpha(x) =& \frac{\pi}{2}\frac{I_{-\alpha}(x) - I_\alpha(x)}{sin(\alpha\pi)}. \end{aligned} \tag{4.15}$$

Since the modified Bessel function of the first kind $I_\alpha(x)$ does not vanish at farfield, the solution can be represented as:

$$\eta(r,\theta) = \sum_{n=-\infty}^{\infty} A_n K_n(\frac{r}{\rho})e^{in\theta}, \tag{4.16}$$

where the coefficients A_n can be calculated from the boundary condition $f(a,\theta)$:

$$A_n = \frac{1}{2\pi K_n(\frac{a}{\rho})}\int_{-\pi}^{\pi} fe^{-in\theta}d\theta. \tag{4.17}$$

where the particle radius is a and this solution will be used for performing tests of numerical accuracies.

4.3 Numerical Results

4.3.1 Convergence Test and Timing Results

The quadrature error $O((\frac{h}{4r})^{2q})$ and the truncation error $O(r_{QBX}^{p+1})$ are provided by [47] where q is the number of points evaluated in Gauss-Legendre quadrature, p the order of the QBX, r_{QBX} the radius of a disk centered at an off-surface point and h the arc length of discretized panels. For the following numerical results, $(\rho = 1)$ is used through all simulations and the system dimension is 2. We begin from testing two Janus particles on the x-axis where they face toward each other, i.e., $\{\mathbf{d}_{left}, \mathbf{d}_{right}\} = \{(1,0),(-1,0)\}$. Figure 4.3 shows surfaces plots and x-y projections when the distance between two centers of particles $r = \{2,3,4,5,6,7\}$. The schematic is shown in the Figure 4.4. As expected that the solution of the action field η in middle decays when the two particles separate. As an example, the panel (b) in the Figure 4.4 is the cross-section plot at x-axis when the distance between two particles is $r = 7$.

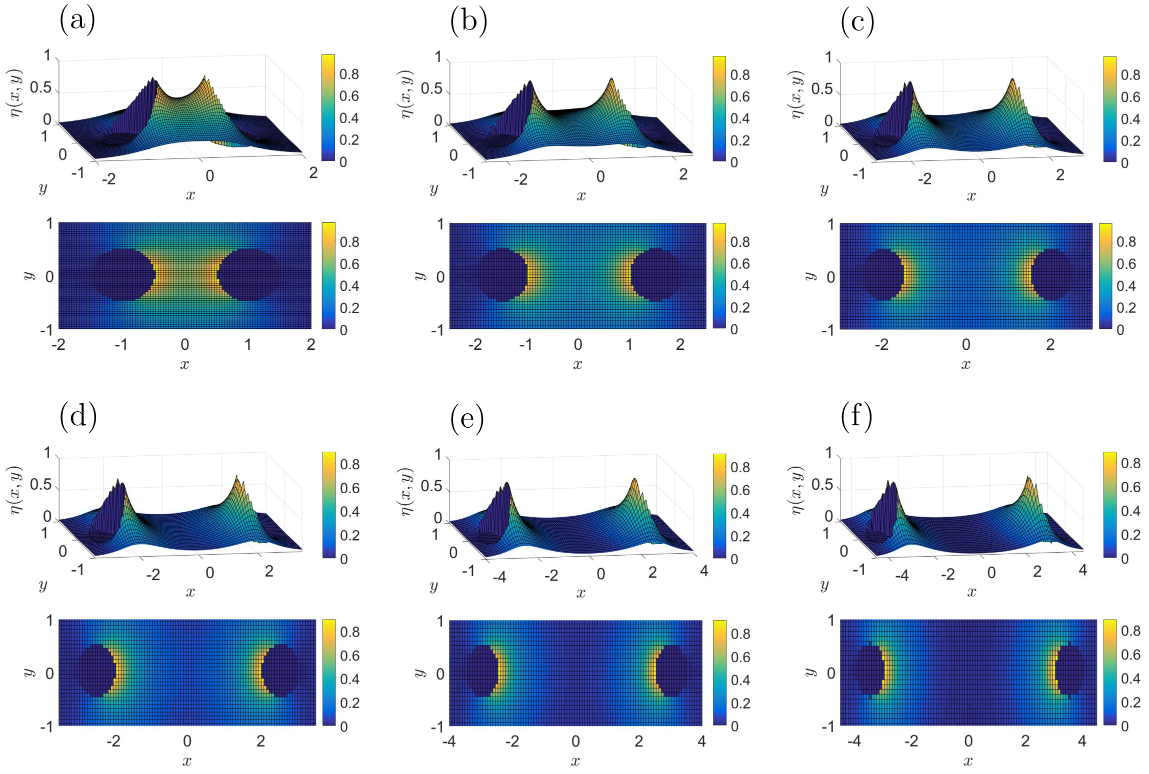


Figure 4.3 (a)–(f) Action field profile for the distances of two Janus particles on the same axis $r = \{2, 3, 4, 5, 6, 7\}$.

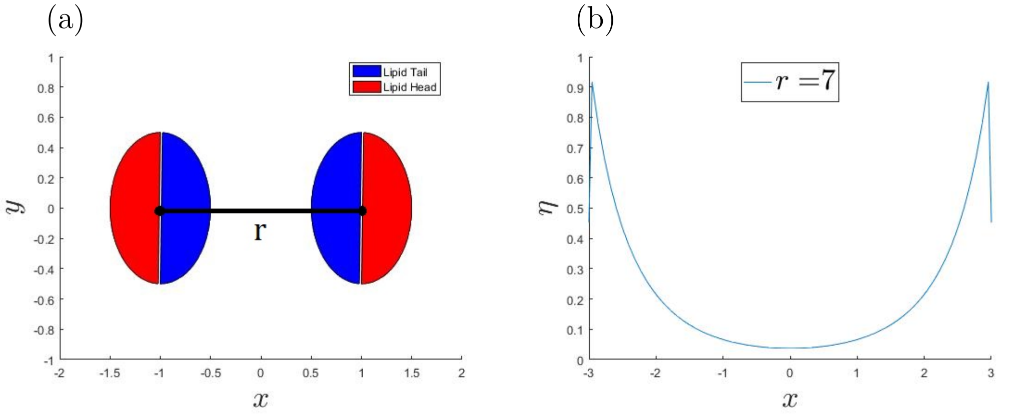


Figure 4.4 (a) Schematic of two Janus particles on the same axis facing each other; (b) Cross-section plot when $r = 7$.

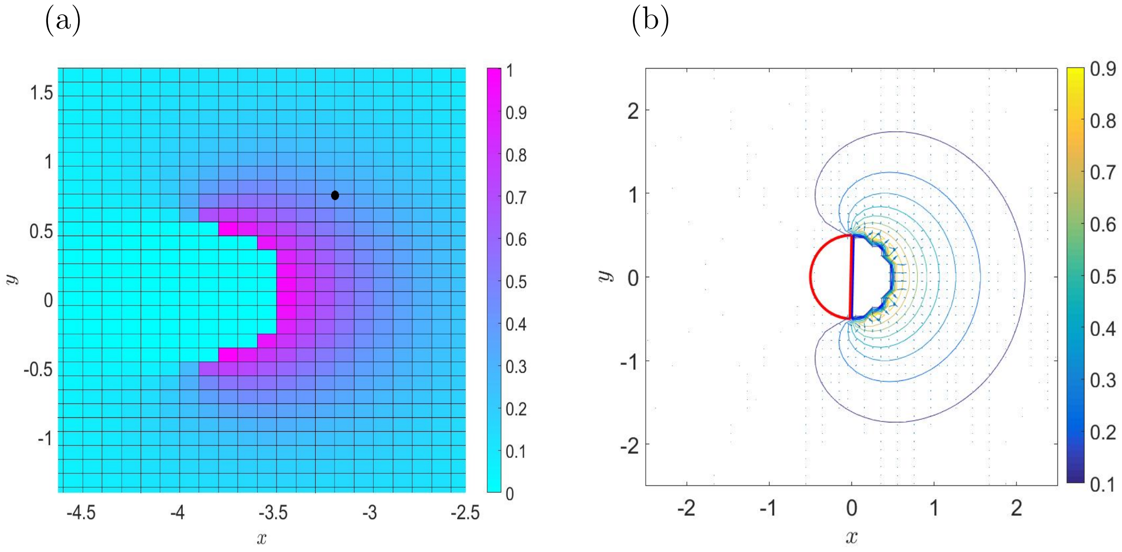


Figure 4.5 (a) Schematic of checking solution accuracy; (b) Contour plot of the action field.

To perform the convergence tests, we test three cases: (1) The number of panel segments is equal to 4 with various numbers of panels on the boundary; (2) The number of panel segments is equal to 8 with various numbers of panels on the boundary; (3) with the fixed number of panel segments and the number of panels on the boundary, we adjust the tolerance of iterative solver. The number of evaluated points on the boundary is given by the following formula: $N_{bdary} = N_{panel} \times (N_{segment} + 1)$. It is clear that with higher number of panels and the number of panel segments, the grid points marked on the boundary increase. As shown in the Figure 4.5, we pick a point near one Janus particle to check the accuracy of solution where the solution can be obtained from Equation (4.16). The results of Table 4.1–4.4 show that with higher number of panel segments, the numerical error can be reduced.

We then show the numerical accuracy of the solutions in Figure 4.6 where we adjust the QBX order p and the QBX radius $r_{QBX} = h/2$ is used for this test. It is clear that the ℓ_2 error of numerical solution is of the order h^{p+1}. Table 4.5 shows that increasing the number of grid does not effect the numerical accuracy of the

Table 4.1 Convergence Test 1: The number of panel segments $N_{segment} = 4$ and the Tolerance $tol_{GMRES} = 1.0 \times 10^{-13}$ are Fixed; Total Grid Number is 100×100

N_{panel}	N_{bdary}	Iterations	l_2 error
20	100	10	1.30×10^{-3}
40	200	10	6.36×10^{-5}
80	400	10	2.56×10^{-6}

Table 4.2 Convergence Test 2: The number of panel segments $N_{segment} = 8$ and the Tolerance $tol_{GMRES} = 1.0 \times 10^{-13}$ are Fixed; Total Grid Number is 100×100

N_{panel}	N_{bdary}	Iterations	l_2 error
20	180	10	8.46×10^{-7}
40	360	10	1.79×10^{-8}
80	720	10	1.92×10^{-9}

Table 4.3 Convergence Test 3: The number of panel segments $N_{segment} = 12$ and the Tolerance $tol_{GMRES} = 1.0 \times 10^{-13}$ are Fixed; Total Grid Number is 100×100

N_{panel}	N_{bdary}	Iterations	l_2 error
20	260	10	4.25×10^{-9}
40	520	10	5.85×10^{-10}
80	1040	10	6.85×10^{-11}

solution. The reason is that the nodes on the boundary are fixed and the numerical error depends on the parameter N_{bdary} as shown in Figure 4.7. Figure 4.8 is generated from Table 4.5 for the timing results on serial code.

We also include the timing results for $\{10, 20, 40, 80\}$ particles in the same size of computational domain in Table 4.6, i.e., the grid points of the domains remain the same over four cases of simulations. This timing results are collected from single CPU performance. Since the QBX method is highly parallelizable, the

Table 4.4 Convergence Test 4: The number of panel segments $N_{segment} = 16$ and the Tolerance $tol_{GMRES} = 1.0 \times 10^{-13}$ are Fixed; Total Grid Number is 100 × 100

N_{panel}	N_{bdary}	Iterations	l_2 error
20	340	10	4.80×10^{-10}
40	680	10	6.56×10^{-11}
80	1360	10	4.02×10^{-12}

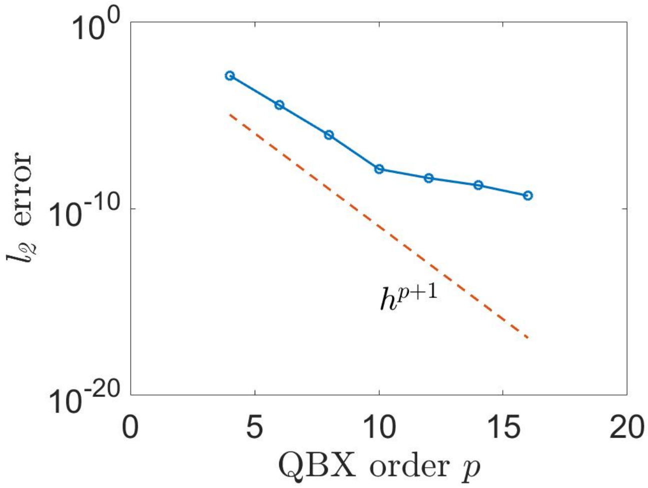


Figure 4.6 Numerical accuracy of the QBX method when the order of QBX varies. h is the arc length of each sub-panel of the boundary and $N_{panel} = 20$.

computational cost is expected to optimize with the use of parallel computing. The positions and orientations of particles are generated randomly in MATLAB where the initial configurations are shown in Figure 4.9.

4.4 Conclusion and Ongoing Works

Our numerical results have shown that with the use of the certain sets of parameters in the QBX method the ℓ_2 error can achieve a least single precision. Moreover, since the FMM is also implemented in solving the solution of BVP, the error due to the FMM is also tunable by adjusting the order of the FMM.

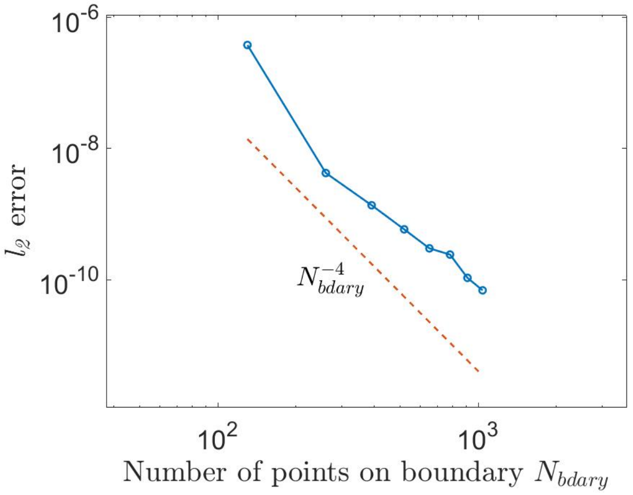


Figure 4.7 Numerical accuracy of the QBX method when the order of QBX fixed and N_{bdary} varies.

Table 4.5 Timing results 1: One Particle in the Domain with Different Number of Grid Points. The number of panel segments $N_{segment} = 8$, $N_{panel} = 20$ and $tol_{GMRES} = 1.0 \times 10^{-13}$ are Fixed

N_{grid}	Iterations	$T_{total}(sec)$	l_2 error
100×100	10	72.38	8.46×10^{-7}
200×200	10	85.73	8.46×10^{-7}
300×300	10	104.47	8.46×10^{-7}
400×400	10	128.23	8.46×10^{-7}
500×500	10	158.44	8.46×10^{-7}
600×600	10	196.25	8.46×10^{-7}
700×700	10	238.62	8.46×10^{-7}
800×800	10	288.62	8.46×10^{-7}

From performing the numerical experiments of particle dynamics, we observe that even the refinement of the quadrature evaluation points on the particle surfaces is turned on, once the particle boundaries are extremely close to each other, the numerical solver will not be able to handle the topology break. Therefore, for the

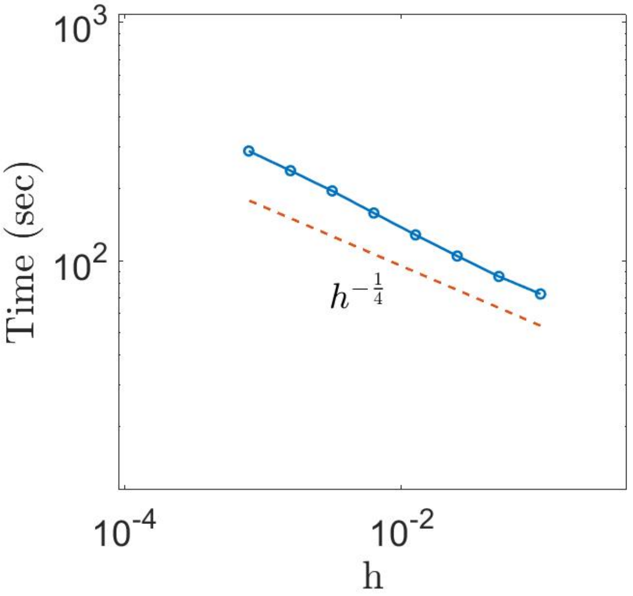


Figure 4.8 Timing results the cases when h varies where h is the length of each subpanel in computational domain. QBX order = 8, $N_{panel} = 20$ and $tol_{GMRES} = 1.0 \times 10^{-13}$ are fixed.

Table 4.6 Timing Results 3: For $N = \{10, 20, 40, 80\}$ Particles in the Domain where $L_x = 20, L_y = 20$.

$N_{particle}$	QBX order	N_{grid}	N_{bdary}	tol_{GMRES}	Iterations	T_{total}(sec)
10	4	500×500	50	1.0×10^{-8}	19	158.46
20	4	500×500	50	1.0×10^{-8}	21	260.68
40	4	500×500	50	1.0×10^{-8}	24	572.65
80	4	500×500	50	1.0×10^{-8}	29	1335.20

continuing work one direction is to enhance the model based on the physics. We may need to include an artificial hard-sphere excluded volume potential similar to the one described in Chapter 2.

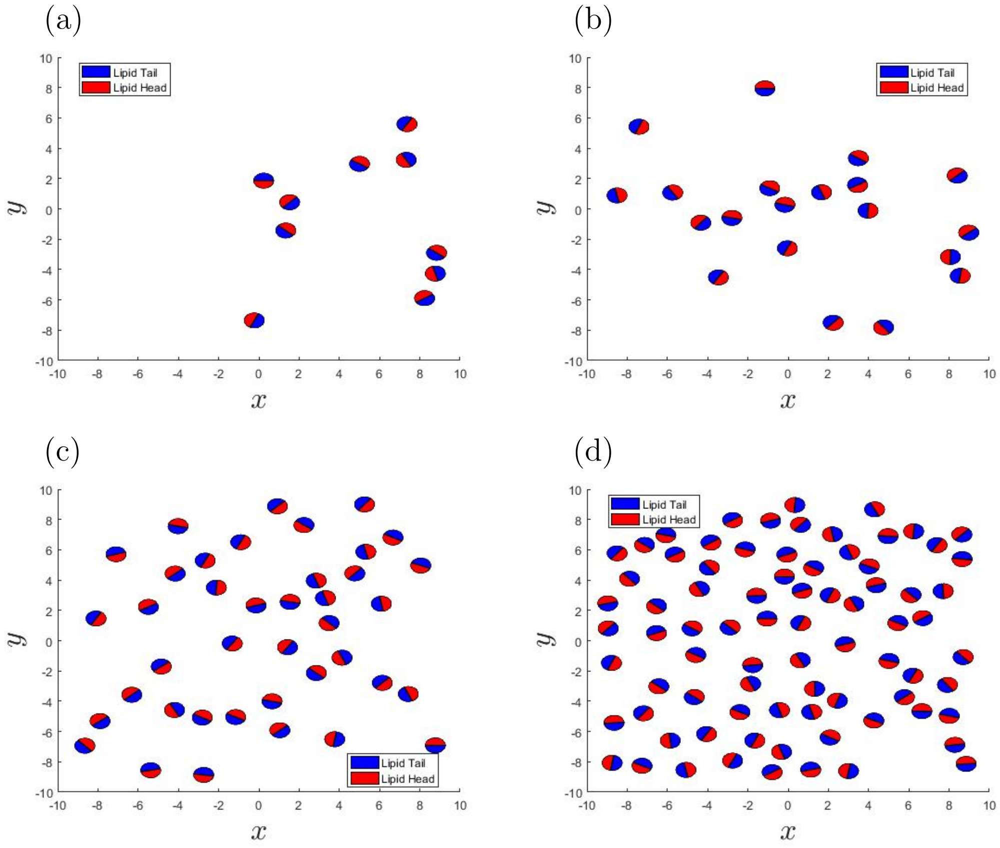


Figure 4.9 (a)–(d) Randomly distributed particles in the same size of domain where $N = \{10, 20, 40, 80\}$.

CHAPTER 5

CONCLUSIONS AND POTENTIAL FUTURE WORK

5.1 Conclusions

In this dissertation, we have shown that we have established capability to simulate from simple biological systems ($N \sim O(10^1) - O(10^3)$) such as DNAs, long-chain polymers immersed in fluid to large systems ($N \sim O(10^4) - O(10^6)$), for instance, vesicles and RBCs with explicit fluid. Also, in order to include as much as known molecular details, we have demonstrated several levels of coarse-graining from the size of $O(1)$ nm to $O(10^{-1})$ μm in particle diameter size. For these numerical simulations, the computational cost is always an important factor. We aim to take advantage of parallel computing such as MPI and OpenMP. Moreover, the fact is that all the work introduced in this dissertation except the LAMMPS simulations are expected to be highly parallelized, for examples: (1) We modified and implemented the Metropolis integrator scheme and the FMM in DNA simulations. For the parallelized MPIFMM, the expensive matrix-vector multiplications can be expedited when the particle distributions are nonuniform [6,99]. For the acceptance probability mentioned in Section 2.2, recent literature have provided a general idea for parallelizing the covariance matrix [3, 15]; (2) The QBX scheme involving in IEM has been built in the fashion of parallelism and GPU implementation [72]. To include a Stokes flow in Janus particle simulations, the large system simulations of DNA dynamics can be a basis of continuing works. The trivial direction of the potential research can refer to the recent published journal article: integral equation formulation for rigid bodies in Stokes flow [18].

5.2 Potential Future Works

5.2.1 CGBD Simulations of DNA Molecules/Vesicles/RBCs in Confined Geometry

Traditional integrators (such as the forward Euler integrator) for SDE often give erroneous results when applied to a confined domain, such as a random walker in a finite box modeled as an infinite potential well. Due to numerical errors of the numerical integrator, the particle often gets trapped near the box boundary where the potential goes to infinity. Such unphysical trapping can be avoided by using the Metropolized stochastic integrators. However, for particles suspended in fluid flow near a solid substrate, the forces from the wall on the particle must be taken into account. Such corrections of Stokeslet and stresslet for Stokes flow in the presence of a solid wall have been investigated [4,8]. Recently Gimbutas *et al.* have incorporated these corrections in FMM to simulate flow through a porous medium with complex geometry [27]. Furthermore, Marple *et al.* managed to simulate a vesicle suspension in a confined periodic channel by successfully dealing with the vesicle-vesicle interactions in a highly packed vesicle suspension [58]. Inspired by these developments, we propose to derive such correction(s) consistent with the CGBD formulation and implement an efficient implicit solvent algorithm for the nonlocal HIs for CGBD in a confined geometry.

Results from the laboratory experiments on cell-wall interactions (RBC adhesion [50] and adhesion of a giant vesicle on a glass substrate [94]) suggest that the interactions between coarse-grained membrane and a treated substrate are even more challenging because the membrane-wall interactions may be dependent on the properties of the wall and the ionic strength in the solution. Modeling work on lipid membrane adhesion [9, 14, 23, 97] shows that, depending on the membrane-wall interactions, the lipid membrane may exhibit non-trivial dynamics (lipid circulation) and shapes. We propose to adopt the approaches in [27, 58] to investigate the membrane-wall interactions in CGBD.

5.2.2 Study of Membrane and Water Bridge System

In recent years broad interests of research have been brought to the physics of nanoscale water bridge system in water capillary analysis. Different from the continuum modeling which adopts the constant contact angle, the target contact angle is related to the water-wall interactions and the size of the system. Due to the nature of dipolar water molecules, not only Lennard-Jones (L-J) but also long rang Coulomb forces are included. Following from the article [90] for numerical observations of force exerted on capillary water-bridge, we perform full-atomic nanoscale simulations in LAMMPS. With the fixed substrates in periodic domain and constant number of water molecules, we vary the height of the water-bridge in order to compare the numerical results with theoretical capillary force.

Vega *et al.* provide the surface tension of the most popular water models by using so-called test-area simulation method [93]. Therefore we can obtain the local surface tension from the averaged pressure components and the formula is given by

$$\gamma = \frac{L_z}{2}[\bar{p}_N(z) - \bar{p}_T(z)], \tag{5.1}$$

where $\bar{p}_N(z)$ and $\bar{p}_T(z)$ are the mean normal and tangential pressure components at position z. Particularly, at temperature $T = 300$°K the surface tension $\gamma = 65.4$ mJ/m^2 = 65.4 mN/m where this value is averaged over virial route and test-area simulation method without long range correction [93].

The water model we adopt is TIP4P/2005 [1] where the hydrogen atoms (H) connect the oxygen atom (O) with fixed chemical bonds and each water molecule maintains a fixed bond angle over simulation time. A Lennard-Jones (L-J) potential and a Coulombic interaction with a fixed charged point (M) are considered in the inter-molecule interactions as described in TIP4P/2005 model. Table 5.1 shows the parameter sets and the detailed comparisons between modern water models can be found in [38, 92].

Two substrates are formed in 3.5Å fcc lattice and the material we use is based on the chemical properties of the silicon. In the preliminary simulations, two substrates are not charged and we will compare our results with charged walls in the later work. A strong L-J potential is considered on the wall particles with no initial velocity. By simulating in a large periodic computational domain (262.5Å, 262.5Å, 262.5Å), we can ensure that there is no "top wall"-"bottom wall" interactions when the walls are located far enough from the domain boundary in z direction. Hence, it is fair enough to claim that our simulations are performed in x-y periodic domain. Table 5.2 shows the parameter set we utilize in particle pair-interactions. Moreover, during simulation the wall particles are subject to have no motion and moving substrate is definitely an another interesting topic to be explored.

The unit setup in LAMMPS we utilize is called "real" which uses angstrom as characteristic length and femtosecond (fs) as characteristic time. For all simulations performed in this work we use $\Delta t = 4$fs as our time step size.

Table 5.1 Parameter Set for TIP4P/2005 Water Model

O mass	H mass	O charge	H charge
15.9994	1.008	-1.0484	0.5242
OH bond	HOH angle	O-O (ϵ, σ)	O-H, H-H (ϵ, σ)
0.9572Å	104.52°	0.16275, 3.16435	0.0, 0.0

The Coulomb force for particle pair $\{r_i, r_j\}$ is given by the following formula:

$$E = C\frac{q_i q_j}{\epsilon|r_i - r_j|}, \quad |r_i - r_j| < r_c, \tag{5.2}$$

Table 5.2 Parameter Set for L-J Interactions between Walls, Oxygen and Hydrogen

Pair	wall-wall	O-O	wall-O	wall-H
Our simulation (ϵ, σ)	16.004, 2.471	0.16275, 3.16435	0.439, 2.5	0.0, 2.5

where ϵ is an the dielectric constant and C is energy-conversion constant. To expedite the computational speed, LAMMPS incorporates the so-called particle-particle particle-mesh solver (p^{3}m) which uses three-dimensional FFT to solve interparticle Coulomb forces in $O(\mathrm{N}\log\mathrm{N})$ operators.

Recall from the theoretical capillary force, we have the following relationship:

$$F = 2\pi R_{wb}\gamma\sin\alpha, \tag{5.3}$$

where we use the values $\gamma = 65.4\mathrm{mN/m}$ and R_{wb} is the radii observed from the contact surfaces between water-bridge and both walls.

Table 5.3 gives the dimensions of all six cases we study and we set the number of water molecules to be a fix number which is $N = 12000$. We first provide some snapshots from visualizations of all cases in Figure 5.1. The computational domain has periodic boundary conditions in all directions. The global cutoff radius is 9 Å which is much smaller than the side of the simulation box $L_x = L_y = L_z = 262.5$ Å. The advantage of this setup is to prevent the effect of duplicated pair-interactions.

Table 5.3 Dimensions for Simulations

Initial # of Oxygen in height	5	10	15	20	25	30
Bottom wall (Å)	120.75	115.5	108.5	96.25	92.75	85.75
Top wall (Å)	141.75	150.5	157.5	162.75	173.25	180.25
Height (Å)	21.0	35.0	49.0	64.75	80.5	94.5

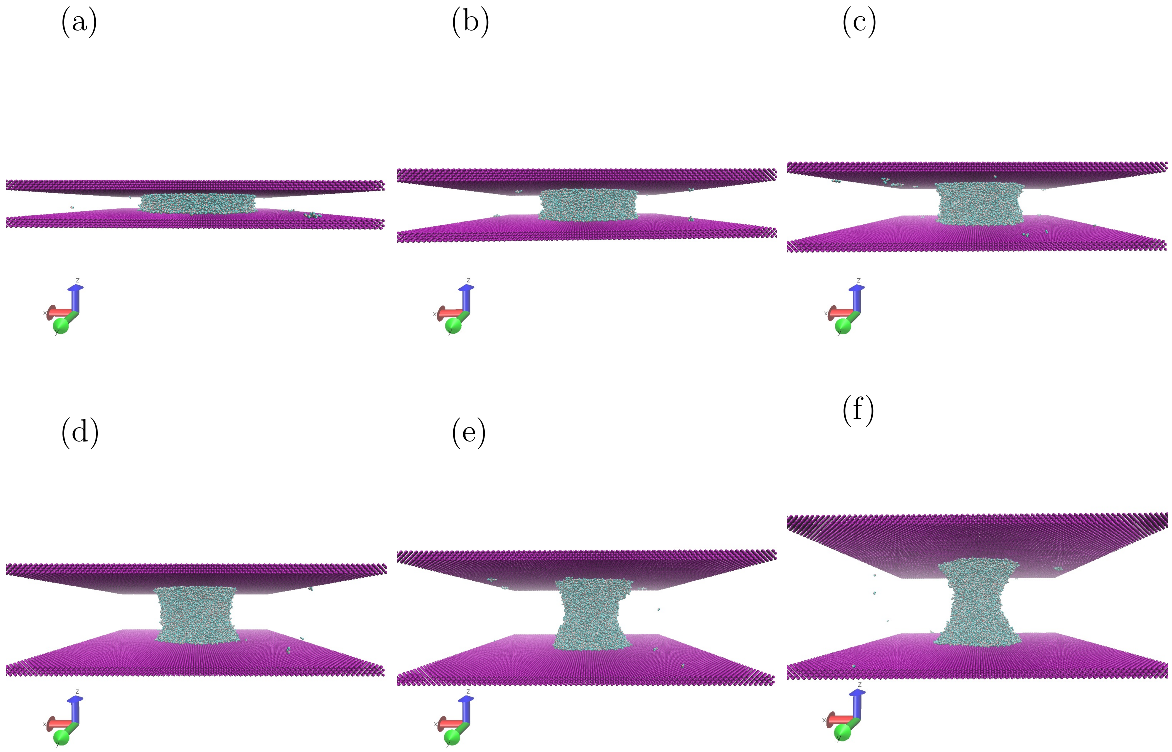


Figure 5.1 VMD snapshots for six cases of simulations. Panel (a)–(f): bridge height $= \{21.0\text{Å}, 35.0\text{Å}, 49.0\text{Å}, 64.75\text{Å}, 80.5\text{Å}, 94.5\text{Å}\}$.

After analyzing the properties of the nano-scale water bridge system we aim to include the membrane into this water bridge. That is, with the use of existing membrane model and water model in LAMMPS, we can study the conformation changes due to the membrane-bridge interactions and van der Waals effects.

5.2.3 Three Dimensional Action Field Approach Using Integral Equation Method

As the continuation of the two-dimensional action field approach on Janus particle simulation using IEM, three dimensional version of numerical implementation is not going to be extremely complicate since the main challenge is the optimization of three dimensional QBX which may take humongous time to evaluate the volume potentials.

Otherwise, with the same setup of numerical algorithm, the only change will be the Yukawa kernel within different system dimensions.

5.2.4 Other Possible Related Projects

Some interesting projects are listed below:

1. Molecular Dynamics Simulations for Micro-pipette Aspiration of Cells.
2. Power law rheology of filament, DNA and cell.
3. Parallelization of general Brownian dynamics simulator: combination of the FMM and the Metropolis.
4. With existing DNA and vesicle models, protein insertion into membranes is a very interesting subject.
5. Adding ion channels on membrane.
6. Polymer Block in both free space and confined geometry.

APPENDIX A

LAMMPS INPUT SCRIPT FOR MEMBRANE SIMULATIONS

For the newly developed pair-potential function, after implementation is completed in LAMMPS, we name it as

```
pair_fluidmembrane
```

where the package is available for download on the following program doi:

```
http://dx.doi.org/10.17632/4v53nkv5hc.1
```

and this package includes: (1) one LAMMPS function script and one LAMMPS header file; (2) one LAMMPS bond function; (3) one MATLAB script for generating the initial configuration for LAMMPS simulations and lastly, (4) we provide an example for simple test run for users.

A.1 Implementation in LAMMPS

For vesicle simulation, it includes membrane, internal water molecules and external water molecules. In the case of CGMD RBC simulations, an addition of cytoskeleton network is generated. In LAMMPS input, the very beginning step is to determine the "unit" of the simulations. Since all the variables and parameters are dimensionless quantities, we set the LAMMPS unit style to be

```
units lj
```

and define the mass of coarse-grained particles as

```
mass bilayer 1.0
```

We then specify multiple atom styles using hybrid function and call “ellipsoid”, “molecular” and “peri”. Here we provide the LAMMPS command in the script:

```
atom_style hybrid ellipsoid peri molecular
```

“peri” is for extracting the initial configurations (x_0 in LAMMPS) of all coarse-grained particles including bilayer membrane and cytoskeleton network [65]. The data format provided below is for satisfying the hybrid atom style listed in Table A.1 from LAMMPS guidelines:

```
atom-ID atom-type x y z ellipsoidflag
density volume density molecule-ID
```

Table A.1 The Atom Styles Used in Current Numerical Simulations

ellipsoid	atom-ID atom-type ellipsoidflag density x y z
peri	atom-ID atom-type volume density x y z
molecular	atom-ID molecule-ID atom-type x y z
hybrid	atom-ID atom-type x y z sub-style1 sub-style2 sub-style3

To call our proposed pair-potential interaction function in LAMMPS, the input commands are given as follows:

```
pair_style hybrid lj/cut 3.6 fluidmembrane 2.6
pair_coeff 1*2 1*2 fluidmembrane 1.0 1.0 2.6 4 3 0
```

where we give the parameters for global cut-off lengths in the pair-potential. The above LAMMPS commands can also be used to call the Lennard-Jones potential, which is used to model the interaction with explicit solvents and internal fluid in Section 3.4. The sequence of parameters for "fluidmembrane" pair function are given by:

$$\epsilon \quad \sigma \quad r_{cut} \quad \zeta \quad \mu \quad \sin\theta_0.$$

Recall from Section 3.1.3, we have total 7 types of coarse-grained particles in RBC simulations (3 types in vesicle simulations): type 1: lipid membrane; type 2: transmembrane protein; type 3: junction complexes; type 4: spectrin tetramers; type 5: ankyrin; type 6: internal water and type 7: external water. Excerpt from the provided input script for LAMMPS simulations, we have the following lines to adopt the thermostat algorithms:

```
variable ini_T equal 0.02
variable T equal 0.23
variable P equal 0.05
variable LD equal 1.0
variable P_damp equal 1
fix 1 water npt temp ${T} ${T} ${LD} iso ${P} ${P} ${LD}
fix 2 network npt temp ${int_T} ${T} ${LD} iso ${P} ${P} ${P_damp}
```

```
fix 3 bilayer nvt/asphere temp ${T} ${T} ${LD}
```

As mentioned in Section 3.2.3, a ramp function is used to gradually adjust the equilibrium length σ_{eq} over time to control the cell volume after the desired initial configuration of the cell is obtained.

```
variable scale1 equal ramp(2.7,2.66)

fix 4 water adapt 1 pair lj/cut sigma 6 6 v_scale1
```

where the `fix` command is to assign an operator or algorithm into LAMMPS system. To implement the specified interactions in LAMMPS, we have the following commands in the input script:

```
pair_coeff 1*2 6*7 lj/cut 0.2 1.0
pair_coeff 1*2 3*5 lj/cut 0.2 1.0
pair_coeff 3*5 3*5 lj/cut 0.2 1.0
pair_coeff 3*5 6*7 lj/cut 0.2 1.0
pair_coeff 6*7 6*7 lj/cut 0.2 2.7
```

A.2 Modified Harmonic Bond Implementation in LAMMPS

As needed in RBC simulations, we have to call the modified harmonic bond function to obtain the stress-free configuration of cytoskeleton. Therefore, we have to specify the bond style in LAMMPS script which is provided by the following:

```
bond_style harmonic1
```

The lines below are the codes we modified from existed bond function:

```
bond_harmonic.cpp
```

and created a new C++ source code and a header file with the name

```
bond_harmonic1.cpp
bond_harmonic1.h
```

Here is the code which calculates the bond lengths l_0 in Equation (3.9):

```
double **x0 = atom → x0;
double **x = atom → x;
delx0 = x0[i1][0] - x0[i2][0];
dely0 = x0[i1][1] - x0[i2][1];
delz0 = x0[i1][2] - x0[i2][2];
delx = x[i1][0] - x[i2][0];
dely = x[i1][1] - x[i2][1];
delz = x[i1][2] - x[i2][2];
rsq = delx⋆delx + dely⋆dely + delz⋆delz;
r = sqrt(rsq);
l0 = sqrt(delx0*delx0 + dely0*dely0 + delz0*delz0);
dr = r - l0;
rk = k[type]*dr;
```

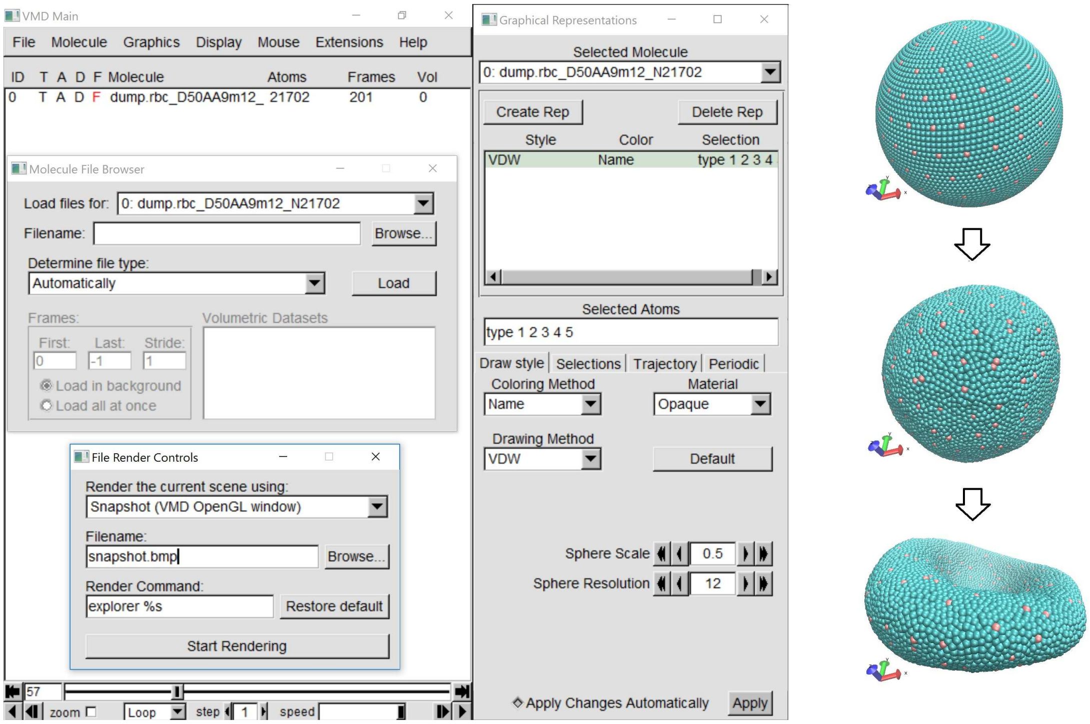
A.3 Descriptions of the Subprograms and Sample Output

The subprograms include:

- create_rbc_with_water (MATLAB),
- bond_harmonic1 (C++),
- in_example (LAMMPS input script).

We have provided the detail of LAMMPS implementations for the LAMMPS input script and modified harmonic bond function in the sections above. Here we include a sample run for RBC simulation where the initial configuration of the RBC is a sphere. The MATLAB script generates an initial data file for the configurations of all atoms including coarse-grained lipid bilayer membrane, cytoskeleton network and water molecules where the generated data file satisfies the file format described above at the beginning of Section A.1. LAMMPS is capable to assign specific regions for simple geometries such as planar membrane or spherical surfaces and fill the regions with desired atoms. For this example, we require a hexagonal network to represent the cytoskeleton network which is not trivial to be done in the LAMMPS input script. Therefore, with the use of MATLAB script as supplementary tool we can create complicated shapes of objects.

The provided LAMMPS input script dumps a LAMMPS trajectory file (.lammpstrj) and it can be read in various of visualization softwares. For this work, Visual Molecular Dynamics (VMD) 1.9.1 is used to generate snapshots from the simulation data and here we show the screen-shot of VMD setting windows and the snapshots of simulation output (without showing water molecules) for this example in Figure A.1.



ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	dump.rbc_D50AA9m12_	21702	201	0

Load files for: 0: dump.rbc_D50AA9m12_N21702

Filename:

Determine file type: Automatically

Frames: First: 0 Last: -1 Stride: 1

◉ Load in background

○ Load all at once

Render the current scene using: Snapshot (VMD OpenGL window)

Filename: snapshot.bmp

Render Command: explorer %s

57

Loop

step 1

Selected Molecule: 0: dump.rbc_D50AA9m12_N21702

Style	Color	Selection
VDW	Name	type 1 2 3 4

Selected Atoms: type 1 2 3 4 5

Coloring Method: Name

Material: Opaque

Drawing Method: VDW

Sphere Scale 0.5

Sphere Resolution 12

Figure A.1 Screenshot of VMD setting windows and 3 snapshots from the output of sample RBC simulation. From top to bottom of the snapshots are (1) initial state of RBC; (2) equilibrium state of stress-free RBC; (3) resting shape of RBC after performing the volume control algorithm.

BIBLIOGRAPHY

[1] J. L. F. Abascal and C. Vega. A general purpose model for the condensed phases of water: Tip4p/2005. *J. Chem. Phys.*, 123:234505, 2005.

[2] S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. W. Hogg, and M. O'Neil. Fast direct methods for gaussian processes and the analysis of nasa kepler mission data. *Preprint*, 2014.

[3] M. Anitescu, J. Chen, and L. Wang. A matrix-free approach for solving the gaussian process maximum likelihood problem. *SIAM J. Sci. Comput.*, 34(1):A240–A262, 2012.

[4] V. Aranda, R. Cortez, and L. Fauci. A model of stokesian peristalsis and vesicle transport in a three-dimensional closed cavity. *J. Biomechanics*, 48(9):1631–1638, 2015.

[5] H. P. Babcock, D. E. Smith, J.S. Hur, E. S. G. Shaqfeh, and S. Chu. Relating the microscopic and macroscopic response of a polymeric fluid in a shearing flow. *Phys. Rev. Lett.*, 85(9):2018–2021, 2000.

[6] A. B. Benson, J. Poulson, K. Tran, B. Engquist, and L. Ying. A parallel directional fast multipole method. *SIAM J. Sci. Comput.*, 36(4):C335–C352, 2014.

[7] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81(8):3684–3690, 1984.

[8] J. R. Blake. A fast multipole method for the evaluation of elastostatic fields in a half-space with zero normal stress. *Proc. Camb. Phil. Soc*, 42:303–310, 2016.

[9] E. M. Blokhuis and W. F. C. Sager. Vesicle adhesion and microemulsion droplet dimerization: Small bending rigidity regime. *J. Chem. Phys.*, 66(15):7062–7074, 1999.

[10] D. Boal. *Mechanics of the Cell Second Edition.* Cambridge University Press, Cambridge, United Kingdom, 2012.

[11] N. Bou-Rabee. Time integrators for molecular dynamics. *Entropy*, 16:138–162, 2014.

[12] N. Bou-Rabee, A. Donev, and E. Vanden-Eijnden. Metropolis integration schemes for self-adjoint diffusions. *Multiscale Model. Simul.*, 12(2):781–831, 2014.

[13] H. Brakhage and P. Werner. Über das dirichletsche außenraumproblem für die helmholtzsche schwingungsgleichung. *Archiv der Mathematik*, 16(1):325–329, 1965.

[14] R. Capovilla and J. Guven. Geometry of lipid vesicle adhesion. *Phys. Rev. E*, 66:041604, 2001.

[15] J. Chen, M. Anitescu, and Y. Saad. Computing f(a)b via least squares polynomial approximations. *SIAM J. Sci. Comput.*, 33:195–222, 2011.

[16] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.*, 155:468–498, 1999.

[17] I. R. Cooke, K. Kremer, and M. Deserno. Tunable generic model for fluid bilayer membranes. *Phys. Rev. E*, 72:011506, 2005.

[18] E. Corona, L. Greengard, M. Rachh, and S. Veerapaneni. An integral equation formulation for rigid bodies in stokes flow in three dimensions. *J. Comput. Phys.*, 332:504 – 519, 2017.

[19] M. Doi and S. F. Edwards. *The Theory of Polymer Dynamics*. Oxford Science Publications, New York, 1986.

[20] J.-M. Drouffe, A. C. Maggs, and S. Leibler. Computer simulations of self-assembled membranes. *Science*, 254:1353, 1991.

[21] Q. Du, C. Liu, R. Ryham, and X. Wang. Energetic variational approaches in modeling vesicle and fluid interactions. *Physica D*, 238:923–930, 2009.

[22] Q. Du and J. Zhang. Adaptive finite element method for a phase field bending elasticity model of vesicle membrane deformations. *SIAM J. Sci. Comput.*, 30(3):1634–1657, 2008.

[23] A. Efremov and J. Cao. Bistability of cell adhesion in shear flow. *Biophys. J.*, 101:1032–1040, 2011.

[24] D. L. Ermak and J. A. McCammon. Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.*, 69:1352–1360, 1978.

[25] S.-P. Fu, Z. Peng, H. Yuan, R. Kfoury, and Y.-N. Young. Lennard-jones type pair-potential method for coarse-grained lipid bilayer membrane simulations in lammps. *Comput. Phys. Commun.*, accepted.

[26] S.-P. Fu, Y.-N. Young, and S. Jiang. Efficient brownian dynamics simulation of dna molecules with hydrodynamic interactions in linear flows. *Phys. Rev. E*, 91:063008, 2015.

[27] Z. Gimbutas and L. Greengard. A fast multipole method for the evaluation of elastostatic fields in a half-space with zero normal stress. *Adv. Comput. Math.*, 42:175–198, 2016.

[28] R. Goetz and R. Lipowsky. Computer simulations of bilayer membranes: Self-assembly and interfacial tension. *J. Chem. Phys.*, 108(17):7397–7409, 1998.

[29] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.

[30] L. F. Greengard and J. Huang. A new version of the fast multipole method for screened coulomb interactions in three dimensions. *J. Comput. Phys.*, 180:642–658, 2002.

[31] B. E. Griffith and C. S. Peskin. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. *J. Comput. Phys.*, 208:75–105, 2005.

[32] N. Grønbech-Jensen, J. R. Hayre, and O. Farago. Application of the g-jf discrete-time thermostat for fast and accurate molecular simulations. *J. Phys. Commun.*, 185:524–527, 2014.

[33] F. Hauer, W. Marth, S. Li, J. Lowengrub, A. Ratz, and A. Voigt. Thermodynamically consistent models for two component vesicles. *Int. J. Biomath. Biostat.*, 2(1):19–48, 2013.

[34] W. Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Z. Naturforsch. C*, 28(11):693–703, 1973.

[35] K. L. Ho and L. Ying. Hierarchical interpolative factorization for elliptic operators: differential equations. *arXiv:1307.2895*, 2013.

[36] K. L. Ho and L. Ying. Hierarchical interpolative factorization for elliptic operators: integral equations. *arXiv:1307.2666*, 2013.

[37] W. G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31(3):1695–1697, 1985.

[38] H. W. Horn, W. C. Swope, J. W. Pitera, J. D. Madura, T. J. Dick, G. K. Hura, and T. Head-Gordon. Development of an improved four-site water model for biomolecular simulations: Tip4p-ew. *J. Chem. Phys.*, 120(20):234505, 2004.

[39] C.-C. Hsieh, L. Li., and R. G. Larson. Modeling hydrodynamic interaction in brownian dynamics: simulations of extensional flows of dilute solutions of dna and polystyrene. *J. Non-Newton Fluid*, 113:147–191, 2003.

[40] W.-F. Hu, Y. Kim, and M.-C. Lai. An immersed boundary method for simulating the dynamics of three-dimensional axisymmetric vesicles in navier–stokes flows. *J. Comput. Phys.*, 257:670–686, 2014.

[41] W.-F. Hu and M.-C. Lai. Unconditionally energy stable immersed boundary method with application to vesicle dynamics. *East Asian J. Applied Math.*, 3:247–262, 2013.

[42] R. M. Jendrejack, J. J. de Pablo, and M. D. Graham. Stochastic simulations of dna in flow: Dynamics and the effects of hydrodynamic interactions. *J. Chem. Phys.*, 116:7752–7759, 2002.

[43] R. M. Jendrejack, M. D. Graham, and J. J. de Pablo. Hydrodynamic interactions in long chain polymers: Application of the chebyshev polynomial approximation in stochastic simulations. *J. Chem. Phys.*, 113:2894–2900, 2000.

[44] S. Jiang, Z. Liang, and J. Huang. A fast algorithm for brownian dynamics simulation with hydrodynamic interactions. *Math Comput.*, 82:1631–1645, 2013.

[45] E. Jung and C. S. Peskin. Two-dimensional simulations of valveless pumping using the immersed boundary method. *SIAM J. Sci. Comput.*, 23(1):19–45, 2001.

[46] S. Kawamoto, M. Klein, and W. Shinoda. Coarse-grained molecular dynamics study of membrane fusion: Curvature effects on free energy barriers along the stalk mechanism. *J. Chem. Phys.*, 143:243112, 2015.

[47] A. Klöckner, A. Barnett, L. Greengard, and M. O'Neil. Quadrature by expansion: A new method for the evaluation of layer potentials. *J. Comput. Phys.*, 252:332–349, 2013.

[48] M.-C. Lai and C. S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *J. Comput. Phys.*, 160:705–719, 2000.

[49] R. G. Larson, H. Hua, D. E. Smith, and S. Chu. Brownian dynamics simulations of a dna molecule in an extensional flow field. *J. Rheol.*, 46:267–304, 1999.

[50] Ph. Lavalle, J.-F. Stoltz, B. Senger, J.-C. Voegel, and P. Schaaf. Red blood cell adhesion on a solid/liquid interface. *Proc. Natl. Acad. Sci. USA*, 93:15136–15140, 1996.

[51] H. Li and G. Lykotrafitis. Two-component coarse-grained molecular-dynamics model for the human erythrocyte membrane. *Biophys. J.*, 102:75–84, 2012.

[52] S. Li, J. S. Lowengrub, and A. Voigt. Locomotion, wrinkling, and budding of a multicomponent vesicle in viscous fluids. *Comm. Math. Sci*, 10(2):645–670, 2012.

[53] Z. Liang, Z. Gimbutas, L. Greengard, J. Huang, and S. Jiang. A fast multipole method for the rotne-prager-yamakawa tensor and its applications. *J. Comput. Phys.*, 234:133–139, 2013.

[54] G. Lim H. W., M. Wortis, and R. Mukhopadhyay. Stomatocyte-discocyte-echinocyte sequence of the human red blood cell: Evidence for the bilayer-couple hypothesis from membrane mechanics. *PNAS*, 99(26):16766–16769, 2002.

[55] J. S. Lowengrub, J-J. Xu, and A. Voigt. Surface phase separation and flow in a simple model of multicomponent drops and vesicles. *Fluid Dyn. Mater. Proc.*, 3:1–19, 2007.

[56] E. Lushi, H. Wioland, and R. E. Goldstein. Fluid flows created by swimming bacteria drive self-organization in confined suspensions. *PNAS*, 111(27):9733–9738, 2014.

[57] J. F. Marko and E. D. Siggia. Stretching dna. *Macromolecules*, 28:8759–8770, 1995.

[58] G. Marple, A. Barnett, A. Gillman, and S. Veerapaneni. A fast algorithm for simulating multiphase flows through periodic geometries of arbitrary shape. *SIAM J. Sci Comput.*, 38(5):B740–B772, 2016.

[59] S. Marčelja. Structural contribution to solute-solute interaction. *Croat. Chem. Acta.*, 49:347–357, 1977.

[60] H. Noguchi. Membrane simulation models from nanometer to micrometer scale. *J. Phys. Soc. Jpn.*, 48(4):041007, 2009.

[61] H. Noguchi and G. Gompper. Meshless membrane model based on the moving least-squares method. *Phys. Rev. E*, 73:021903, 2006.

[62] H. Noguchi and M. Takasu. Self-assembly of amphiphiles into vesicles: A brownian dynamics simulation. *Phys. Rev. E*, 64:041913, 2001.

[63] S. D. Olson, L. Fauci, and S. S. Suarez. Mathematical modeling of calcium signaling during sperm hyperactivation. *Mol. Hum. Reprod.*, 17(8):500–510, 2011.

[64] O. S. Pak, Y.-N. Young, G. R. Marple, S. Veerapaneni, and H. A. Stone. Gating of a mechanosensitive channel due to cellular flows. *PNAS*, 112(32):9822–9827, 2015.

[65] M. L. Parks, R. B. Lehoucq, S. J. Plimpton, and S. A. Silling. Implementing peridynamics within a molecular dynamics code. *Comput. Phys. Commun.*, 179:777–783, 2008.

[66] Z. Peng, X. Li, I. V. Pivkin, M. Dao, G. E. Karniadakis, and S. Suresh. Lipid bilayer and cytoskeletal interactions in a red blood cell. *PNAS*, 110(33):13356–13361, 2013.

[67] Z. Peng, A. Mashayekh, and Q. Zhu. Erythrocyte responses in low-shear-rate flows: effects of non-biconcave stress-free state in the cytoskeleton. *J. Fluid Mech.*, 742:96–118, 2014.

[68] T. T. Perkins, D. E. Smith, and S. Chu. Single polymer dynamics in an elongational flow. *Science*, 276:2016–2021, 1997.

[69] I. V. Pivkin and G. E. Karniadakis. Accurate coarse-grained modeling of red blood cells. *Phys. Rev. Lett.*, 101:118105, 2008.

[70] C. Pozrikidis. Resting shape and spontaneous membrane curvature of red blood cells. *Math. Med. Bio.*, 22:34–52, 2005.

[71] J. R. Prakash. Rouse chains with excluded volume interactions in steady simple shear flow. *J. Rheol.*, 46:1353–1380, 2002.

[72] M. Rachh, A. Klöckner, and M. O'Neil. Fast algorithms for quadrature by expansion i: Globally valid expansions, 2016.

[73] A. Rahimian, S. Veerapaneni, D. Zorin, and G. Biros. Boundary integral method for the flow of vesicles with viscosity contrast in three dimensions. *J. Comput. Phys.*, 298:766–786, 2015.

[74] J. Rotne and S. Prager. Variational treatment of hydrodynamic interaction in polymers. *J. Chem. Phys.*, 50:4831–4837, 1969.

[75] R. J. Ryham, T. S. Klotz, L. Yao, and F. S. Cohen. Calculating transition energy barriers and characterizing activation states for steps of fusion. *Biophys. J.*, 110:1110–1124, 2016.

[76] D. Salac and M. J. Miksis. Reynolds number effects on lipid vesicles. *J. Fluid Mech.*, 711:122–146, 2012.

[77] C. M. Schroeder, H. P. Babcock, E. S. G. Shaqfeh, and S. Chu. Observation of polymer conformation hysteresis in extensional flow. *Science*, 301:1515–1519, 2003.

[78] C. M. Schroeder, E. S. G. Shaqfeh, and S. Chu. Effect of hydrodynamic interactions on dna dynamics in extensional flow: Simulation and single molecule experiment. *Macromolecules*, 37:9242–9256, 2004.

[79] J. T. Schwalbe, P. M. Vlahovska, and M. J. Miksis. Monolayer slip effects on the dynamics of a lipid bilayer vesicle in a viscous flow. *J. Fluid Mech.*, 647:403–419, 2010.

[80] U. Seifert. Configurations of fluid membranes and vesicles. *Advances in Physics*, 46(1):13–137, 1997.

[81] U. Seifert, K. Berndl, and R. Lipowsky. Shape transition of vesicles: Phase diagram for spontaneous-curvature and bilayer-coupling models. *Phys. Rev. A*, 44(2):1182–1202, 1991.

[82] E. S. G. Shaqfeh. The dynamics of single-molecule dna in flow. *J. Non-Newtonian Fluid Mech.*, 130:1–28, 2005.

[83] J. Simons, L. Fauci, and R. Cortez. A fully three-dimensional model of the interaction of driven elastic filaments in a stokes flow with applications to sperm motility. *J. Biomechanics*, 48(9):1639–1651, 2015.

[84] K. Sinha and M. Graham. Dynamics of a single red blood cell in simple shear flow. *Phys. Rev. E*, 92:042710, 2015.

[85] D. E. Smith, H. P. Babcock, and S. Chu. Single-polymer dynamics in steady shear flow. *Science*, 283:1724–1727, 1999.

[86] D. E. Smith and S. Chu. Response of flexible polymers to a sudden elongational flow. *Science*, 281:1335–1340, 1998.

[87] M. Somasi, B. Khomami, N. J. Woo, J. S. Hur, and E. S. G. Shaqfeh. Brownian dynamics simulations of bead-rod and bead-spring chains: numerical algorithms and coarse-graining issues. *J. Non-Newtonian Fluid Mech.*, 108:227–255, 2002.

[88] D. Swigon, S. Lim, and Y. Kim. Dynamical simulations of dna supercoiling and compression. *Biochem. Soc. Trans.*, 41:554–558, 2013.

[89] A.-K. Tornberg and M. Shelley. Simulating the dynamics and interactions of flexible fibers in stokes flows. *J. Comput. Phys.*, 196:8–40, 2004.

[90] G. E. Valenzuela, J. H. Saavedra, R. E. Rozas, and P. G. Toledo. Force exerted by a nanoscale capillary water bridge between two planar substrates'. *Phys. Chem. Chem. Phys*, 18:11176–11183, 2016.

[91] S. K. Veerapaneni, D. Gueyffier, D. Zorin, and G. Biros. A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2d. *J. Comput. Phys.*, 228(7):2334–2353, 2009.

[92] C. Vega, J. L. F. Abascal, and I. Nezbeda. Development of an improved four-site water model for biomolecular simulations: Tip4p-ew. *J. Chem. Phys.*, 125:034503, 2006.

[93] C. Vega and E. de Miguel. Surface tension of the most popular models of water by using the test-area simulation method. *J. Chem. Phys.*, 126:154707, 2007.

[94] C. Vézy, G. Massiera, and A. Viallat. Adhesion induced non-planar and asynchronous flow of a giant vesicle membrane in an external shear flow. *Soft Matter*, 3:844–851, 2007.

[95] P. M. Vlahovska. Asymmetric shapes and pearling of a stretched vesicle. *J. Fluid Mech.*, 754:1–4, 2014.

[96] Y. Wang, J. K. Sigurdsson, E. Brandt, and P. Atzberger. Dynamic implicit-solvent coarse-grained models of lipid bilayer membranes: Fluctuating hydrodynamics thermostat. *Phys. Rev. E*, 88:023301, 2013.

[97] F. G. Woodhouse and R. E. Goldstein. Shear-driven circulation patterns in lipid membrane vesicles. *J. Fluid Mech.*, 705:165–175, 2012.

[98] J. Wrobel, S. Lynch, A. Barrett, L. Fauci, and R. Cortez. Enhanced flagellar swimming through a compliant viscoelastic network in stokes flow. *J. Fluid Mech.*, 792:775–797, 2016.

[99] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J. Comput. Phys.*, 196:591–626, 2004.

[100] H. Yuan, C. Huang, J. Li, G. Lykotrafitis, and S. Zhang. One-particle-thick, solvent-free, coarse-grained model for biological and biomimetic fluid membranes. *Phys. Rev. E*, 82:011905, 2010.

[101] H. Yuan, C. Huang, and S. Zhang. Dynamic shape transformations of fluid vesicles. *Soft Matter*, 6:4571–4579, 2010.